Chapter 3

NEUROSCALE

3.1 Introduction

The distance-preserving criteria for determining topographic mappings such as the Sammon mapping, or the majority of the multidimensional scaling models, are intuitively appealing. Simple STRESS measures of the form $\sum_{ij}(d^*_{ij} - d_{ij})^2$ explicitly embody the notion of structure-retention with their tendency to retain distance relationships on both local and global scales.

However, one major restrictive property of both the Sammon and MDS methods is that there is no *transformation* defined from the input space to the feature space. Configurations are generated by the direct iterative adjustment of their component vectors, and once determined, act effectively as look-up tables. There is no mechanism to project one or more new data points without expensively re-generating the entire configuration from the augmented dataset. In the neural network vernacular, there is no concept of *generalisation* for defined mappings.

For example, in a discriminatory application, a Sammon mapping might be constructed for a large dataset in order to reveal inherent clustering which may correspond to membership of particular classes. It would then be of benefit to project new data (of unknown class) and so permit inferences to be drawn concerning class membership from that projection, rather than undergoing the computationally expensive task of re-mapping the entire dataset with the new points included.

This problem has recently motivated several researchers to develop transformational variants of both the Sammon mapping and of certain MDS procedures. The transformation may be effected by a neural network, taking as its input the raw data, and generating the topographic configuration at its output. Such a model, when trained, can then be used to project novel data in the obvious manner by forward propagation through the network.

As an extension of this earlier work, this chapter introduces "NEUROSCALE" — an implementation of the Sammon mapping utilising a Radial Basis Function (RBF) feed-forward neural network. Such a model is a potentially powerful alternative to the established neural network paradigm, the Kohonen Self-Organising Feature Map (SOFM), and can be expected to offer several advantages over that latter approach. These advantages will be discussed in Section 3.2, along with a description of the training algorithm for the network.

An important feature of NEUROSCALE is its capacity to exploit additional available knowledge about the data, and to allow this to influence the mapping. This permits the incorporation of *supervisory* information in a technique which is strictly *unsupervised*, and this concept will be considered in depth in Section 3.3. The basic principles of the technique are then illustrated for some, mainly synthetic, datasets in Section 3.4. (An application to the visualisation and exploratory analysis of a difficult, real-world dataset will be presented in detail in the next chapter.)

This new approach is a development of previous work in the fields of topographic mapping, neural networks and feature extraction. The key research papers in these areas will be reviewed at the end of the chapter, and related to the NEUROSCALE technique.

3.2 Training a Neural Network Sammon Mapping

3.2.1 Relative Supervision

Clearly the training algorithm for a neural network implementation of the Sammon mapping is non-trivial. In a conventional *supervised* training scenario, there is an explicit 'target' for each input data point to be mapped to; in the case of a topographic transformation, only a measure of *relative* distance from all the other data points is available. A standard, supervised training algorithm cannot therefore be applied in this instance. This has led to the development of what has been termed a *relative supervision* algorithm [Lowe 1993], for the purposes of optimising error measures similar to the Sammon STRESS. This permits calculation of the weight derivatives required by most optimisation routines. Recall that the expression for the Sammon STRESS, ignoring normalisation terms, is of the form

$$E = \sum_{i}^{N} \sum_{j}^{N} (d^*_{ij} - d_{ij})^2. \tag{3.1}$$

The standard Euclidean distance metric will be assumed unless otherwise indicated (this is a sensible choice as it implies that configurations of points are rotationally invariant with respect to their STRESS measure), so

$$\begin{aligned} d_{ij} &= \| \mathbf{y}_i - \mathbf{y}_j \|, & (3.2)\\ &= \left[(\mathbf{y}_i - \mathbf{y}_j)^{\mathrm{T}} (\mathbf{y}_i - \mathbf{y}_j) \right]^{1/2}, & (3.3) \end{aligned}$$

and similarly for d^*_{ij}. In the standard Sammon mapping, STRESS is minimised by adjusting the location of the points $\mathbf{y}_i$ directly, according to a gradient-descent scheme. However, if each point $\mathbf{y}_i$ is defined as a parameterised nonlinear function of the input, such that $\mathbf{y}_i = \mathbf{f}(\mathbf{x}_i; \mathbf{w})$ where $\mathbf{w}$ is a parameter, or weight, vector, then the STRESS becomes

$$E = \sum_{i}^{N} \sum_{j}^{N} (d^*_{ij} - \| \mathbf{f}(\mathbf{x}_i; \mathbf{w}) - \mathbf{f}(\mathbf{x}_j; \mathbf{w}) \|)^2. \tag{3.4}$$

This expression may be differentiated with respect to the parameters $\mathbf{w}$ (rather than the actual points themselves in the case of the traditional Sammon mapping) and these parameters adjusted in order to minimise E. Weight derivatives are then calculated for *pairs* of input patterns, and the weights may be updated *on-line*, pattern-pair by pattern-pair, or may be subsequently updated in a *batch* fashion, after the presentation of all $(N-1)N/2$ possible combinations. Note that this concept is entirely general and not restricted to the neural network domain. The transformation function $\mathbf{f}(\cdot)$ may represent any arbitrary, continuous, differentiable function (even linear) and need not be a neural network model.

The formulation of a topographic mapping model in this manner has several advantages:

① As underlined previously, the existence of the transformation $\mathbf{f}(\cdot)$ permits the projection of unseen data, and affords the mapping a generalisation property. This is of major benefit as it allows the network to be used as a tool for future prediction and inference.

② The number of free parameters in the mapping may be reduced. For a Sammon mapping of N points to q dimensions, the number of adjustable parameters is $(N \times q)$, and is determined by the abundance of data alone. Effecting the mapping as a parameterised function allows the number of parameters to be determined according to the *complexity* of the problem. It would be intuitively expected that fewer than $(N \times q)$ parameters would be required in order to obtain reasonable performance in terms of generalisation.

③ A side-effect of this parameter reduction is that the nonlinear optimisation procedures employed to minimise the STRESS measure become more efficient. Some schemes, for example the quasi-Newton BFGS [Press et al. 1992], require memory storage that scales badly with the number of parameters.

3.2.2 Calculating Weight Derivatives

For the purposes of most nonlinear optimisation routines, the derivatives of the STRESS measure with respect to each parameter w_k are required. These may be calculated as follows.

