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.

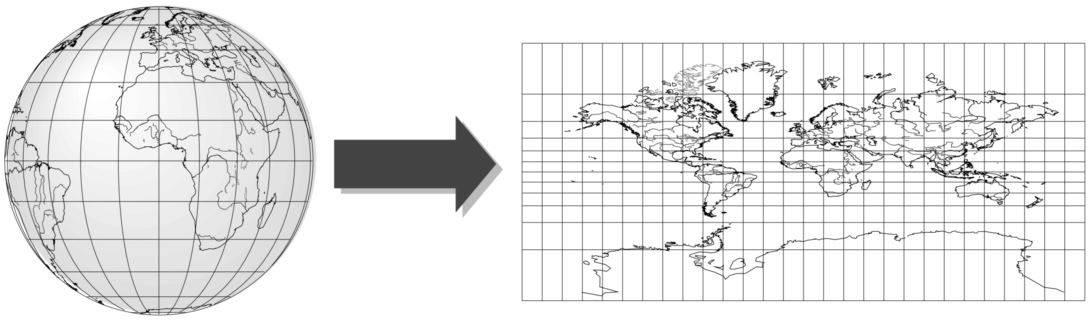


Figure 1.1: A Mercator's projection of the spherical Earth down to a two-dimensional map.

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.

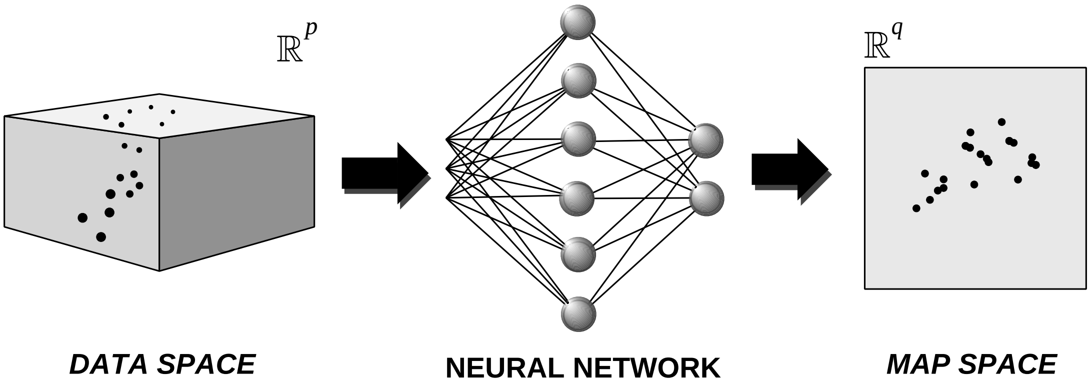


Figure 1.2: A neural network effecting a topographic transformation.

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, $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)^{\mathrm{T}}$
$\mathbf{y}_i$	A point vector in the feature space
$\mathbf{Y}$	The matrix of row-vector mapped points, $(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N)^{\mathrm{T}}$
$\mathbf{A}^{\mathrm{T}}$	The transpose of matrix (or vector) $\mathbf{A}$
$\mathrm{tr}\,[\mathbf{A}]$	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 **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

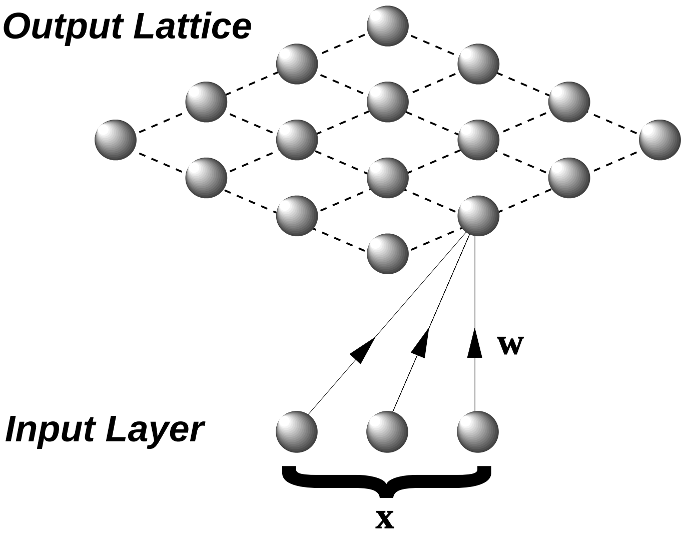


Figure 2.1: A schematic of the architecture of a two-dimensional output layer Kohonen network. For clarity, only the weight connections from the input to a single neuron are shown.

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

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

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.

