Topographic Mappings and Feed-Forward Neural Networks

MICHAEL E. TIPPING

Doctor Of Philosophy

THE UNIVERSITY OF ASTON IN BIRMINGHAM

February 1996

THE UNIVERSITY OF ASTON IN BIRMINGHAM

Topographic Mappings and Feed-Forward Neural Networks

MICHAEL E. TIPPING

Doctor Of Philosophy, 1996

Thesis Summary

This thesis is a study of the generation of topographic mappings — dimension reducing transformations of data that preserve some element of geometric structure — with feed-forward neural networks.

As an alternative to established methods, a transformational variant of Sammon's method is proposed, where the projection is effected by a radial basis function neural network. This approach is related to the statistical field of multidimensional scaling, and from that the concept of a 'subjective metric' is defined, which permits the exploitation of additional prior knowledge concerning the data in the mapping process. This then enables the generation of more appropriate feature spaces for the purposes of enhanced visualisation or subsequent classification.

A comparison with established methods for feature extraction is given for data taken from the 1992 Research Assessment Exercise for higher educational institutions in the United Kingdom. This is a difficult high-dimensional dataset, and illustrates well the benefit of the new topographic technique.

A generalisation of the proposed model is considered for implementation of the classical multidimensional scaling (CMDS) routine. This is related to Oja's principal subspace neural network, whose learning rule is shown to descend the error surface of the proposed CMDS model.

Some of the technical issues concerning the design and training of topographic neural networks are investigated. It is shown that neural network models can be less sensitive to entrapment in the sub-optimal global minima that badly affect the standard Sammon algorithm, and tend to exhibit good generalisation as a result of implicit weight decay in the training process. It is further argued that for ideal structure retention, the network transformation should be perfectly smooth for all inter-data directions in input space.

Finally, there is a critique of optimisation techniques for topographic mappings, and a new training algorithm is proposed. A convergence proof is given, and the method is shown to produce lower-error mappings more rapidly than previous algorithms.

Keywords: Information Processing, Feature Extraction, Sammon Mapping, Multidimensional Scaling, Research Assessment Exercise

Contents

List of Figures

Chapter 1

Introduction

> Where is the knowledge we have lost in information?
>
> **T.S. Eliot** — *The Rock* (1934).

It is often said that we are living in the *information age*. The technological revolution of the latter half of the twentieth century has placed previously undreamt-of quantities of information at our fingertips. The maturity of the digital computer with its continual exponential growth in both power and storage capacity, allied with the emergence of multi-media and the dramatic recent expansion of global connectivity known rather grandiloquently as the 'digital information super-highway', offers unprecedented access to vast amounts of data, all over the world, for millions of users.

However, as the ease of access to information increases, so, inevitably, do the accompanying difficulties in its interpretation and understanding. It is very easy to become overwhelmed by the sheer volume available. One particular on-line information resource is the data collected for the 1992 Research Assessment Exercise for higher educational institutions in the United Kingdom. Even the small fraction of this large dataset that is studied later in this thesis contains over thirty-two thousand numbers and there is clearly little to be gained by study of the naked data alone; the knowledge remains locked away, impenetrably it may sometimes seem, behind the anonymous digits.

The key, therefore, lies in *information processing*. Whether for the purposes of visualisation, exploratory analysis or for subsequent computation, it is essential that the information be manipulated into a form which facilitates its ultimate use. The emphasis has thus shifted from the problem of the acquisition of information, to that of its exploitation for the purposes of deriving useful knowledge.

The type of information, or data, that will be considered in this thesis is that in *numeric* form. Data will, characteristically, be comprised of a set of measurements concerning a corresponding set of objects. For example, Fisher's familiar 'Iris' dataset contains measurements of sepal length, sepal width, petal length and petal width for fifty samples of each of three different varieties of iris flower. The previously mentioned Research Assessment data comprises nearly one-hundred-and-fifty different variables for over four thousand invidual departments from every university in the United Kingdom — variables which describe such quantities as the number of staff, the number of Ph.D. students and the number and value of research grants. This numeric form lends itself naturally to a vector-space interpretation, such that in the Iris dataset, each set of four sample measurements can be considered a distinct vector in four-dimensional space. In general, then, for all such datasets with p different fields, the data may be considered as a collection of similar point vectors in a p-dimensional space.

Given this interpretation, information processing can often be intuitively posited as a *dimensionality*

reduction problem. For visualisation, human perception is attuned to two- or three-dimensional images, and real-world numeric data, which is generally of naturally high dimension, must be processed into more readable forms without loss of salient detail. In data-modelling applications, the sizable number of variables implied by high-dimensional data can be seriously disadvantageous. Sensible pre-processing of the data before the model-building stage can help alleviate these problems.

This thesis concerns one particular approach to extracting knowledge that is concealed within information. It is an investigation into the use of feed-forward neural networks to effect a particular class of dimension-reducing information-processing strategies — *topographic mappings*. Exactly what a topographic mapping is, why a neural network should be used to produce one and what is the relevant contribution of this thesis, are questions considered during the remainder of this introduction.

1.1 What is a Topographic Mapping?

Topographic mappings are a class of data-processing mechanisms which seek to preserve some notion of the *geometric structure* of the data within the reduced-dimensional representation. The term 'geometric structure' will be used in this thesis in the sense that *distance* relationships are important, so that points that lie close together in the data space will appear similarly close together in the map[1], and equally, under certain interpretations, points that are more distant in data space will, after mapping, remain likewise separated.

This latter question of interpretation exemplifies that, in practice, there may be alternative emphases placed on the nature of the structure preservation. One emphasis is that *all* distance relationships between data points are important, which implies a desire for global *isometry* between the data space and the map space. Alternatively, it may only be considered important that *neighbourhood relationships* are maintained, such that points that originally lie close together are likewise preserved in the map, and this is referred to as *topological ordering*.

While the word 'topological' is often used in certain contexts as a substitute for 'topographic', it is important to make the distinction between the distance-based criteria considered in this thesis and the notion of *topological invariance* in its strictly mathematical sense. Indeed, "spaces which appear quite different — geometrically for instance — may still be topologically equivalent." [Gamelin and Greene 1983]. In this thesis, 'topographic' will be considered synonymous with 'geometric', in that it is desired that all distance relationships be preserved in the mapping.

Perhaps the most intuitive, and certainly the most literal, example that may be given of a topographic map is that of the projection of the naturally spherical surface of the Earth down onto a two dimensional plane. Such a projection is shown in figure 1.1 below.

This simple illustration also serves to demonstrate an important principle — that when data undergoes a reduction in dimension, some structure is inevitably lost. In practical applications this is an important point, as for high-dimensional datasets the map will normally be of a much lower dimension compared to the original data, and this dimensional imbalance tends to accentuate that problem. In order to represent the topography of the surface of a three-dimensional globe on a two-dimensional plane in figure 1.1, it is necessary to introduce some distortion. While much structure is still retained — consider the interior geography of Europe, for example — at extremes of latitude distances in the map are considerably exaggerated, and even more severely, the left and right longitudinal edges of the map have been drastically separated. Hence the development of various alternatives to Mercator's technique within the field of cartography, such as the Cylindrical Equal Area and Peters' projections, with each introducing its own particular class of distortion.

[1] Throughout this thesis, the word "map" will be used in its intuitive visual sense to refer to the image of the mapping process, rather than in its mathematical sense, as a synonym for transformation.

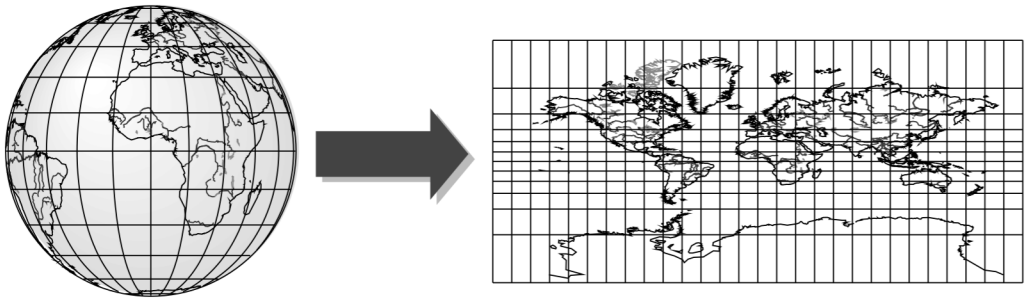

Figure 1.1: A Mercator's projection of the spherical Earth down to a two-dimensional map.

As is evident from the geographical example above, topographic maps can be highly valuable as tools for *visualisation* and *data analysis*. Structure-retaining maps can generally be interpreted quite intuitively, and, as will be seen later, often much more so than other reduced-dimension representations. Many important relationships between the data points can be inferred by viewing the map — notably the detection of *clusters*, or sets of points closely grouped in the data space and which should be similarly adjacent in the projection. However, as will be discussed later in this thesis, under certain conditions, apparent structure exhibited in a map may in fact be *artefactual*, and not be representative of the true geometry in data space. The potential for such phenomena should always be borne in mind when interpreting topographic mappings.

1.2 Why Use a Feed-Forward Neural Network?

There are already some well-established methods for topographic mapping. From the domain of engineering, there is the *Sammon mapping*, or *Nonlinear Mapping*, [Sammon 1969] which is closely related to some of the techniques from the statistical field of *multidimensional scaling* [Davison 1983]. While still in popular use, both approaches possess several inherent disadvantages, the most significant being that when a map has been generated, it effectively acts as a *look-up table* such that there is no potential for projecting new, previously unseen, data. Importantly, this implies that there is no facility for *generalisation*, a principal feature of neural networks and one which, after a given network has been trained, enables prospective inferences to be drawn and predictions to be made concerning new data.

There is also an existing neural network architecture designed specifically for topographic mapping, and that is Kohonen's ubiquitous *self-organising feature map* [Kohonen 1995], which exploits implicit lateral connectivity in the output layer of neurons. This neuro-biologically inspired scheme, however, also exhibits several disadvantages and this thesis will propose an alternative paradigm which exploits the standard feed-forward network architectures.