Considering equations (3.1), (3.2) and (3.4) and applying the chain rule gives:

$$\frac{\partial E}{\partial w_k} = \sum_i^N \frac{\partial E}{\partial \mathbf{y}_i} \cdot \frac{\partial \mathbf{y}_i}{\partial w_k}, \tag{3.5}$$

$$= \sum_i^N \frac{\partial E}{\partial \mathbf{y}_i} \cdot \frac{\partial \mathbf{f}(\mathbf{x}_i; \mathbf{w})}{\partial w_k}. \tag{3.6}$$

The first term is simply that from Sammon's derivation and may be obtained by direct differentiation of equation (3.1) above to give

$$\frac{\partial E}{\partial \mathbf{y}_i} = -2 \sum_{j \neq i}^N \left(\frac{d^*_{ij} - d_{ij}}{d_{ij}} \right) (\mathbf{y}_i - \mathbf{y}_j). \tag{3.7}$$

The derivatives of the second term are also calculable directly and depend on the form of the function $\mathbf{f}(\cdot)$. In the case of a multilayer perceptron, the derivatives are those which are implicitly calculated using the familiar *back-propagation* procedure [Rumelhart, Hinton, and Williams 1986]. Alternatively, for a model linear in the weights, such as a radial basis function network with fixed centres, they may be directly derived in a straightforward fashion.

An illustrative code fragment of an implementation of this algorithm is given in figure 3.1. Note that although the algorithm must loop $O(N^2)$ times, the (potentially computationally expensive) forward and back-propagation through the network is only required N times.

Given the values of these derivatives, the network may be trained via any of the popular nonlinear optimisation algorithms — gradient-descent (with momentum), conjugate-gradient and BFGS are examples. (See [Bishop 1995, Ch. 7] for a detailed overview of those and other algorithms.)

```
// Relative Supervision Algorithm
//
// For training a Neural Network to effect a Sammon Mapping

// Initialise the weight changes vector to zero as in a standard 'batch' algorithm.
//
sumDerivatives = 0;
// Generate the set of output pattern vectors and zero the relative error vector
// for each point
//
for (n=0; n<numberOfPatterns; n++)
{
  relativeErrorVector[n]  = 0;
  networkOutput[n] = networkForwardPropagate(inputPattern[n]);
}

for (i=0; i<numberOfPatterns; i++)
{
  for (j=i+1; j<numberOfPatterns; j++)
  {
    d = distance(networkOutput[i], networkOutput[j]);
    if (d!=0)
    {
      // Calculate the relative error for points i and j
      // Note that the distance matrix dStar may be calculated in advance
      //
      tempVector = ((dStar(i,j) - d) / d) * (networkOutput[i]-networkOutput[j]);
      // Update the relative error for both points
      //
      relativeErrorVector[i] += tempVector;
      relativeErrorVector[j] -= tempVector;
    }
  }
  // Forward propagate through the network in order to back-propagate the
  // total relative error vectors, which are equivalent to dE/dy
  // in standard, supervised, back-propagation
  //
  networkForwardPropagate(inputPattern[i]);
  sumDerivatives += networkBackPropagate(relativeErrorVector[i]);
}
```

Figure 3.1: A code fragment to implement the relative supervision algorithm.

3.3 Exploiting Additional Knowledge

The relative supervision training algorithm as described in the previous section is a purely *unsupervised* procedure, in that no extra information concerning the data is utilised in the mapping. The network learns a transformation from the input space to the feature space, with the constraints on the output configuration imposed by the Euclidean distance function over the input vectors. This distance measure will be referred to as the *objective metric*, and its corresponding metric space, the *objective space*. Networks based on this objective metric have been developed previously [Webb 1992; Jain and Mao 1992; Tattersall and Limb 1994; Mao and Jain 1995], and will be reviewed later in Section 3.5.

This 'objective' nomenclature has been chosen deliberately in order to distinguish the conventional spatial (Euclidean) interpretation from what will be referred to as the *subjective metric* and corresponding *subjective space*. The motivation for this dichotomy, and the important distinction between the subjective and objective spaces, will be developed in the remainder of this section.

3.3.1 Class Knowledge

For a given set of data, accompanying the explicit spatial information — perhaps referred to as the input data, the sensor data, the measurement variables or the explanatory variables — there is often additional related information. Probably the most common such form this may take is that of *class labels*, where each data point has an associated label of membership of one of a number of distinct classes.

Now, if one purpose of the topographic mapping is to discriminate between classes or to enhance relevant clusters, then the information provided by class labels may be usefully incorporated. This can be achieved through the mechanism of minimising a modified STRESS measure:

$$E' = \sum_{i}^{N} \sum_{j}^{N} (\delta_{ij} - \parallel \mathbf{y}_i - \mathbf{y}_j \parallel)^2, \tag{3.8}$$

which is identical to the simplified Sammon STRESS with the exception that the inter-point distance in the data space d^*_{ij} is replaced by the variable δ_{ij}. The variable δ_{ij} can incorporate the class information if

$$\delta_{ij} = \begin{cases} d^*_{ij} & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are in the same class,} \\ d^*_{ij} + k & \text{otherwise.} \end{cases} \tag{3.9}$$

Thus, the inter-point distances for pairs of points in different classes are modified by the addition of some constant term k, such that their separation should be exaggerated in the resultant map. An alternative formulation is

$$\delta_{ij} = \begin{cases} 0 & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are in the same class,} \\ d^*_{ij} & \text{otherwise,} \end{cases} \tag{3.10}$$

which tends to enhance clustering of points belonging to identical classes.

These class-based modifications have been incorporated in mapping schemes by Koontz and Fukunaga [1972], Cox and Ferry [1993] and Webb [1995], and will be reviewed further in Section 3.5.

3.3.2 Generalised Knowledge and the Subjective Metric

The use of class labels to enhance clustering as described above is simplistic in that it blindly treats all classes identically. In many problems there may be further knowledge available regarding class relationships, and one particularly convenient mechanism for encapsulating this is within a framework of what may be termed *subjective dissimilarity*. This is best explained by reference to one particular example in the literature [Lowe 1993], once again described in Section 3.5.