Feed-forward neural networks are now well established as tools for many information-processing tasks — regression, function approximation, time series prediction, nonlinear dimension-reduction, clustering and classification are examples of the diverse range of applications. (See [Haykin 1994] for a comprehensive coverage.) Divorced from their neuro-biological foundation, the major attraction of neural network models is that certain classes thereof have been shown to be *universal function approximators*, such that they are capable of modelling any continuous function over a bounded domain, given sufficient network complexity. This property implies that, given appropriate design and training, neural networks can be employed as *semi-parametric* models, and thus require fewer prior assumptions about the underlying relationships in the data.

It would be attractive, then, to generate topographic mappings using such architectures. That is, the function that *transforms* the vectors in the data space to a corresponding set of image vectors in the map will be effected by a feed-forward neural network. This concept is illustrated in figure 1.2.

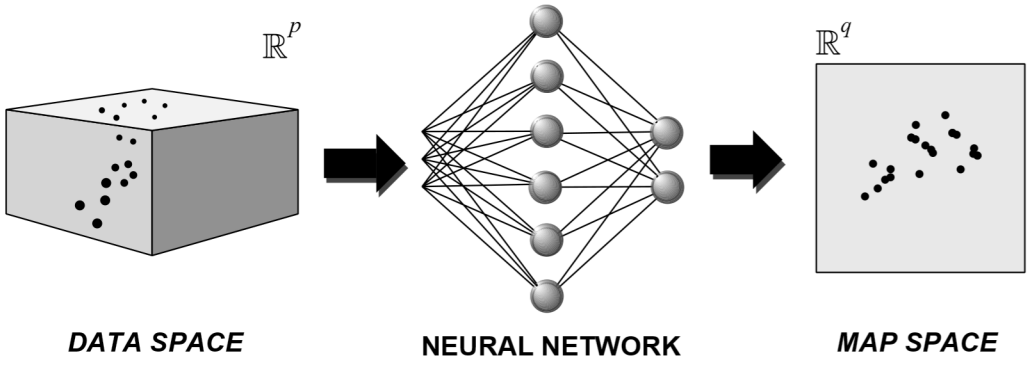

Figure 1.2: A neural network effecting a topographic transformation.

On initial consideration, the training of such a topographic transformation might appear problematic. In the majority of neural network applications, for example regression or classification, there are a set of *target* vectors, corresponding to the set of input vectors — effectively a set of *desired* outputs that the network is trained to reproduce. This scenario is a referred to as a *supervised* problem. In the *unsupervised* topographic case, for each input datum there is no such specific target information, and alternative training algorithms must be developed, based on structural (distance) constraints.

The specific neural network model introduced in this thesis, for reasons of textual brevity, will be known as 'NEUROSCALE', as it is a neural network 'scaling' procedure. NEUROSCALE utilises a *radial basis function neural network* (RBF) [Broomhead and Lowe 1988; Lowe 1995] to transform the p-dimensional input vector to the q-dimensional output vector, where, in general, $p > q$. An RBF comprises a single hidden layer of h neurons, as exhibited by the network in figure 1.2, which represents a set of *basis functions*, each of which has a *centre* located at some point in the input space. The number of such functions is generally chosen to be fewer than the number of data points, and their corresponding centres are initially distributed (and are generally fixed) amongst the data, such that their distribution approximates that of the data points themselves. The output of each hidden node for a given input vector is then calculated as some function (e.g. Gaussian) of the distance from the data point to the centre of the function. In this way the basis functions are *radially symmetric*. The output of the network is then calculated as a weighted, linear summation of the hidden nodes, which for supervised problems with sum-of-squares error functions, permits the weights to be trained by standard linear algebraic methods [Strang 1988]. So mathematically, for a p-dimensional input vector $\mathbf{x} = (x_1, x_2, \ldots, x_p)$, the q-dimensional output vector $\mathbf{y} = (y_1, y_2, \ldots, y_q)$ is given by:

$$y_i = \sum_{j=1}^{h} w_{ij} \phi_j(\| \mathbf{x} - \boldsymbol{\mu}_j \|), \tag{1.1}$$

where $\phi_j(\cdot)$ is the j^{th} basis function with centre $\boldsymbol{\mu}_j$, and w_{ij} is the weight from that basis function to output node i. An important result concerning this particular type of network is that it is capable of *universal approximation* [Park and Sandberg 1991].

The training algorithm for the RBF constrains vectors in the output space to be located such that they preserve, as optimally as possible, the distance relationships between their corresponding vectors in the input space. This is in contrast to Kohonen's approach, in which the distribution of the output vectors is approximately representative of the data *density*. This can be one of the disadvantages of the latter approach, particularly in applications where global relationships are considered important.

A further important feature of the NEUROSCALE approach to topographic mapping is the inclusion of a unique mechanism for incorporating *preferential* information. This enables additional knowledge concerning the data (for example class labels or other relevant measurements) to be exploited for the purposes of enhancing clustering, improving group separation or even to impose some additional global ordering upon those groups. Such a facility can be considered as adding a *supervisory* component to the otherwise unsupervised feature extraction process, and this interpretation provides an appropriate basis for comparison with other established information-processing paradigms.

That this supervisory mechanism is of tangible benefit, and that NEUROSCALE in general is an effective tool for the exploratory analysis of data, will be shown in an application to one particularly complex dataset. The data in question is taken from the *1992 Research Assessment Exercise* for higher educational institutions in the United Kingdom and is typical of real-world datasets. The data itself is high-dimensional and polluted by noise, and there is additional information available in terms of a class label ("research rating") that is biased by the subjective opinion of an assessment panel. Nevertheless, this extra knowledge will be exploited to generate improved visualisation spaces which can be used as a basis for subsequent prediction of unclassified data.

The NEUROSCALE approach as detailed in this thesis is an incremental development of recent research effort directed at exploiting neural networks to perform structure-retaining mappings. The author is unaware of any significant theoretical investigation into the training and application of such models, and a considerable portion of this thesis is devoted to such detailed analysis.

1.3 Plan of This Thesis

Chapter 1 is this introduction.

Chapter 2 will describe standard approaches to topographic mapping — Kohonen's self-organising feature map, the Sammon mapping and multidimensional scaling — and consider the key distinctions between the three, along with their respective advantages and disadvantages.

Chapter 3 introduces the NEUROSCALE model and relates it to previous work, giving examples of its application to various datasets. These illustrate both the topographic property of the neural network transformation and the facility to exploit additional knowledge.

Chapter 4 is a detailed study of data taken from the 1992 Research Assessment Exercise. Data from the subject areas of physics, chemistry and biological sciences is analysed, both by NEUROSCALE and by other established feature extraction techniques. The emphasis of this chapter is on the visualisation and exploratory analysis of the high-dimensional data, but there are additional results presented for classification experiments, including the use of NEUROSCALE as a pre-processor in prediction models.

Chapter 5 describes a generalisation of the NEUROSCALE approach to classical multidimensional scaling. This is closely related to other neural networks specifically designed for generating principal component projections, notably Oja's principal subspace network, and the parallels are analysed.

Chapter 6 is a study of some of the underlying theoretical aspects of training neural networks to effect topographic mappings. The problem of local minima is considered, and the dynamics of the relative supervision learning algorithm investigated. Analysis is presented concerning the necessary

form and smoothness for topographic transformations which is highly relevant to the question of generalisation.

Chapter 7 considers the optimisation of topographic transformations. Standard techniques are compared, and alternative heuristic strategies also reviewed. An efficient new training algorithm for networks linear in their weights is presented, and its properties studied.

Chapter 8 concludes the thesis with a summary of the significant results therein and suggests directions for future research.

The content of this thesis represents original research. The work within has not previously appeared elsewhere, with the exception of those research papers produced during the normal course of its preparation. Material from Chapters 3 and 4 has appeared in [Lowe and Tipping 1995; Lowe and Tipping 1996], while a paper based on Chapter 5 has been submitted for future publication [Tipping 1996].

1.4 Notation

In general, throughout this thesis, the notation below in table 1.1 will be adopted:

Symbol	Meaning
N	The number of data points
p	The dimension of input space
q	The dimension of the map, or feature, space
h	The number of hidden units in a neural network
$\mathbf{x}_i$	A point vector in the input space
$\mathbf{X}$	The matrix of row-vector input points, $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)^{\mathbf{T}}$
$\mathbf{y}_i$	A point vector in the feature space
$\mathbf{Y}$	The matrix of row-vector mapped points, $(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N)^{\mathbf{T}}$
$\mathbf{A}^{\mathbf{T}}$	The transpose of matrix (or vector) $\mathbf{A}$
tr $[\mathbf{A}]$	The trace of matrix $\mathbf{A}$
$\lvert\mathbf{A}\rvert$	The determinant of matrix $\mathbf{A}$
$\lVert\mathbf{v}\rVert$	The (L_2) norm of vector $\mathbf{v}$
$\mathbf{u}_k$	The k-th eigenvector of some matrix
λ_k	The corresponding eigenvalue
①,②, ...	A numbered list of items
❶,❷, ...	A sequential algorithm

Table 1.1: Notation

Chapter 2

Established Techniques for Topographic Mapping

2.1 Introduction

This chapter considers three particular established schemes for the generation of mappings that preserve some notion of topography or geometric structure — the *Kohonen Self-Organising Feature Map* (Section 2.2), the *Sammon Mapping* (2.3) and the statistical field of *Multidimensional Scaling* (2.5). Each of these approaches is individually described, comparisons between them are drawn and respective advantages and disadvantages outlined.

2.2 The Kohonen Self-Organising Feature Map

The archetypal topographic neural network is Kohonen's self-organising feature map (often simply referred to as the 'Kohonen Map' or abbreviated to 'SOFM') [Kohonen 1982; Kohonen 1990; Kohonen 1995]. The motivation for Kohonen's model is neuro-biological and was developed as an abstraction of earlier work in the field of ordered neural connections by Willshaw and von der Malsburg [1976].