In this application, there are 11 distinct classes, representing concentrations of ethanol in water of 0%, 10%, 20%, and so on to 100%. Because these classes are both ordered and 'linear', there is an implied notion of dissimilarity between them, independent of the sensor information associated with each measurement datum. For example, it is natural to consider that in terms of concentration, the 60% class is twice as 'distant', or dissimilar, from the 80% or 40% classes as it is from the 70% or 50% classes. Because it is only the *relative* dissimilarities that are important, such values may be assigned arbitrarily, as long as the relationships previously defined still hold. It would be most intuitive, though, to assign a dissimilarity of 10 to classes 60% and 70%, and a dissimilarity of 20 to classes 60% and 80%. These simple examples may be obviously extended to derive *a value of subjective dissimilarity for every class-pair*.

This assignment of class dissimilarity means that for every pair of data points (assuming they are all labelled), in addition to the *objective dissimilarity*, there is a dual measure of *subjective dissimilarity*. This latter measure will be denoted by s_{ij}, corresponding to each d^*_{ij}.

It should be emphasised that this concept of subjective dissimilarity is not limited to class-labelled data alone, but is intended to embody alternative knowledge in general, particularly where there are no convenient discrete class groupings. For the example of the ethanol/water classes above, rather than the value of concentration being controllable, it may be a variable and so need to be measured during the experiment, in which case it will take on a continuous range of values. In such circumstances, despite the absence of any discrete class groupings, there still exists a natural measure of dissimilarity — the absolute difference between two concentration values. Another example might be in photo-chemistry, where certain measured chemical properties result in a particular colouration response. In this instance, the subjective dissimilarity between data points might be derived as the inter-response distance within the RGB colour cube.

The existence of a set of subjective dissimilarities s_{ij}, consistent with the additional knowledge related to the data, can be naturally interpreted as an alternative *metric* implicitly defined over the input space — the previously introduced *subjective metric*. (Note that for this interpretation to be strictly appropriate, the values of s_{ij} should be consistent with the axioms of equations (2.5)-(2.8) given in Section 2.5 in the previous chapter.) It is this metric that is variably incorporated in the NEUROSCALE model and provides a measure of *supervisory* input.

3.3.3 NEUROSCALE

The NEUROSCALE technique is effected by a feed-forward radial basis function network which transforms the p-dimensional input space into the q-dimensional feature space (generally, $q < p$). As this technique is mainly relevant to the visualisation and exploratory analysis of data, the dimension of the feature space q will generally be 2 or 3. The network is trained by the relative supervision algorithm, outlined in Section 3.2, and minimises the STRESS measure:

$$E_{ns} = \sum_{i}^{N} \sum_{j<i}^{N} (\delta_{ij} - \| \mathbf{y}_i - \mathbf{y}_j \|)^2, \tag{3.11}$$

where

$$\delta_{ij} = (1 - \alpha) d^*_{ij} + \alpha s_{ij}. \tag{3.12}$$

The parameter 'α', where $0 \leq \alpha \leq 1$, therefore controls the degree to which the subjective metric influences the output configuration, and can be considered as defining an interpolation between an unsupervised mapping and a supervised variant.

Thus, from the perspective of the neural network, the input data vectors, the transformation mechanism and the form of the topographic constraint remain identical for all values of α. The relative supervision algorithm of 3.1 is constant, the only alteration to the procedure is to adapt the pre-calculated elements of the input space distance matrix (‘`dStar`’ in the algorithm of figure 3.1), to take account of the particular value of α. Adjustment of that parameter may therefore be interpreted as re-defining the metric over the input space. (It is trivial to see that if the measures d^*_{ij} and s_{ij} are metrics, then δ_{ij} is also.) With $\alpha = 0$, the network is effecting a parameterised Sammon mapping. With $\alpha = 1.0$, the output configuration is no longer explicitly determined by the spatial distribution of the input vectors, but is controlled by the subjective metric alone.

How this latter metric is formulated depends both upon the knowledge of the data, of course, but also on the intended purpose of the mapping process. It may be considered, therefore, that the subjective, or supervisory, element of NEUROSCALE is an expression of *preference* on the topology of the extracted feature space. For example, if clustering is important, then defining intra-class dissimilarities to be zero will emphasise that aspect in the mapping. Alternatively, if a particular inter-class global structure is preferred, that influence may also be applied. Selecting an intermediate value of α will both retain some of the objective (spatial) topology, and impose some measure of preference onto the configuration. That there is real merit in such a hybrid feature space will be demonstrated in the next chapter.

To minimise E_{ns}, various optimisation algorithms were employed, and these are evaluated in Chapter 7. The network weights may be initialised at random, or alternatively, for $\alpha = 0$, may be set such that the initial network outputs are the first two principal components of the data. For $\alpha \neq 0$, the starting configuration can be initialised as the CMDS mapping of the data. However, this procedure requires calculation of the eigenvectors of a $(N \times N)$ matrix, so for large N, it can be more efficient to initialise at random.

The operation of NEUROSCALE may then be summarised by the schematic of figure 3.2 below.

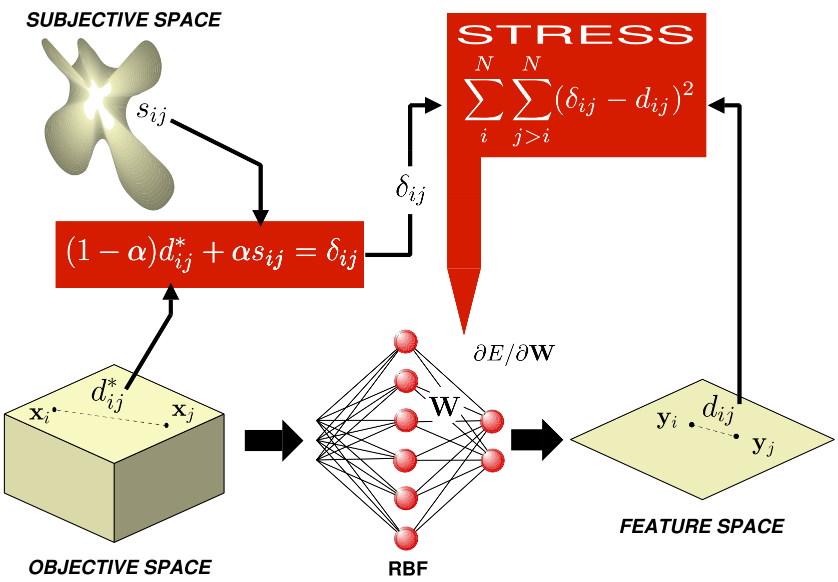


Figure 3.2: A schematic of the operation of NEUROSCALE.