The SOFM can be viewed as a neural network comprising a set of input neurons and a set of output neurons, each of which is connected by a weight vector $\mathbf{w}$, in the standard manner, to the input. However, in contrast to the standard single-layer model — the simple perceptron — there is an inherent additional structure within the output layer. These neurons may be considered to form a fixed *lattice*, usually one- or two-dimensional, with associated lateral connectivity in addition to the connection to the input layer. In the most common two-dimensional case, the output of the SOFM network is a 'sheet' of interconnected neurons in a rectangular or hexagonal configuration. Such an architecture is illustrated in figure 2.1.

When successfully trained, such a network will exhibit the property that adjacent neurons in this lattice structure respond to similar (nearby) input vectors, or features, and the map is then said to be *topologically ordered*. This ordered mode of neural response has been observed on the neocortex in the brains of higher animals, notably the auditory [Suga and O'Neill 1979], the visual [Blasdel and Salama 1986] and somatosensory [Kaas et al. 1979] cortices. For example, on the human auditory cortex, there is a near-logarithmic frequency ordering of responsive cells — this is the so-called *tonotopic map* — such that nearby neurons on the cortex respond to sounds of a similar pitch. This, and

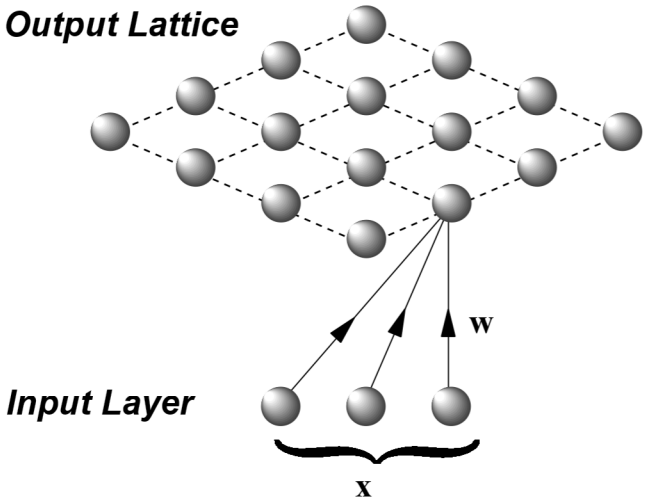

Figure 2.1: A schematic of the architecture of a two-dimensional output layer Kohonen network. For clarity, only the weight connections from the input to a single neuron are shown.

other mappings within the brain, involve a vast number of similar cortical connections (estimated at around 10^{13}), and this almost certainly precludes the possibility that this topological ordering is genetically determined, thus suggesting that these properties evolve during brain development according to some alternative systematic process. The learning procedure for the Kohonen SOFM is a generalised abstraction of such a potential mechanism, and the model has been successfully applied across a considerable variety of distinctly non-biological domains. Good examples include speech recognition, image processing, interpretation of EEG traces and robot arm control, and these, and other applications, are comprehensively reviewed (with references) in [Kohonen 1995]. In addition, there is a large on-line biography ($>$ 1630 references) available concerning theory and applications of the self-organising map [Anonymous 1996].

To construct the SOFM, consider a network as outlined above with inputs from some p-dimensional space connected to a q-dimensional output lattice of K neurons with associated weight vectors $\mathbf{w}_i$, each of which effectively defines a point in the input space. The Kohonen algorithm is thus:

❶ Choose the dimension, size and topology of the map according to the prior knowledge of the problem. Some preprocessing of the data may also be necessary as the map is sensitive to scaling of the input features.

❷ At time step $t = 0$ initialise all the weight vectors $\mathbf{w}_i$ to random values.

❸ Present an input pattern $\mathbf{x}_t$ to the network, drawn according to the input distribution defined by the probability density function $f(\mathbf{x})$.

❹ Determine the "winning" neuron, $v(\mathbf{x}_t)$, whose weight vector $\mathbf{w}_i$ is closest to the input point $\mathbf{x}_t$. That is

$$v(\mathbf{x}_t) = \underset{i}{\operatorname{argmin}} \| \mathbf{w}_i - \mathbf{x}_t \|$$

❺ Adjust the weight vector of the winning neuron, and those of its neighbours, in a direction towards the input vector. That is

$$\mathbf{w}_i = \mathbf{w}_i + \eta(t)\Lambda[i, v(\mathbf{x}_t)](\mathbf{x}_t - \mathbf{w}_i),$$

where $\eta(t)$ is a learning-rate parameter. The function $\Lambda[i, v(\mathbf{x}_t)]$ is the *neighbourhood* function, and is described in detail below.

❻ Repeat from step ❸ until the map has stabilised.

The key to the topographic nature of the mapping is the neighbourhood function, $\Lambda(i, v(\mathbf{x}))$. This is some function defined over the output lattice space which is generally non-negative and decreases with the distance (in the lattice space) between the winning neuron $v(\mathbf{x})$ and any other neuron i. This implies that the weight vector of the winning neuron receives maximum perturbation, while the corresponding vectors of more distant neurons are adjusted to a lesser extent. Popular choices for this function are the Gaussian, $\Lambda(r) = \exp(-r^2/2\sigma^2)$ — where r is the distance from the winning neuron to another neuron — and the 'bubble', which is simply a constant value over a fixed neighbourhood width σ. The parameter σ therefore controls the degree of weight adjustment with distance, a property that is sometimes referred to as the "stiffness" of the lattice.

Thus, when an input pattern is presented to the network, the nearest, winning, neuron will be moved in the direction of that input vector along with its neighbours by an amount that decreases with their distance (within the *lattice*) from the winning neuron. In this way, the weights for nearby neurons will converge to the same region of input space, thus exhibiting the characteristic topological ordering. The 'width' of this neighbourhood function, the parameter σ, is t-dependent. It is usually set to a relatively high value (as much as half the lattice width or greater) at initialisation, and decreases with time. This allows the coarse global structure of the map to be formed in the early stages of training, while the local structure is fine-tuned later. The learning-rate $\eta(t)$ also varies with time, decreasing monotonically to some arbitrarily small value when the map is "frozen".

Three example plots taken during the training of a SOFM are illustrated in Figure 2.2 below. These show the evolution of the output lattice for input data sampled uniformly at random from within a 2-dimensional unit square. The weight vectors $\mathbf{w}_i$ are plotted in the input space, and those corresponding to adjacent nodes in the neuron lattice are shown connected. Note that these connections are not explicit within the SOFM architecture; it is the action of the neighbourhood function that implicitly inter-connects all the output layer neurons to some degree.

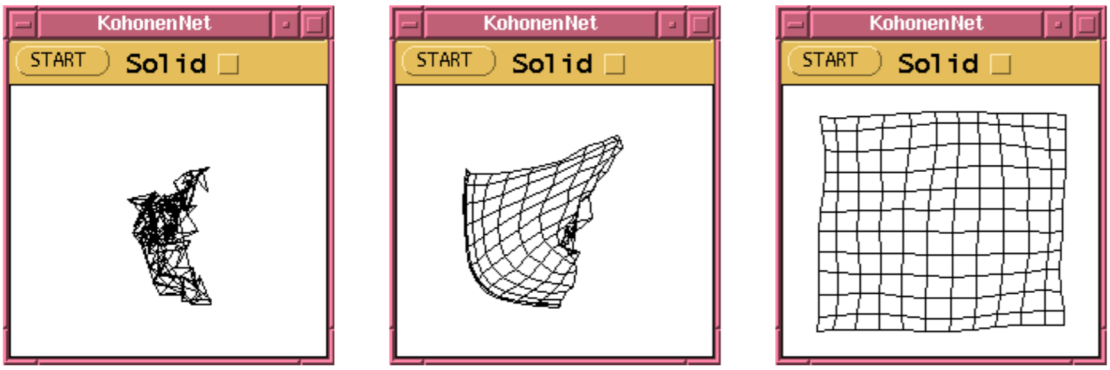

Figure 2.2: A 2-D Kohonen map of data sampled uniformly at random from the unit square. The final map is after 5,000 time steps.

For a neighbourhood function of zero width, there is no topological ordering in the map and the algorithm becomes equivalent to a *vector quantisation* (VQ) scheme (specifically, the LBG algorithm of Linde, Buzo, and Gray [1980]), where the weights $\mathbf{w}_i$ are analogous to the codebook vectors of VQ. It is also, therefore, closely related to the *k-means* technique for data clustering [MacQueen 1967].

Ideally, it would be convenient if the *density* of the distribution of the $\mathbf{w}_i$ in the input space were directly representative of the input probability density $f(\mathbf{x})$. This, however, is not the case. An exact result has been derived only for the single-dimensional feature map which reveals that the density of

the $\mathbf{w}_i$, $m(\mathbf{w})$, is in fact given by [Ritter and Schulten 1986; Ritter 1991]:

$$m(\mathbf{w}) \propto f(\mathbf{x})^{\beta}, \quad \text{with} \tag{2.1}$$

$$\beta = \frac{2}{3} - \frac{1}{3[\sigma^2 + (\sigma + 1)^2]}, \tag{2.2}$$

where σ is the number of neurons to each side of the winning neuron that are adjusted at each training step. For $\sigma = 0$, this is equivalent to VQ, and $m(\mathbf{w}) \propto f(\mathbf{x})^{1/3}$.

In this case, and in general for higher dimensions, the Kohonen SOFM over-emphasises regions of low input density at the cost of under-emphasising those of high density. An illustration of this effect may be seen later in Section 2.4.

One important feature of the SOFM is that it exists only at the algorithmic level. It has been shown [Erwin, Obermayer, and Schulten 1992] that the above procedural description of the Kohonen Map cannot be interpreted as minimising a single energy (or error) function. This implies that there is no direct measure of "quality" of a map, although several indirect alternatives have been proposed [Bauer and Pawelzik 1992; Bezdek and Pal 1995; Goodhill, Finch, and Sejnowski 1995].