Some of the underlying issues concerning the application of the RBF network — such as the choice of basis functions, local minima behaviour and the effect of optimisation strategy — are considered in Chapters 6 and 7. The following section, however, illustrates the application of NEUROSCALE to some mainly synthetic datasets.

3.4 Examples of Application

3.4.1 The 'Iris' Data

This is a well-known real dataset, used by Fisher [1936] for the development of his linear discriminant function. The data comprises 50 examples of each of three varieties of iris, with each example described by four physical measurements. This data was used by Jain and Mao [1992], and a similar experiment to that reported in their paper can be repeated here. Figure 3.3 illustrates the 2-dimensional feature space generated by NEUROSCALE for 75 patterns chosen from the dataset (25 of each class). Figure 3.4 shows the trained network when applied to the entire 150-pattern dataset, and demonstrates an apparently good generalisation capability. Note that the RBF utilised for the projection comprised 75 basis functions (that is, as many basis functions as patterns), yet, counter-intuitively, there is no explicit evidence of 'over-fitting'. Why this is so is considered in Chapter 6.

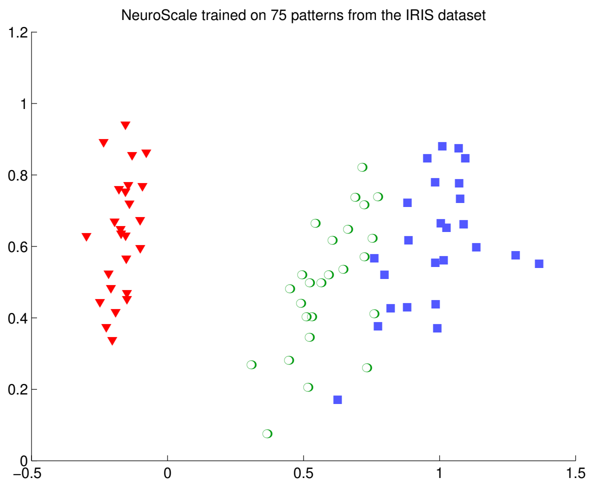


Figure 3.3: The resulting projection when NEUROSCALE is trained on 75 patterns of the Iris dataset. The STRESS for this configuration is 0.00275.

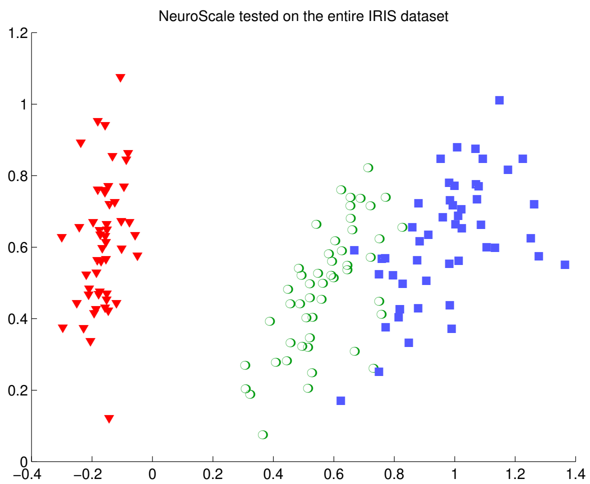


Figure 3.4: The projection when the trained NEUROSCALE network of the previous figure is tested on all 150 patterns of the Iris dataset. The STRESS for this configuration is 0.00325.

3.4.2 Four 'Linear' Gaussian Clusters

This is a synthetic data set, comprising four Gaussian clusters in four dimensions, centred in a line at $(x_c, 0, 0, 0)$, where $x_c \in \{1, 2, 3, 4\}$. The Gaussians have diagonal covariance matrices and the common variance in all dimensions was 0.5. A NEUROSCALE RBF was trained on a subset of the data — the three clusters 1,2 and 4 — and the output configuration is shown in figure 3.5. The trained network was then tested on all four clusters, and the resulting plot given in figure 3.6. This illustrates remarkably excellent generalisation to data that is not sampled from the same distribution as the training set. Again, discussion of this phenomenon may be found in Chapter 6.

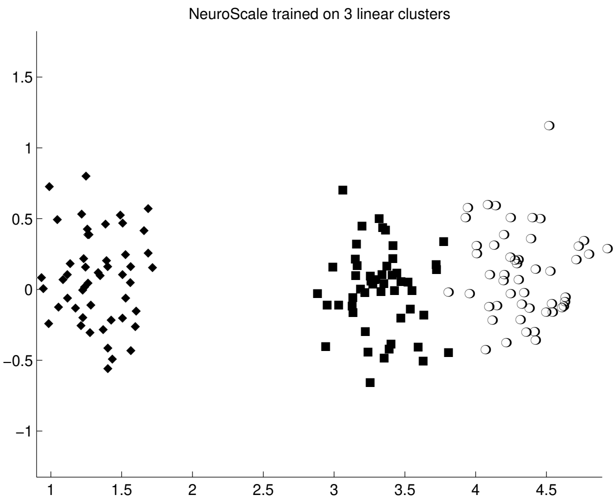


Figure 3.5: The resulting projection when NEUROSCALE is trained on 3 of 4 linear clusters. The STRESS for this configuration is 0.00515.

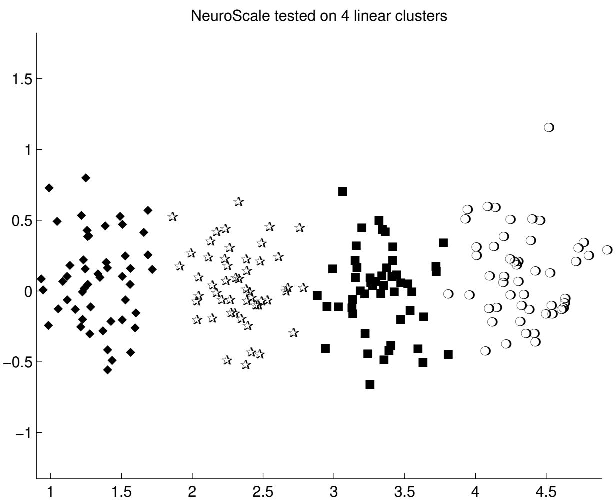


Figure 3.6: The projection when the trained NEUROSCALE network of the previous figure is tested on all 4 clusters. The STRESS for this configuration is 0.00532.