There have been several extensions made to the basic SOFM model since its introduction. For application to classification problems, there are the *Learning Vector Quantisation* (LVQ) schemes [Kohonen 1990], where sets of weight vectors are allocated exclusively to a single class and the learning algorithm adjusted such that inter-class decision boundaries are emphasised. There have also been variants of the map proposed which permit arbitrary and dynamic output layer topology, such as the *neural gas* of Martinetz and Schulten [1991] and the *growing cell structures* of Fritzke [1994]. Such schemes permit a better match between the network topology and that of the data distribution, but considerably complicate the generation of convenient visualisations, such as that illustrated for the standard SOFM in Section 2.4. This restriction makes these approaches less suitable for data analysis, and they will not be considered further in this thesis.

2.3 The Sammon Mapping

If the definition of a topographic mapping is to be understood as implying a retention of global metric relationships, then the *Sammon Mapping* [Sammon 1969], sometimes referred to as the *Non-Linear Mapping* or NLM, is the most intuitive basis for its definition. In contrast to the Kohonen mapping, the Sammon mapping may be determined by the optimisation of an error, or 'STRESS', measure which attempts to preserve all inter-point distances under the projection. The Sammon STRESS is defined as

$$E_{ss} = \frac{1}{\sum_i \sum_{j<i} d^*_{ij}} \sum_i \sum_{j<i} \frac{[d^*_{ij} - d_{ij}]^2}{d^*_{ij}}, \tag{2.3}$$

where d^*_{ij} is the distance $\| \mathbf{x}_i - \mathbf{x}_j \|$ between points i and j in the input space $\mathbb{R}^p$, and d_{ij} is the distance $\| \mathbf{y}_i - \mathbf{y}_j \|$ between their images in the map, or feature, space $\mathbb{R}^q$. These distance measures are generally Euclidean but need not strictly be so.

The $[d^*_{ij} - d_{ij}]^2$ term is clearly a measure of the deviation between corresponding distances, and the Sammon STRESS thus represents an optimal, in the least-squares sense, matching of inter-point distances in the input and map spaces. The first fractional term in the expression is a normalising constant which reduces the sensitivity of the measure to the number of input points and their scaling. The inclusion of the d^*_{ij} term in the denominator of the sum serves to moderate the domination of errors in large distances over those in smaller distances, and renders the overall measure dimension-less. The inclusion of this term is not justified in the original paper, and has the effect of making the mapping more sensitive to absolute (though not proportional) errors in local distances.

Given this STRESS measure E_{ss}, it is straightforward to differentiate with respect to the mapped coordinates $\mathbf{y}_i$ and optimise the map using standard error-minimisation methods. Setting the constant $c = \sum_i \sum_{j<i} d^*_{ij}$ for simplification gives

$$\frac{\partial E_{ss}}{\partial \mathbf{y}_i} = \frac{-2}{c} \sum_j \frac{(d^*_{ij} - d_{ij})}{d^*_{ij} d_{ij}} (\mathbf{y}_i - \mathbf{y}_j). \tag{2.4}$$

All the points $\mathbf{y}_i$ in the configuration can thus be simultaneously iteratively adjusted to minimise the error. It should be noted that each partial derivative requires N cycles of computation and therefore to calculate the entire set of derivatives will require a double sum over the data. (In fact, $N(N-1)/2$ loops.) Sammon used a simple gradient-descent technique in his original paper, but less naive methods may be employed, and a *conjugate-gradient* routine [Press, Teukolsky, Vetterling, and Flannery 1992] was found to be considerably more effective.

The Sammon mapping originated in the engineering field and was designed as a computational tool for data structure analysis and for visualisation, and indeed, its use is still popular in many domains — Domine et al. [1993] provide a good review of applications in the field of chemometrics. Feature space dimension, q, is thus naturally chosen as either 2 or 3. Because of the metric nature of the map, clusters of data points tend to be retained under the projection and are manifest in the feature space. In addition to this *local* clustering structure, the inter-cluster *global* relationships are also preserved to some extent. Sammon emphasised this latter feature in the paper, giving several illustrative examples where a linear projection onto the first two principal axes (the orthogonal axes that maximise the variance under projection) confused multiple distinct clusters in contrast to the Sammon mapping which maintained their separation.

While minimisation of E_{ss} implies preservation of the input geometry, the extent to which the integrity of the structure of the input space can be retained is dependent both upon the intrinsic dimensionality of the data, and also upon its topology. In the process of dimension reduction, some information, in all but the most degenerate cases, will be lost, and furthermore, apparent structure may be elucidated which is truly artefactual in nature. A minor example of such structure will be seen for spherical data in the next chapter, and reference to a more controversial case will be made shortly in discussion of

multidimensional scaling. Some investigation of artefactual structure was undertaken by Dzwinel [1994]. One particular illustration was given for data generated uniformly at random from within a 100-dimensional hypercube, which resulted in a circular configuration when mapped down to two dimensions. The cause of this particular configuration was actually explained by the author, with reference to the "curse-of-dimensionality", but this and other such projections may often appear inconsistent to the human observer because "our intuitive notions of low dimensions don't carry over well to high dimensions" [Friedman 1995].

Despite the simple, intuitive appeal of the Sammon Mapping, there are, however, some significant disadvantages and limitations to its application.

① The mapping is generated iteratively and has been observed to be particularly prone to sub-optimal local minima.

② The computational requirements scale with the square of the number of data points, making its application intractable for large data sets.

③ There is no method to determine the dimensionality of the feature space *a priori*.

④ The map is generated as a 'look-up table' — that is, there is no way to project new data without re-generating the entire map with the new data points included.

Sammon himself appreciated the restriction posed by item ② above, conceding that with the computing facilities available at that time, a practical upper limit of 200 data points was imposed. To partially overcome this, he proposed applying some *a priori* clustering process to extract prototypes, and then mapping these with the algorithm. This, and other approaches to the computational problem, will be considered in Chapter 7, with an investigation of problem ①, local minima, in Chapter 6. A considerable part of this thesis will be concerned with approaches to the problem posed by item ④, and this will be considered in more detail in the following chapters.

2.4 Comparison of the Kohonen SOFM and the Sammon Mapping

It has already been stated that, unlike the Sammon mapping, the generation of a Kohonen SOFM cannot strictly be interpreted as the minimisation of a single energy or cost function [Erwin, Obermayer, and Schulten 1992]. Aside from this, there is a more fundamental underlying difference between the two methods. It is the mechanism of the local neighbourhood function in the Kohonen map that affords the topographic nature of the scheme. However, there is no explicit retention of global structure, and indeed, the emphasis of the algorithm is to model the *density* of the underlying input data distribution. This contrast between the two techniques may be illustrated by the following simplistic example.

Both the SOFM and Sammon's algorithm are applied to the mapping of a synthetic dataset comprising three clusters in three dimensions. The clusters, C_1, C_2 and C_3, are centred at $(0,0,0)$, $(1,-1,0)$ and $(4,5,0)$, and contain 50, 100 and 50 points respectively. Each cluster is dispersed uniformly at random inside a cube centred at each point, with the size of each edge of the cube for C_3 being double that of C_1 and C_2. This distribution of data is illustrated via two orthogonal projections in figure 2.3. A (12×10) Kohonen Map of this data and the corresponding Sammon mapping are shown in figure 2.4. For comparison, the first two principal components of the data are plotted in figure 2.5.

This particular distribution of data was chosen deliberately to emphasise the differences in the methods. As illustrated in figure 2.4, the Sammon mapping offers a good representation of the original

topography. The variation of inter-cluster separation is still clear, and the increased dispersion of C_3 is also evident. The Kohonen SOFM has retained the local topology, but because it is a density-driven approach, fails to capture both the global relationships between the clusters and the local dispersion of C_3. Underlining this behaviour, the concentration of neurons in the region of class C_2, which contained twice the point density of the other classes, is also evident. The number of nodes activated for each of the three classes is 23, 40 and 27 respectively, which indicates that the map has over-represented the lower density clusters. That the cluster C_3 is significantly larger than C_1 and C_2 is not evident, and neither is its greater distance from those clusters.

In this simple case, a principal component projection is apparently adequate for retaining the topography (although close inspection will reveal better dispersion within the three clusters in the case of the Sammon Mapping). For real, higher-dimensional, datasets, this linear technique is generally limited in its application, as will be illustrated in Chapter 4.

An additional phenomenon inherent in the SOFM is the introduction of some topographic distortion due to the fixed topology of the lattice of output neurons. As asserted by Li, Gasteiger, and Zupan [1993], "global topology distortions are ... inevitable" in all but the most trivial situations. This effect is a result of mismatch between the topology of the lattice and that of the input data. This conclusion is also confirmed by Bezdek and Pal [1995] who claim that "the Sammon method preserves metric relationships much better than [the SOFM]." This assertion is a result of assessing the alternative mappings according to a measure of *metric topology preservation*, derived from Spearman's rank coefficient. With respect to this criterion, the Sammon mapping scored higher for all datasets tested.

The distortive aspect may be demonstrated by the example in figures 2.6 and 2.7. This illustrates a (12×12) 2D-sheet mapping of data points lying on three concentric 3-dimensional spheres, with radii 0,1 and 2 units respectively. Fifty points were distributed at random over each of the spheres and a small amount of Gaussian random noise was added, making the centre sphere effectively a cluster. The diagram in figure 2.6 shows the map, with its inevitable discontinuities, and below, in figure 2.7, is an illustration of the form of the sheet embedded in the input space — the 'frustration' in the lattice is clearly visible in this latter diagram. When such mismatch occurs, it may also induce poor performance from a clustering point of view. Such degradation, in comparison with the standard 'k-means' procedure, has been observed by Balakrishnan, Cooper, Jacob, and Lewis [1994].

Regarding these criticisms it should be noted that Kohonen's SOFM was developed as an analogue of observed neuro-biological behaviour, rather than being explicitly motivated by the criterion of faithful preservation of universal topography. In addition, in stark contrast to Sammon's technique, it has the attractive feature of good computational behaviour. It is this tractability for sizable datasets which makes the Kohonen SOFM a popular topographic mapping tool. However, on the basis of the discussion in this section, for applications in data analysis and visualisation, the Sammon mapping should be preferred for smaller datasets.

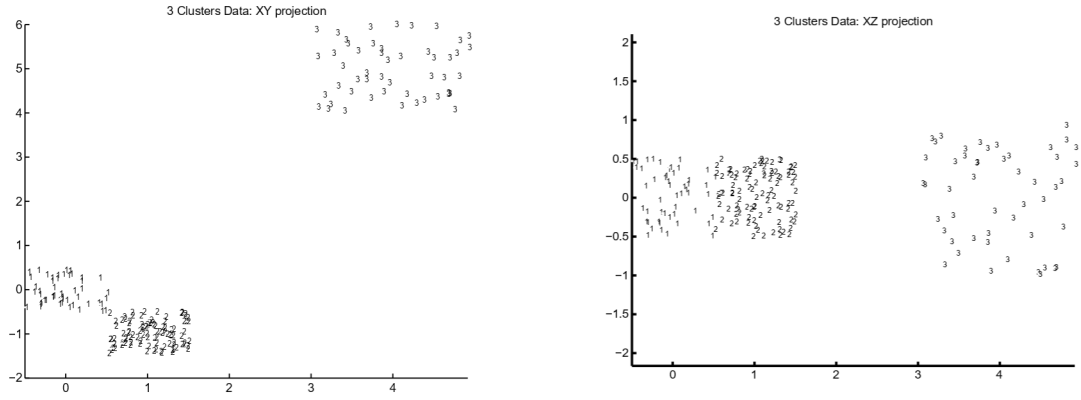

Figure 2.3: Synthetic data distributed in 3 clusters in 3-dimensional space.

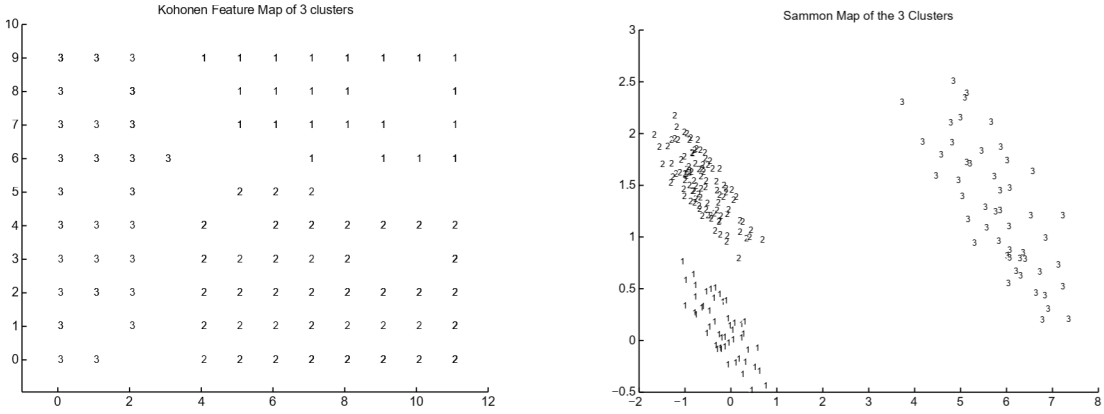

Figure 2.4: Kohonen and Sammon Mappings of the 3 clusters.

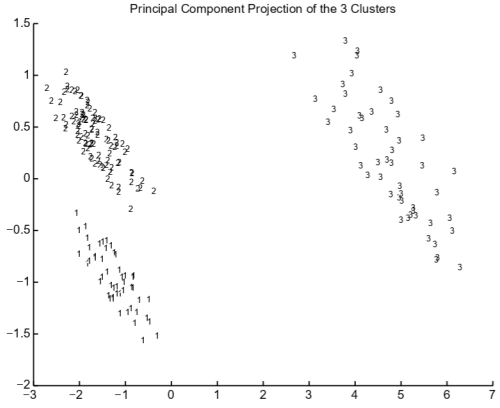

Figure 2.5: Projection onto the first two principal axes of the 3 clusters.

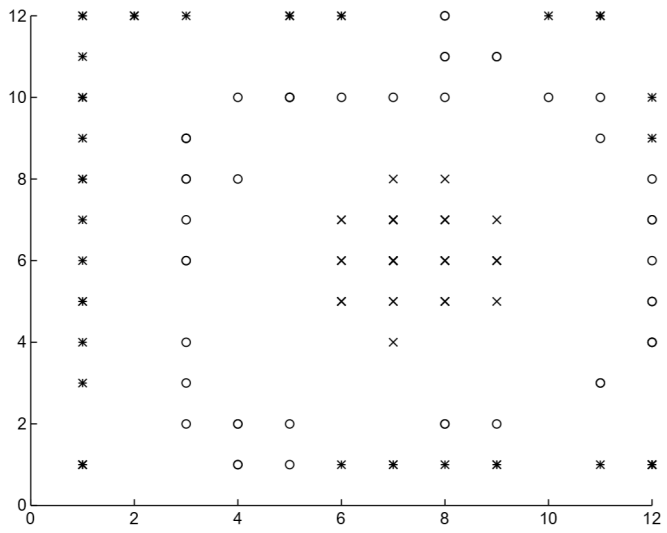

Figure 2.6: A Kohonen Mapping of data on 3 concentric spheres.

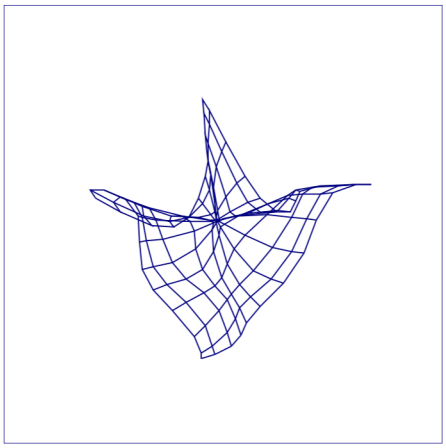

Figure 2.7: The Kohonen lattice embedded in the original space.

2.5 Multidimensional Scaling

2.5.1 The Underlying Principle

Multidimensional Scaling (MDS) is described by Davison [1983] as

> ...a set of multivariate statistical methods for estimating the parameters in, and assessing the fit of, various spatial distance models for proximity data.

This definition is a relatively narrow one, and some authors (e.g. Carroll and Arabie [1980]) accept a broader view and include other methods for modelling multivariate proximity data (such as factor analysis or cluster analysis) within the scope of MDS. However, it is the topographic properties of MDS that are relevant in this context, so the spatial distance definition is the most appropriate here.

The raw data to which MDS is applied is *proximity data.* This is generally in the form of a square symmetric ($N \times N$) matrix, where each row and column enumerates a set of objects and the elements of the matrix are measures of the relative proximity of those respective objects. In this context, proximity may refer to either *similarity* or *dissimilarity* of objects. It is then the purpose of MDS techniques to represent the structure of the proximity matrix in a more simple and perspicuous geometrical model. The classic example is that, given a matrix of road-distances (which can be considered analogous to dissimilarities) between cities, the data can be modelled by a two-dimensional map (e.g. see Krzanowski and Marriott [1994], pp113). In this instance, the scaling procedure greatly facilitates visualisation of the data and eliminates the redundancy in the description.

For the case of the road-distances, the geometric interpretation is intuitive and clearly valid. Typically, however, the proximity data processed by MDS models will have been gathered in a more subjective manner, often by means of psychological experiment where human subjects are asked to assess the likeness, or *similarity*, of each pair of objects, or *stimuli.* The fundamental assumption underlying the application of MDS in these contexts is that these empirical observations can be meaningfully fitted to a set of points in some metric space, where the distance between the points representing each pair of stimuli corresponds to their perceived *dissimilarity*. (The measure of 'dissimilarity' may be simply derived from that of 'similarity', for example, by subtracting from a constant.) This basic principle was originally proposed by Richardson [1938]. Given this assumption, it is then hoped that such a fitted configuration will aid visualisation of the data and also provide insight to the processes that generated it. These techniques have been successfully applied in a variety of fields — the behavioural and social sciences, psychology, acoustics, olfactory analysis, education and industrial relations are examples. A comprehensive list of many such applications is given by Davison [1983]. MDS remains a very popular tool, with a search of citation indices revealing relevant annual publications in the hundreds. A prominent, recent, and controversial example of the application of MDS techniques is in the study of connectivity of regions in the visual cortex of the macaque monkey [Young 1992]. This has provoked some significant debate over the validity of the structure inferred from such a model [Goodhill, Simmen, and Willshaw 1995], as to whether it is artefactual or truly representative of the underlying relationships in the data.

One particularly good illustration of MDS applied to psychological data concerns a study of colour vision. In this experiment, performed by Ekman [1954], participants estimated the similarity of all combinations of pairs of 14 different sample colours presented to them. An MDS technique was used to convert these similarity measurements into a configuration of points in two dimensions, where they were found to lie in a spectrally ordered manner on an annular, horseshoe, structure. This 'bending' of the colour line is a result of the phenomenon that many subjects (reasonably) perceive similarity between the two extreme ends of the spectrum — red and violet. The similarity data and the resulting mapping, which is clearly informative in this case, are given in figures 2.8 and 2.9 respectively.

$$\begin{bmatrix}
1 & .86 & .42 & .42 & .18 & .06 & .07 & .04 & .02 & .07 & .09 & .12 & .13 & .16 \\
.86 & 1 & .50 & .44 & .22 & .09 & .07 & .07 & .02 & .04 & .07 & .11 & .13 & .14 \\
.42 & .50 & 1 & .81 & .47 & .17 & .10 & .08 & .02 & .01 & .02 & .01 & .05 & .03 \\
.42 & .44 & .81 & 1 & .54 & .25 & .10 & .09 & .02 & .01 & .00 & .01 & .02 & .04 \\
.18 & .22 & .47 & .54 & 1 & .61 & .31 & .26 & .07 & .02 & .02 & .01 & .02 & .00 \\
.06 & .09 & .17 & .25 & .61 & 1 & .62 & .45 & .14 & .08 & .02 & .02 & .02 & .01 \\
.07 & .07 & .10 & .10 & .31 & .62 & 1 & .73 & .22 & .14 & .05 & .02 & .02 & .00 \\
.04 & .07 & .08 & .09 & .26 & .45 & .73 & 1 & .33 & .19 & .04 & .03 & .02 & .02 \\
.02 & .02 & .02 & .02 & .07 & .14 & .22 & .33 & 1 & .58 & .37 & .27 & .20 & .23 \\
.07 & .04 & .01 & .01 & .02 & .08 & .14 & .19 & .58 & 1 & .74 & .50 & .41 & .28 \\
.09 & .07 & .02 & .00 & .02 & .02 & .05 & .04 & .37 & .74 & 1 & .76 & .62 & .55 \\
.12 & .11 & .01 & .01 & .01 & .02 & .02 & .03 & .27 & .50 & .76 & 1 & .85 & .68 \\
.13 & .13 & .05 & .02 & .02 & .02 & .02 & .02 & .20 & .41 & .62 & .85 & 1 & .76 \\
.16 & .14 & .03 & .04 & .00 & .01 & .00 & .02 & .23 & .28 & .55 & .68 & .76 & 1
\end{bmatrix}$$

Figure 2.8: The proximity matrix for Ekman's colour data. Each value is a normalised, averaged, measure of observed similarity between 14 distinct sample colours.