3.4.3 Data on Adjacent Surfaces

For this example, 50 data points were distributed uniformly at random over each of two adjacent surfaces. Each surface was formed by taking a plane of height 5 units and width 2 units, and then curving it through an angle of 30°. The two surfaces were then placed in the input space such that they were parallel and offset by 0.5 units. A cross-sectional illustration of this arrangement is shown in figure 3.7. Figure 3.8 shows the unsupervised ($\alpha = 0$) mapping. With the loss of a dimension under the projection, the minimum STRESS solution requires that both planes are confused, and this behaviour would be likewise exhibited by both principal component and SOFM projections. Figure 3.9, however, gives the projection for $\alpha = 0.5$ where each plane is considered to represent a separate class of points, with the subjective dissimilarity between the two classes set to unity. Incorporation of this additional information now means that the resulting feature space exhibits a good separation between classes *and* additionally retains much of the local topology in each plane. This is emphasised by the two overlaid grids in the plot.

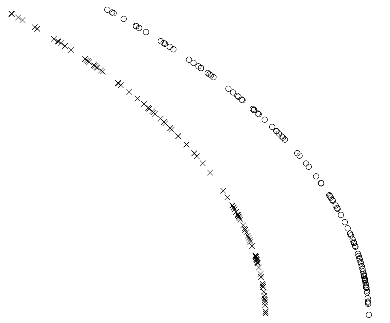


Figure 3.7: Cross section of the two adjacent surfaces.

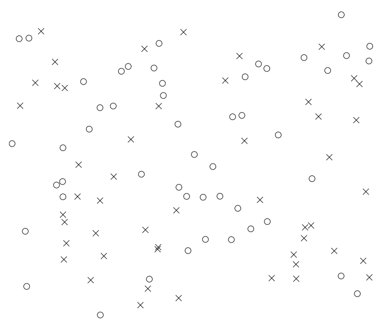


Figure 3.8: An RBF topographic projection of two adjacent surfaces with $\alpha = 0$.

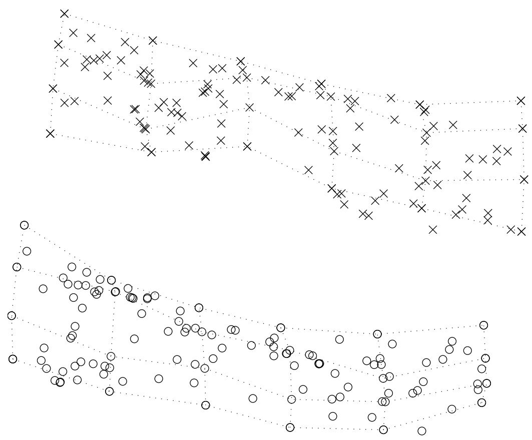


Figure 3.9: An RBF topographic projection of two adjacent surfaces with $\alpha = 0.5$. A grid indicating lines of constant 'height' and 'width' is superimposed.

3.4.4 Data on Three Concentric Spheres

To further illustrate the principle of the NEUROSCALE method, consider the problem of 150 data points in 3-dimensional space, comprising 3 sets of 50 points, each set lying on one of three concentric spheres. All spheres were centred at the origin with radii 0,1 and 2 units respectively and some Gaussian noise added, so that the innermost sphere is effectively a cluster. The data points $\mathbf{x}_i = (x_{i1}, x_{i2}, x_{i3})^{\mathrm{T}}$ were generated by the formula

$$\mathbf{x}_i = (r_k + \nu_i). \begin{bmatrix} \cos\theta_i \sin\phi_i \\ \sin\theta_i \sin\phi_i \\ \cos\phi_i \end{bmatrix} , \tag{3.13}$$

where r_k is the radius ($r_k \in \{0, 1, 2\}$), ν_i is a Gaussian random variable with zero mean and variance 0.05, and θ_i,ϕ_i are uniform random variables in the ranges $[0, 2\pi)$ and $[0, \pi)$ respectively. This collection of points will be referred to as the `SPHERES_3` dataset.

All points on each sphere were considered to belong to a single class and two different schemes for subjective dissimilarities were considered. In the first, each sphere is a distinct class with the subjective dissimilarities simply characterised by the difference in radii. So, the matrix of subjective dissimilarities between spheres is naturally given by

$$\mathbf{C}_1 = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 0 & 1 \\ 2 & 1 & 0 \end{bmatrix} ,$$

where the columns are ordered from the innermost sphere to the outermost sphere. In the second case the innermost and outermost spheres are considered to be the same class, so the matrix becomes

$$\mathbf{C}_2 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} .$$

Values of s_{ij} can therefore be determined for every pair of points, given the knowledge of which spheres they lie on, by referring to one of the above matrices.

The `SPHERES_3` dataset is a problem for which a topographic projection based on a Kohonen network is unsuitable. The unsupervised Kohonen feature map of this data was shown in figure 2.6 in the previous chapter, and illustrates the difficulty of projecting the three distinct surfaces within the data.

A NEUROSCALE transformation was trained for both class models and for values of α of 0, 0.5, 0.75 and 1.0. The resulting projections are given in figures 3.10 and 3.11, for each subjective dissimilarity matrix respectively. These results were obtained using a network with 50 Gaussian basis functions.

The plot for $\alpha = 0$ in figure 3.10, displaying the 'opening out' of the spheres, is characteristic of such structure preserving transformations. The *inter*-sphere distance errors, rather than the *intra*-sphere errors, tend to dominate the STRESS, and these distances are optimally retained by the circular configurations observed. The mapping of a single sphere results in a less 'severe' transformation, as seen in [Webb 1995]. Although no subjective class information has been exploited, there is still a natural separation of the spheres. As α is increased, the spheres are gradually 'folded' until at $\alpha = 1$, the RBF has optimally mapped all the data points in each sphere approximately to a single point. A similar phenomenon is evident in figure 3.11, where the middle sphere is extracted and the other two spheres eventually merged. The combination of both topographic and subjective constraints can be seen in the $\alpha = 0.5$ plot, as some of the spherical structure is still evident.