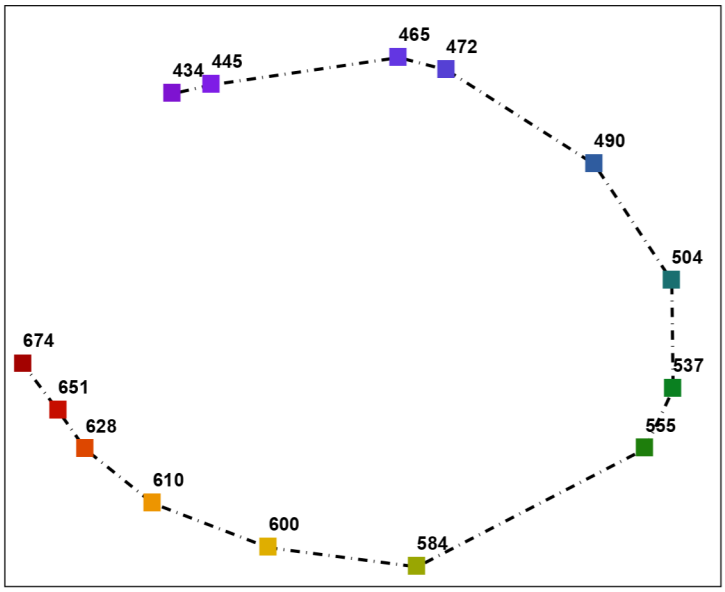

Figure 2.9: The resultant map, with wavelength shown for each sample, for Ekman's colour data.

2.5.2 Scaling Algorithms

The measured dissimilarity between a pair of objects (i,j), known as a *stimulus pair*, can be formalised as the variable δ_{ij}, which is an element of the $(N \times N)$ dissimilarity matrix $\mathbf{\Delta}$. It is the purpose of MDS to turn this data into a $(N \times q)$ configuration matrix $\mathbf{Y}$. In general, and in common with the Sammon Mapping, the dimension of the feature space q is unknown *a priori*.

The configuration of points $\mathbf{y}_i : i \in \{1 \ldots N\}$ must be determined such that the values δ_{ij} match some *distance function*, $d(i,j)$, defined over all possible pairs of points $(\mathbf{y}_i, \mathbf{y}_j)$. For $d(i,j)$ to be a distance function, the following four axioms must hold:

$$d(a,b) \geq 0, \tag{2.5}$$

$$d(a,a) = 0, \tag{2.6}$$

$$d(a,b) = d(b,a), \tag{2.7}$$

$$d(a,b) + d(b,c) \geq d(a,c). \tag{2.8}$$

In a psychological context (e.g. consider the colour data), these first three axioms, (2.5)-(2.7) appear intuitively reasonable, although there is no apparent support or contradiction for (2.8), known as the triangular inequality axiom. Whilst much experimental work corroborates the distance model for psychological data, the results of some tests appear to violate some of the axioms (e.g. Rothkopf 1957). However, this is just one aspect of the application of MDS — in other contexts these contradictions are not manifest.

Usually, the metric employed in the configuration space is the standard Euclidean, although general Minkowski distances have been used in particular applications.

There are two main branches of MDS models — the *metric* and the *nonmetric* methods. In the former, the dissimilarities should correspond as closely as possible to the inter-point distances in the generated configuration. In the latter scheme this constraint is relaxed, with psychological justification, such that the *ordering* of the dissimilarities should correspond to the *ordering* of the distances. The metric techniques, originally developed by Torgerson [1952] as *classical* MDS, have been superseded by the more flexible and effective nonmetric models. The following two subsections cover these methods in more detail.

2.5.3 Classical Multidimensional Scaling (CMDS)

One of the first MDS algorithms was proposed by Torgerson [1952, 1958]. By definition as a metric method, it assumes the identity relationship between distance in the feature space and corresponding object dissimilarity:

$$\delta_{ij} = d(i,j). \tag{2.9}$$

As such, it requires somewhat restrictive assumptions, and is seldom used in its original form, although many more developed algorithms build on it. It does, however, have the advantage of an analytical derivation.

The CMDS procedure is as follows:

❶ From the dissimilarity matrix $\mathbf{\Delta}$, generate the *double-centred inner product matrix* $\mathbf{B}^*$, given by:

$$\mathbf{B}^* = -\frac{1}{2}\mathbf{H}\mathbf{\Delta}_2\mathbf{H}, \tag{2.10}$$

with $\mathbf{\Delta}_2$ the matrix whose elements are the square of those of $\mathbf{\Delta}$. That is, $\mathbf{\Delta}_2 = \{\delta_{ij}^2\}$. The matrix $\mathbf{H}$ is the *centring* matrix, given by $\mathbf{I} - \mathbf{1}/N$, where $\mathbf{1}$ is the square matrix whose elements are all 1.

❷ Factorise $\mathbf{B}^*$ into:

$$\mathbf{B}^* = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\mathbf{T}, \tag{2.11}$$

$$= \mathbf{Y}\mathbf{Y}^\mathbf{T}. \tag{2.12}$$

The matrix $\mathbf{\Lambda}$ is the diagonal matrix of eigenvalues of $\mathbf{B}^*$, with $\mathbf{U}$ the corresponding matrix of eigenvectors.

❸ The matrix $\mathbf{Y} = \mathbf{U}\mathbf{\Lambda}^{1/2}$ is the configuration of points in $p = N$ dimensions that satisfies exactly the dissimilarity measures specified in $\mathbf{\Delta}$. To reduce to q dimensions, select the q columns of $\mathbf{U}$ corresponding to the q largest eigenvalues, giving an $(N \times q)$ data matrix $\mathbf{Y}_q$.

There are several points to be noted about the CMDS procedure:

- The points $\mathbf{Y}$ are centred at the origin, so $\sum_i^N \mathbf{y}_i = \mathbf{0}$.
- Calculation of $\mathbf{\Lambda}^{1/2}$ requires that $\mathbf{B}^*$ is positive semi-definite. Techniques adopted for dealing with the problem of negative eigenvalues are typically heuristic. One is to ignore the small, negative eigenvalues. Another is the *trace criterion*, where the sum of the discarded negative eigenvalues should equal the sum of the positive discards, so the sum of the remaining eigenvalues is still equal to the trace of the matrix. A large negative eigenvalue is nevertheless a major problem. Mardia [1978] proposed "goodness of fit" measures for such non-Euclidean data.
- If $\mathbf{B}^*$ is positive semi-definite, then $\mathbf{\Delta}$ is a Euclidean distance matrix. That is, the dissimilarities δ_{ij} correspond exactly to the Euclidean distances between a set of points embedded in at most $(N-1)$ dimensions.
- If $\mathbf{B}^*$ is positive semi-definite, then the points $\mathbf{y}_i$ are referenced to their principal axes. (That is, $\mathbf{Y}^\mathbf{T}\mathbf{Y}$ is diagonal). Furthermore, if the elements of $\mathbf{\Delta}$ are the inter-point Euclidean distances of a given set of data points, then the CMDS solution in q dimensions is identical to a projection onto the first q principal axes of the data. Indeed, CMDS is sometimes known, after Gower [1966], as *principal co-ordinates analysis*.
- For a *similarity* matrix $\mathbf{S}$, where $0 \leq s_{ij} \leq 1$ and $s_{ii} = 1$ (such as that given for the colour data in figure 2.8), then a corresponding dissimilarity matrix can be formed by $\delta_{ij} = \sqrt{(1 - s_{ij})}$. In that case, $\mathbf{\Delta}$ is a Euclidean distance matrix [Gower and Legendre 1986].

2.5.4 Nonmetric Multidimensional Scaling (NMDS)

In Nonmetric, or *ordinal*, Multidimensional Scaling (NMDS) the requirement that distances in the projected space optimally fit the dissimilarities is relaxed so that only the *ordering* of distances is retained. That is, the two most dissimilar stimuli should also be the two most distant points in the configuration and the second most dissimilar pair of stimuli be the second most distant pair of points etc. It is therefore not necessary for all corresponding pairs of distances and dissimilarities to be identical. In fact, the ordinal constraint implies that it is only necessary that the dissimilarities be some arbitrary monotonically increasing function of the distances.

Thus in contrast to equation (2.9), for nonmetric models the relationship between dissimilarity and spatial distance becomes

$$\delta_{ij} = f(d_{ij}) = f\left[\sum_k (x_{ik} - x_{jk})^2\right]^{1/2}, \tag{2.13}$$

where f is a monotone function such that

$$\forall i,j,i',j' : d_{ij} < d_{i'j'} \Rightarrow f(d_{ij}) < f(d_{i'j'}). \tag{2.14}$$

Note that the function f need never be known explicitly, although its form may be recovered after the scaling procedure.