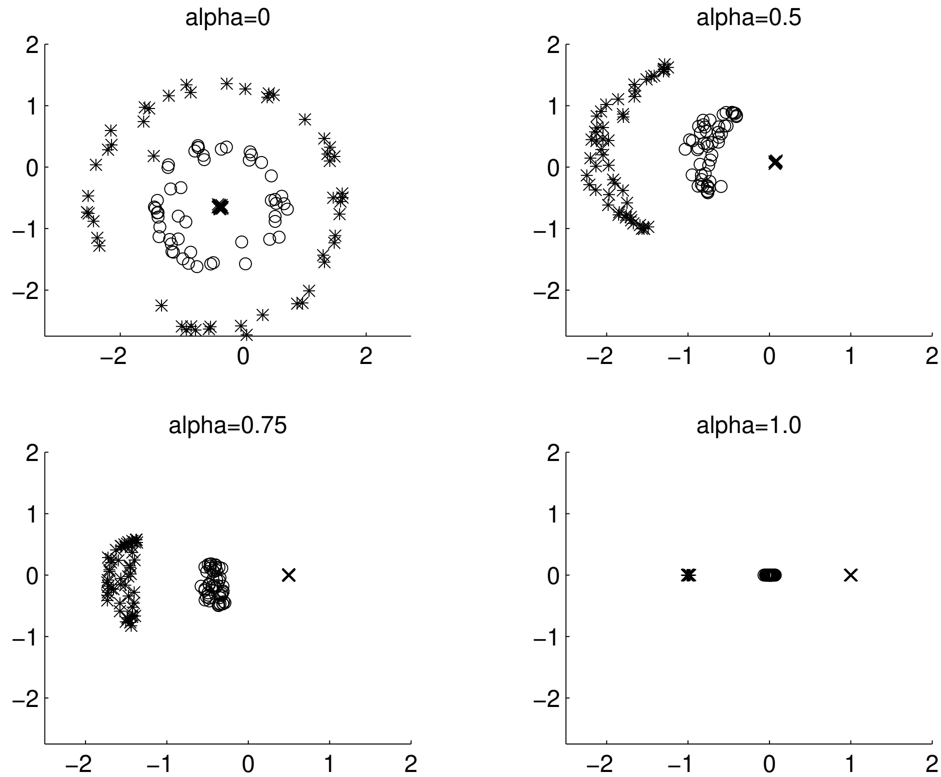


Figure 3.10: Projections of the 3-Spheres data for subjective matrix $\mathbf{C}_1$.

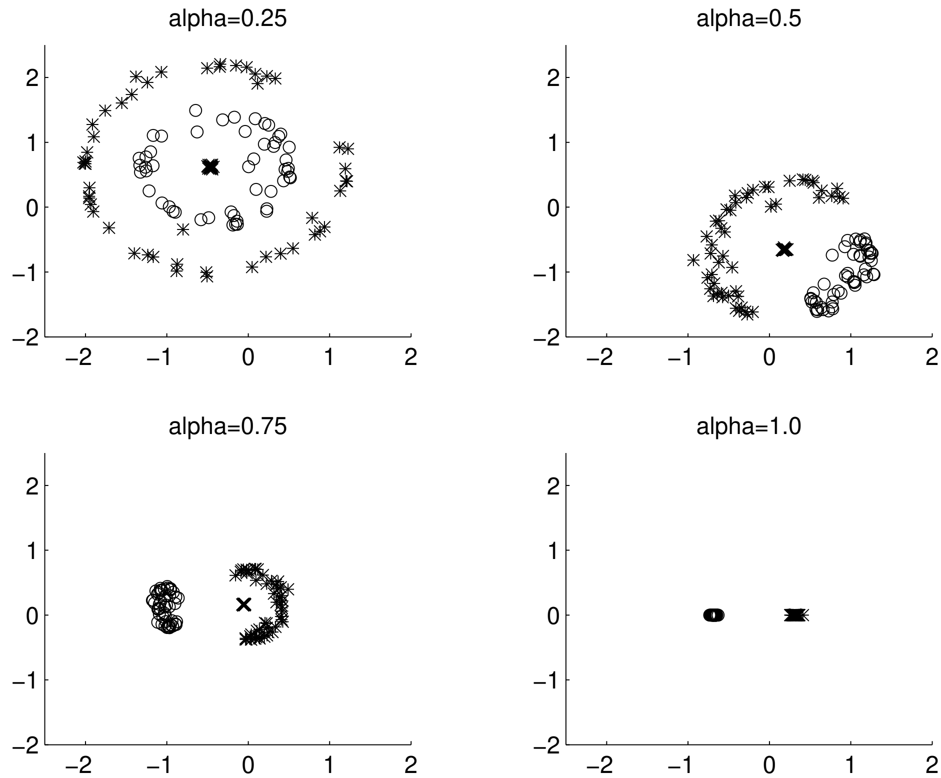


Figure 3.11: Projections of the 3-Spheres data for subjective matrix $\mathbf{C}_2$.

3.5 A Survey of Previous Related Work

The underlying concept of exploiting a neural network model to implement a STRESS-constrained topographic mapping has been suggested independently by more than one researcher. This previous work is summarised in the following three subsections, which group the various approaches into purely topographic mappings, those using binary class dissimilarities and that using a more generalised measure of dissimilarity.

3.5.1 Purely Unsupervised Mappings

Jain and Mao [1992]

The authors originally introduced their model in 1992, applying both 2 and 3 hidden-layer multilayer perceptrons to produce the Sammon mapping, deriving the weight derivatives in a direct fashion, rather than exploiting the derivatives available from standard back-propagation. The output layer neurons were sigmoidal (thus bounding the maximum inter-point distance in the output configuration), so the input patterns had to be normalised *a priori* to presentation to the network. They found the 2-layer network to be the more effective, and gave example projections of the ubiquitous Iris data, including a plot, similar to figure 3.4, illustrating the generalisation capability of the model.

This work was extended in a subsequent journal paper [Mao and Jain 1995], which comprised a survey of feature extraction methods using neural networks. As well as the above Sammon technique, comparison was made with principal components analysis, linear discriminant analysis, the Kohonen SOFM and nonlinear discriminant analysis. These methods were all applied to 4 synthetic and 4 real datasets.

The Sammon model was still implemented by an MLP, with sigmoidal outputs, and was trained by gradient-descent with momentum. A development in this case is that the network is initially trained to produce a PCA projection "because when all the inter-pattern distances in a data set are maximally preserved, the variance of the data is also retained to a very high degree." There is indeed a relationship between variance maximisation and distance preservation, and this is considered in Section 5.2.