As well as adhering to the first three Euclidean distance axioms, this concept preserves many intuitive psychological properties and in many cases permits the generation of more useful, lower-dimension, lower-STRESS mappings. Furthermore, in some experiments, only ordinal data is available, notably where human subjects are required to rank various stimuli in order of merit or preference.

In contrast to the classical method, these ordinal configurations are generated by minimisation of a particular cost function, or STRESS measure, and must be generated iteratively via some nonlinear optimisation procedure. Because of the ordinal constraint, generating a configuration is particularly computationally expensive due to the additional requirement of a *monotonic regression* step.

The first nonmetric scheme was proposed by Shepard [1962a, 1962b], in response to experimental evidence that in certain applications, observed dissimilarities were related to some nonlinear function of the spatial distances in a putative model (e.g., Shepard 1958). These methods were further and more formally developed by Kruskal and that work remains the basis of modern implementations. Kruskal [1964a] formalised the method by defining a measure of *goodness-of-fit*. The proposed MDS technique is thus to determine a point configuration $\mathbf{Y}$ that optimises this. A practical computer implementation of the algorithm is described in a companion paper [Kruskal 1964b]. To clarify the NMDS technique, consider the following description of Kruskal's procedure.

Given a set of experimentally obtained dissimilarity data δ_{ij} and a configuration of N points in q dimensions, the dissimilarities can be ranked according to their magnitude

$$\delta_{i_1 j_1} < \delta_{i_2 j_2} < \ldots < \delta_{i_M j_M} \tag{2.15}$$

where $M = N(N-1)/2$. It is then possible to determine a set of *disparities* $\hat{d}_{ij}$ that are "nearly equal" to d_{ij}, whilst still retaining monotonicity with respect to the corresponding δ_{ij}. That is:

$$\hat{d}_{i_1 j_1} \leq \hat{d}_{i_2 j_2} \leq \ldots \leq \hat{d}_{i_M j_M} \tag{2.16}$$

The $\hat{d}_{ij}$ are said to be *monotonically related* to the d_{ij}, and the fitting of those values is a monotonic regression of distance upon dissimilarity. The multi-pass procedure for the determining the disparities is as follows.

At each monotonic regression phase, the disparities $\hat{d}_{ij}$ are initialised as the distances d_{ij}, and are listed such that their corresponding dissimiliarities δ_{ij} are in ascending order. In the first pass, each pair of adjacent (list-wise) disparity values is compared and if the two variables are not correctly ordered, they are combined into a 'group' with a common disparity value equal to the arithmetic mean of the combined values. Subsequent passes are then similar, except previously combined groups may be compared together as well as with other adjacent groups and / or single disparity values. The procedure terminates when no further groupings are required and the listed disparities are either equal (within groups) or in the requisite ascending order.

Kruskal then defined an objective measure based on these disparities:

$$\text{STRESS} = \sqrt{\frac{\sum_{i<j}(d_{ij} - \hat{d}_{ij})^2}{\sum_{i<j} d_{ij}^2}}, \tag{2.17}$$

where the denominator again normalises for the number and scaling of the dissimilarities.

The monotonic regression step occurs first, determining the $\hat{d}_{ij}$, following which those calculated values are used in a gradient-descent minimisation step of equation (2.17). These alternate steps are then repeated until a local minimum of STRESS is attained. Inevitably this procedure is computationally demanding and prone to finding sub-optimal local minima. Also, there is again no indication of choice

of dimensionality q, so it is usually necessary to generate several configurations in a number of dimensions for comparative purposes.

The modern approach to NMDS is the *Alternating Least Squares* procedure, ALSCAL, and is available in the popular SPSS software package [Young and Harris 1990]. Since their introduction, the nonmetric schemes have become the dominant scaling models, with the classical procedure now rarely used.

2.6 Comparison of MDS and the Sammon Mapping

In Sammon's original paper he briefly mentioned the connection to MDS methods, and this relationship was further clarified by Kruskal [1971].

Sammon's mapping is effectively a metric, but nonlinear, scaling method. As such, its exact analogue does not exist in the MDS domain. Whilst all these latter scaling techniques may be applied to dissimilarities generated directly from a set of points, doing so defeats the primary motivation behind their development which is to produce such a spatial configuration from non-spatial data. Nevertheless, comparison of equations (2.17) and (2.3) indicates that the operation of the Sammon mapping, ignoring normalisation terms, is identical to a nonmetric scaling procedure without the monotonic regression step.

Algorithmically, MDS and the Sammon Mapping are effectively identical; it is only the difference in the source of the input data that differentiates between the two schemes. Conceptually, this is illustrated in figure 2.10 below.

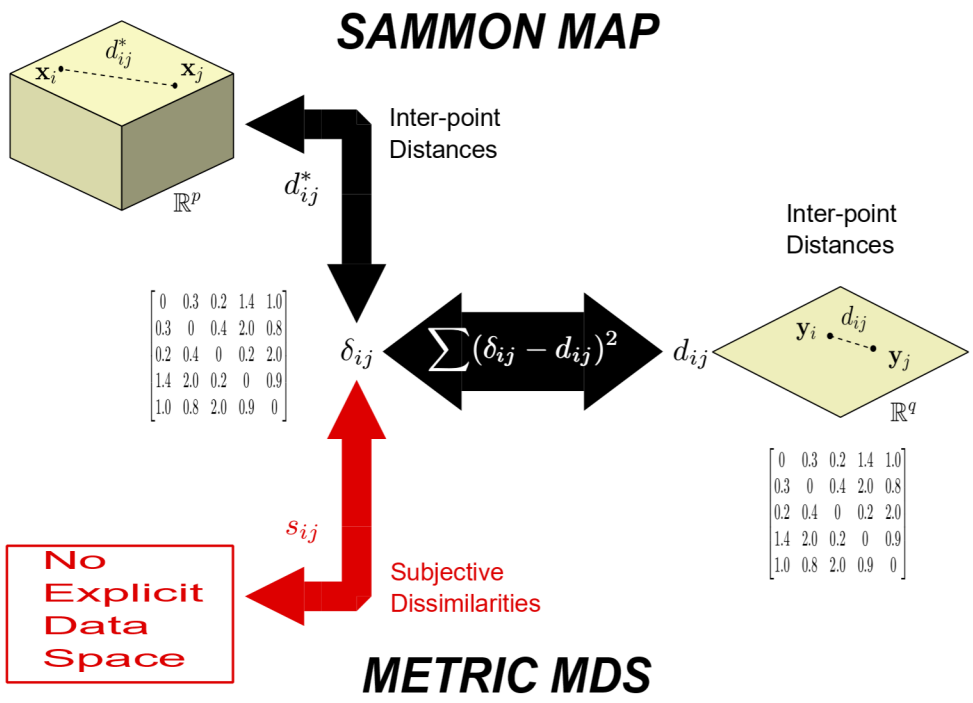

Figure 2.10: A schematic of the operation of the Sammon Mapping and MDS, emphasising the conceptual distinction between the two.

2.7 Conclusions

This chapter has described the standard methods for topographic feature extraction in both the neural network and statistical domains. The next chapter will introduce an alternative, *feed-forward* neural network approach to topographic mapping, which will be based on Sammon's projection. The emphasis of the particular method proposed is for the purposes of visualisation or exploratory data analysis, and this has motivated the choice of the latter technique as the basis for its design. As illustrated in Section 2.4, Sammon's approach to topographic mapping retains significantly more of the salient global data structure than the SOFM paradigm.

The key principle from MDS outlined in Section 2.5 — that informative configurations of points can be generated via topographic constraints from non-spatial data — will be incorporated into the method to enable the exploitation of additional subjective knowledge.

A generalisation of this neural network model to classical MDS in particular will be examined in Chapter 5, in the context of principal components analysis with neural networks.

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

$$d_{ij} = \| \mathbf{y}_i - \mathbf{y}_j \|, \tag{3.2}$$

$$= \left[(\mathbf{y}_i - \mathbf{y}_j)^{\mathrm{T}} (\mathbf{y}_i - \mathbf{y}_j) \right]^{1/2}, \tag{3.3}$$

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:

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

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.

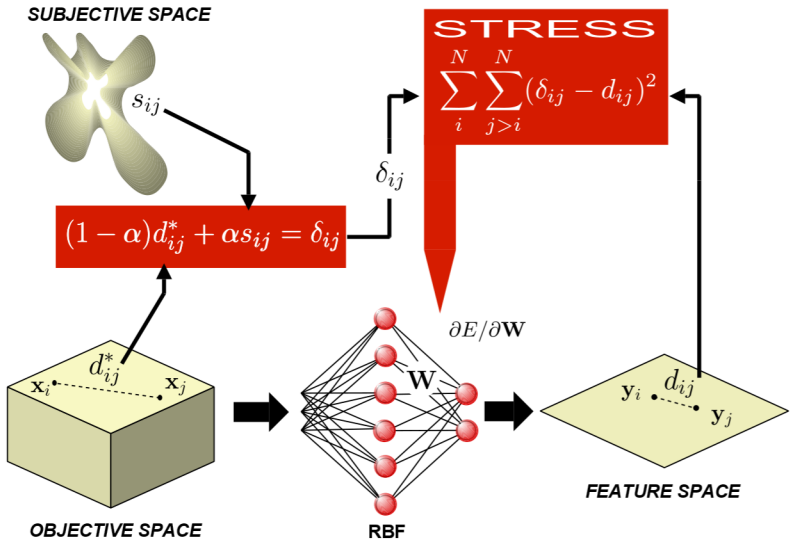

Figure 3.2: A schematic of the operation of NEUROSCALE.

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.