One of the key features of this approach is the mechanism to present data to the network. The authors chose to select pairs of patterns at random, and adjust the network weights 'on-line' for each such pair, rather than accumulating weight changes in a 'batch' fashion. The latter method is that exemplified by the algorithm of figure 3.1 earlier. While for large datasets this stochastic approach would appear sensible, it may be seen to be computationally inefficient. To understand why, consider the learning cycle for $N/2$ pattern pairs. This requires N forward and backward propagations through the network, along with N additional STRESS derivative calculations. For the example algorithm of figure 3.1, a similar number of propagations are required to train the network for $N(N-1)/4$ pattern pairs, although an additional $N(N-2)/2$ STRESS derivative calculations are involved. For equivalent numbers of patterns, these latter calculations will be much less computationally expensive than the additional network propagations, so for datasets of a reasonable size, the presented batch algorithm should offer a much better return on computational investment. This will be illustrated more quantitatively in a study of training methods as part of Chapter 7.

Webb [1992]

The concept of a neural network transformation within MDS was introduced by the author in 1992. This approach utilised a two-layer MLP (with linear outputs) incorporated within the standard *non-metric* MDS procedure, and therefore also required the monotonic regression stage. Although the benefits of generalisation to new data were alluded to, no illustration of this capability was given.

Tattersall and Limb [1994]

This again was a two-layer MLP (sigmoid outputs) implementation, deriving the weight adaptation equations in the same fashion as Jain and Mao [1992], and also on an on-line, pattern-pair by pattern-pair, basis. The authors name this approach the "hidden target mapping".

An additional feature within the implementation was the inclusion of a "locality control". Having derived equation (3.7), the denominator d_{ij} was replaced with the term $\lambda d_{ij} + (1 - \lambda)$, where $0 \leq \lambda \leq 1$. The motivation for this is that the implicit weighting in the error measure between larger, global, and smaller, local distances can be controlled. It is noted that "the mapping becomes much more sensitive to errors in mapping points which are close together rather than far apart" because "if two points are close together in the map, d_{ij} is very small and tends to amplify the value of the error derivative."

This assertion is, however, erroneous. The factor d_{ij} may be divided into the term $(\mathbf{y}_i - \mathbf{y}_j)$ to give an expression

$$\frac{\partial E}{\partial \mathbf{y}_i} = -2 \sum_{j \neq i} \left(d^*_{ij} - d_{ij} \right) \hat{\mathbf{r}}_{ij}, \tag{3.14}$$

where $\hat{\mathbf{r}}_{ij}$ is a unit vector in the direction $(\mathbf{y}_i - \mathbf{y}_j)$. The magnitude of this derivative is determined solely on the residual distance error, $d^*_{ij} - d_{ij}$, and is independent of the distance between points i and j in the mapped space.

3.5.2 Simple 'Binary' Mapping of Class-Labelled Data

The most common form of prior knowledge associated with data is that of *class labels*. Each data point $\mathbf{x}_i$ is considered to belong to one of a finite number of classes, usually conveniently labelled with an integer such that the class of point $\mathbf{x}_i$ is given by ω_i.

In applications where topographic mappings are to be employed in the projection of class-labelled data, this information may be exploited in the generation of the projection in order to increment its utility with respect to some classification or clustering criterion. Variations on this approach have been adopted by the following.

Koontz and Fukunaga [1972]

This nonlinear feature extraction procedure, motivated in part by MDS ideas, was developed in 1972. In order to generate mappings with improved class separability in the feature space, the authors optimised a combined criterion incorporating both structural and discriminatory elements:

$$J = J_{SE} + \lambda J_{SP}, \tag{3.15}$$

where J_{SE} is a *separability* criterion, and J_{SP} the usual *structure preservation* measure. (There is a clear parallel with the objective and subjective nomenclature utilised in the description of the NEUROSCALE model earlier.) The constant λ determines the relative contributions towards the STRESS of the two criteria. The structure preservation term is then given by

$$J_{SP} = \sum_i \sum_{j<i} \alpha_{ij} [d^*_{ij} - d_{ij}]^2, \tag{3.16}$$

where α_{ij}, a constant for each point pair, is from standard NMDS and is

$$\alpha_{ij} = \frac{1/d^*_{ij}}{\sum_i \sum_{j<i} d^*_{ij}}. \tag{3.17}$$

The separability term is

$$J_{SE} = \sum_i \sum_{j<i} \delta(\omega_i, \omega_j)\alpha_{ij} d^2_{ij}, \tag{3.18}$$

where $\delta(\omega_i, \omega_j)$ is defined as

$$\delta(\omega_i, \omega_j) = \begin{cases} 0 & \omega_i \neq \omega_j, \\ 1 & \omega_i = \omega_j. \end{cases} \tag{3.19}$$

with ω_i being the class label associated with data point $\mathbf{x}_i$. Hence this term tends to minimise the inter-class scatter by penalising patterns that are in the same class but map to distant points in the output configuration.

The algorithm derived for the projection, the *distance-difference mapping*, was highly heuristic, requiring some expert knowledge and certain assumptions. Nevertheless, it was illustrated how nonlinear transformations based on spatial criteria could be beneficially adapted to include class information.

Cox and Ferry [1993]

These authors also exploited an identical form of class information used above, but in a standard NMDS procedure. The elements of the dissimilarity matrix, $\mathbf{\Delta} = \{\delta_{ij}\}$, were adjusted according to the classes of stimuli i and j, and in this case,

$$\delta_{ij} = \begin{cases} \gamma\delta_{ij} & \text{if} \quad \omega_i \neq \omega_j, \\ \delta_{ij} & \quad\quad \omega_i = \omega_j. \end{cases} \tag{3.20}$$

This embodies an alternative philosophy for discrimination to that adopted by Koontz and Fukunaga. Here, different classes are intended to be more distant in the configuration, rather than identical classes to be more close.

In order to produce a transformational variant of this mapping, a simple linear or quadratic model was fitted to the configuration *a posteriori*, rather than generating that model implicitly in the scaling procedure as incorporated by Webb [1992].

Webb [1995]

This paper represented an extension of work in the earlier paper [Webb 1992], described previously. In contrast to that implementation, the monotonic regression phase was discarded and a radial basis function network was used to effect the transformation, as suggested by Lowe [1993]. A further extension of the procedure was to include a mechanism for discrimination, similar to that employed by Koontz and Fukunaga above. Instead of minimising the standard stress measure, the author employed one of the form

$$J = (1 - \lambda)J_{SE} + \lambda J_{SP}, \tag{3.21}$$

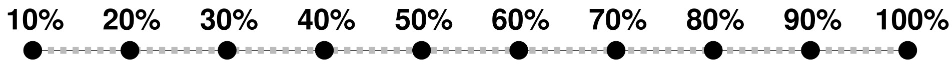
where the two criteria J_{SE} and J_{SP} were those as used by Koontz and Fukunaga. The parameter λ ($0 \leq \lambda \leq 1$) allows a mixing of the two criteria. The hybrid STRESS measure, J, was then minimised via the *iterative majorisation* method, as employed in the popular ALSCAL procedure [Young and Harris 1990].

The discriminance property of the transformation was illustrated for the Iris data, with a contraction of the three classes evident in the projection for $\lambda = 0.1$.

3.5.3 General Mapping of Class-Labelled Data

Lowe [1993]

This paper described the development of an 'artificial nose', employing a relative supervision algorithm to train a radial basis function network performing a topographic transformation. The input data was taken from a number of chemical vapour sensor responses to samples of varying discrete concentrations (0%, 10%, ... , 100%) of ethanol in water. However, rather than optimise the stress of the output configuration in terms of the inter-point distances in the sensor space, the values d^*_{ij} were derived from the alternative knowledge of the data. This is based on the concept of the metric in concentration space, rather than in data space. This metric could be reasonably expected to have the property that points corresponding to samples of 40% concentration would be twice as distant, or dissimilar, from those of 20% and 60% than those of 30% and 50%. This implies an ordering of, and linearity between, samples in concentration space, and may be encapsulated in a simple illustration thus:



The entries in the dissimilarity matrix, δ_{ij} were thus set by

$$\delta_{ij} = |\,\omega_i - \omega_j|, \tag{3.22}$$

where ω_i is the concentration value associated with point $\mathbf{x}_i$. Although input for the RBF was the raw sensor data, in spirit the procedure was more akin to MDS than the Sammon mapping, as the structure of the feature space was based exclusively on the subjective dissimilarities.

Thus a nonlinear mapping was defined from the input data space to a feature space, with the configuration therein constrained by a set of dissimilarity values consistent with the additional knowledge of the concentration data. New data could then be projected onto this derived concentration 'line', such that the network might be considered to be mimicking an interpolating classifier. In that it uses the class information associated with the data, this approach can be considered to be of a *supervised* nature, even though it is effected via a strictly unsupervised topographic mapping procedure.

3.5.4 Relationships with NEUROSCALE

The artificial nose application is a good practical example that illustrates the utility of permitting additional information to influence the topographic mapping procedure. The definition of subjective dissimilarity measures is an appropriate mechanism for encapsulating this knowledge such that it is conveniently assimilated by the mapping algorithm and, importantly, combined in a consistent manner with the spatial metric.

In general, these subjective dissimilarities are assigned according to the knowledge of the problem. Those chosen for the concentration data, and similarly for the research ratings in the next chapter, correspond to intuitive preference. Alternatively, in the true spirit of MDS, the dissimilarities may be those obtained by psychological experiment. In the concentration coding experiment the mapping is determined entirely by this alternative knowledge. In contrast, previous techniques either used spatial input relations alone, or augmented this with some class separability criterion. In analogous manner to the use of the parameter λ in equation 3.21 to combine spatial and separability criteria, a similar parameterisation has been exploited in the NEUROSCALE model to control the influences of objective and subjective knowledge in the ultimate mapping.

3.6 Conclusions

This chapter has introduced the NEUROSCALE model, a neural network approach closely related to MDS methods and the Sammon mapping. Such a scheme is a natural extension of recent work to develop alternative neural network approaches to topographic mapping, and which, importantly, incorporate a capacity for generalisation.

The NEUROSCALE approach views the problem as one of constructing a transformation such that the topography of the transformed space reflects the metric information inherent in both the objective and subjective spaces. The weight ascribed to either of these two criteria may be controlled and will depend on the exact purpose of the map, and in addition, might be influenced by the expected reliability of the additional knowledge.

The concept of the subjective metric is clearly shown in the projections derived from the artificial examples. However, a more effective exposition of the model will be given when applied to a difficult, real-world dataset in Chapter 4, where it will be compared with other established techniques for feature extraction.

To summarise the advantageous features of NEUROSCALE:

- The technique produces a transformation of the data, rather than just a simple mapping as in the case of standard Sammon mapping or MDS models. Thus new data may be projected.
- It permits the incorporation of varying degrees of subjective knowledge which can be allowed to influence the extracted feature space.
- The number of parameters in the non-linear optimisation process scales only with the size of the network, rather than with the number of patterns. This is of particular benefit when employing memory-hungry optimisation routines (such as BFGS). In fact, reduction in training time of some 40% (compared to a standard Sammon mapping) was observed for 200 patterns projected to two dimensions using the conjugate-gradient optimisation routine. Such improvements are more exaggerated as the number of patterns increases.
- Extracted feature spaces are often more 'representative' of the problem than the space extracted by a Kohonen network (e.g. the SPHERES_3 problem).
- Again, in contrast to the SOFM, there is a cost function associated with a particular configuration. This permits the integrity of individual maps to be assessed, and alternatives to be compared.

There nevertheless remain some limitations:

- The computational requirements of the technique still scale with the square of the number of patterns (although the RBF component of the procedure, in terms of the transformation of patterns and calculation of derivatives $\partial \mathbf{y}/\partial w_k$, only scales linearly). This limits the number of training patterns that can be used to produce a transformation.

- Problems of local minima are inherent in nonlinear local optimisation procedures, and STRESS-based mappings have been observed to be particularly bad in this respect. In the case of NEUROSCALE, sub-optimal local minima were not found to be problematic, and further consideration of this limitation forms a significant part of Chapter 6.

- A choice of parameter α is necessary. Appropriate values can only be ascertained on a trial and error basis, and depend upon the preference and knowledge of the user. The effect of a particular value of α is also very much dependent on the order of magnitude of distances in the input space and of the subjective dissimilarities applied, so some scaling of the latter quantities may need to be taken into account.