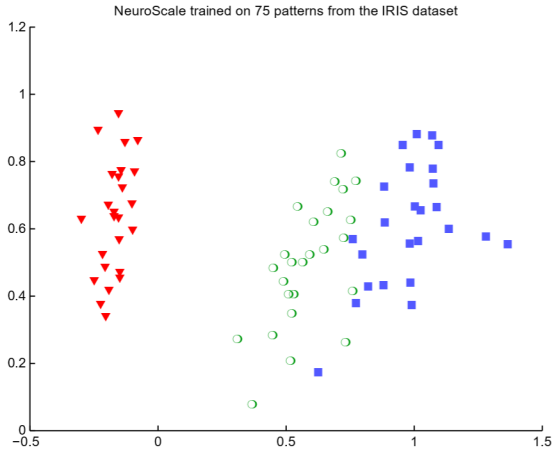

Figure 3.3: The resulting projection when NEUROSCALE is trained on 75 patterns of the Iris dataset. The STRESS for this configuration is 0.00275.

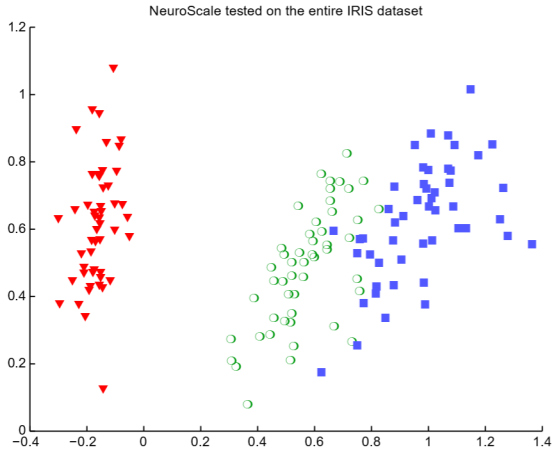

Figure 3.4: The projection when the trained NEUROSCALE network of the previous figure is tested on all 150 patterns of the Iris dataset. The STRESS for this configuration is 0.00325.

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.

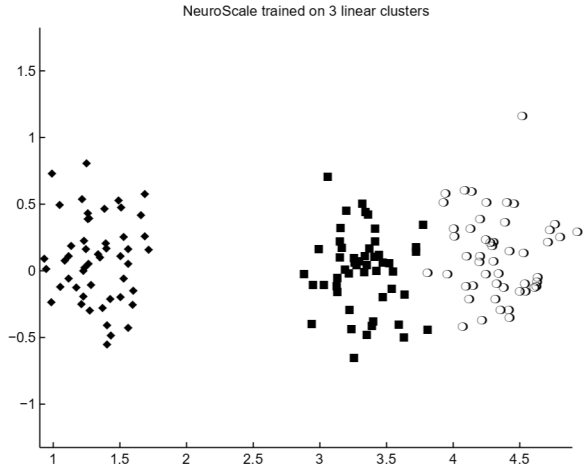

Figure 3.5: The resulting projection when NEUROSCALE is trained on 3 of 4 linear clusters. The STRESS for this configuration is 0.00515.

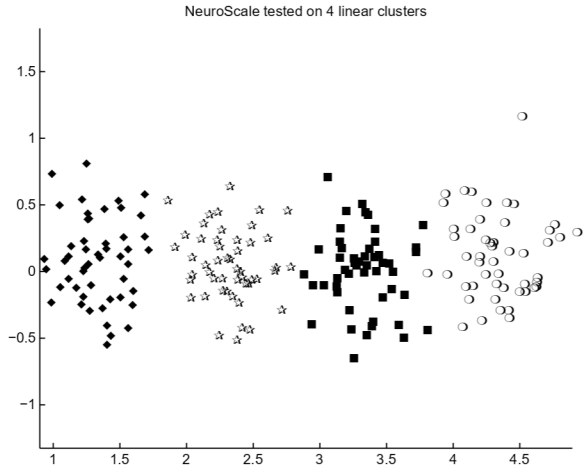

Figure 3.6: The projection when the trained NEUROSCALE network of the previous figure is tested on all 4 clusters. The STRESS for this configuration is 0.00532.

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.

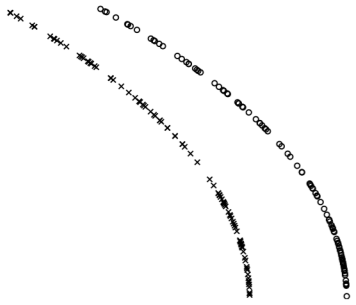

Figure 3.7: Cross section of the two adjacent surfaces.

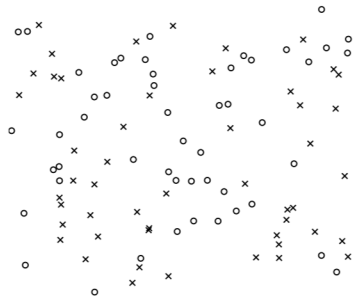

Figure 3.8: An RBF topographic projection of two adjacent surfaces with $\alpha = 0$.

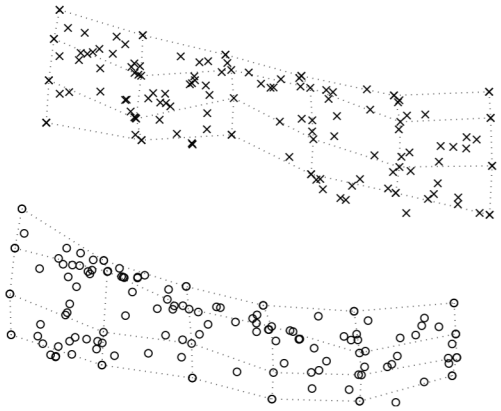

Figure 3.9: An RBF topographic projection of two adjacent surfaces with $\alpha = 0.5$. A grid indicating lines of constant 'height' and 'width' is superimposed.

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.

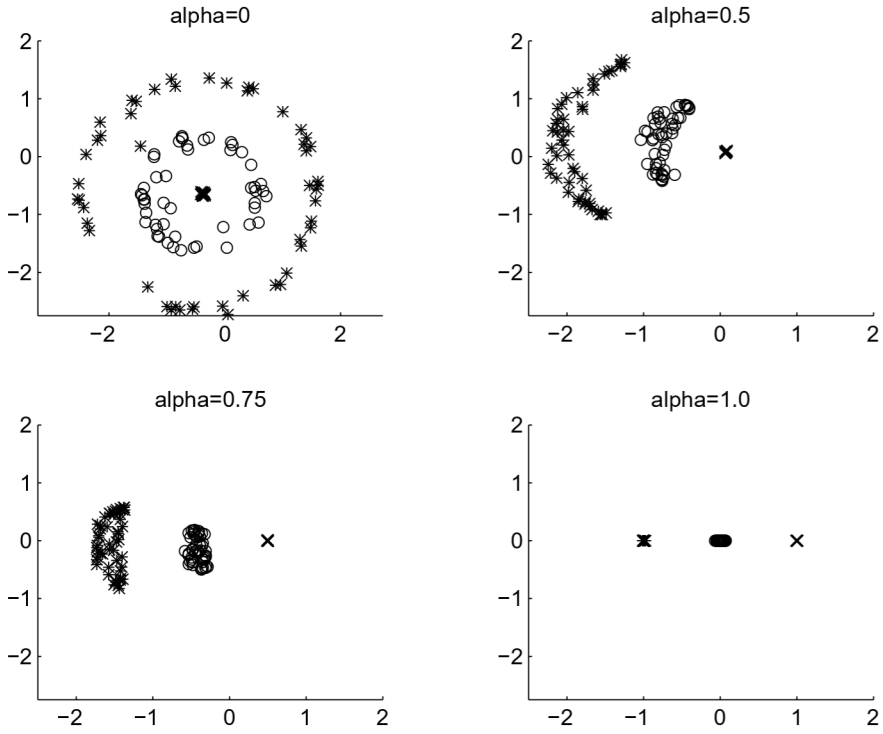

Figure 3.10: Projections of the 3-Spheres data for subjective matrix $\mathbf{C}_1$.

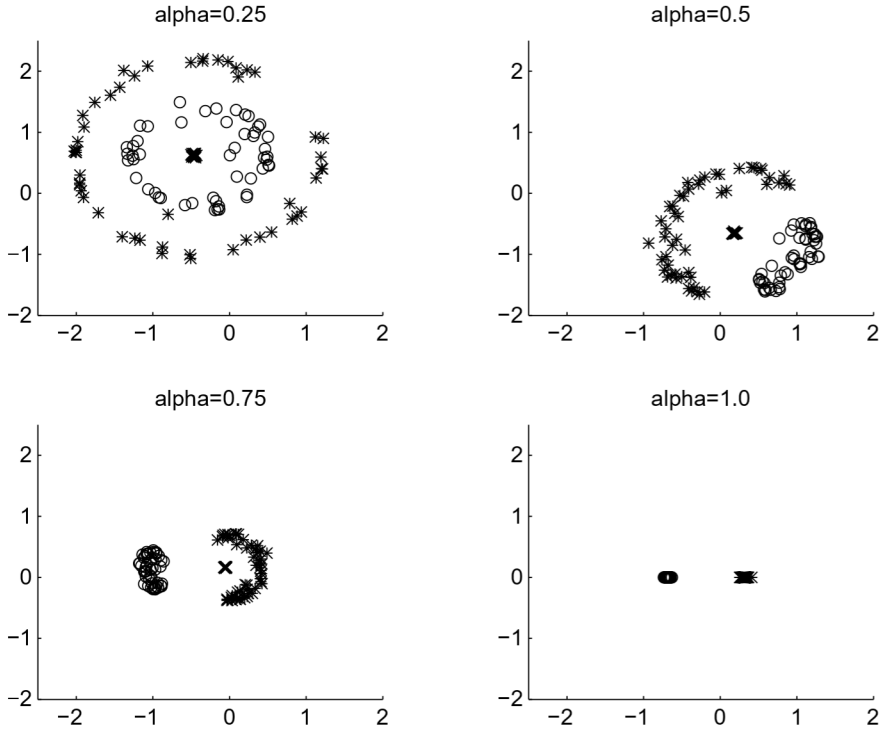

Figure 3.11: Projections of the 3-Spheres data for subjective matrix $\mathbf{C}_2$.

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 ben-

efits 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,

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

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