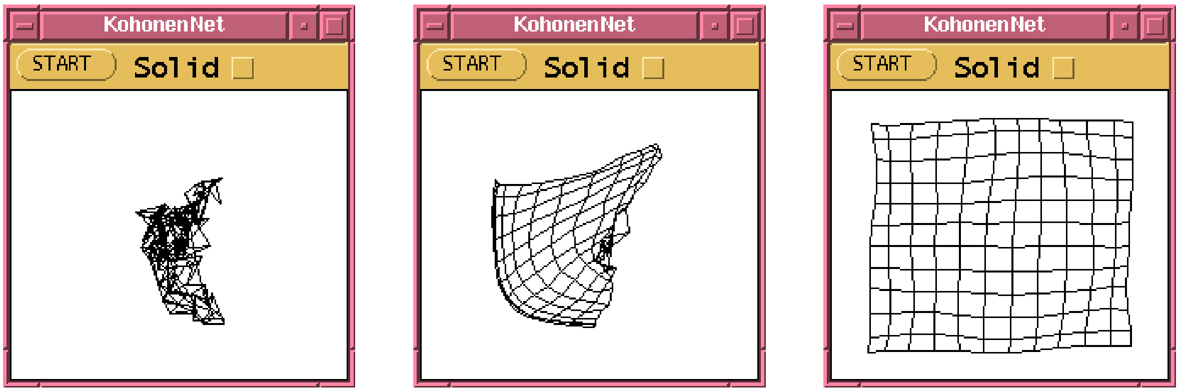


Figure 2.2: A 2-D Kohonen map of data sampled uniformly at random from the unit square. The final map is after 5,000 time steps.

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.

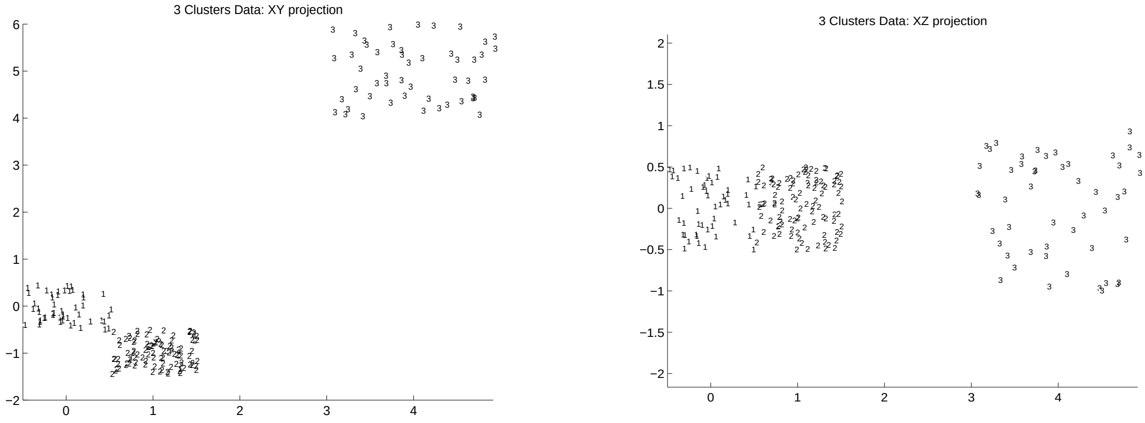


Figure 2.3: Synthetic data distributed in 3 clusters in 3-dimensional space.

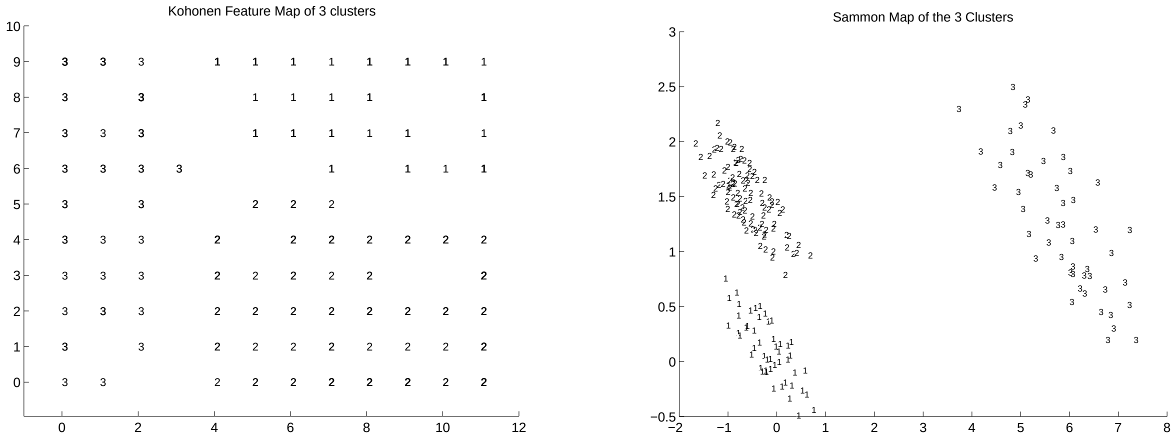


Figure 2.4: Kohonen and Sammon Mappings of the 3 clusters.

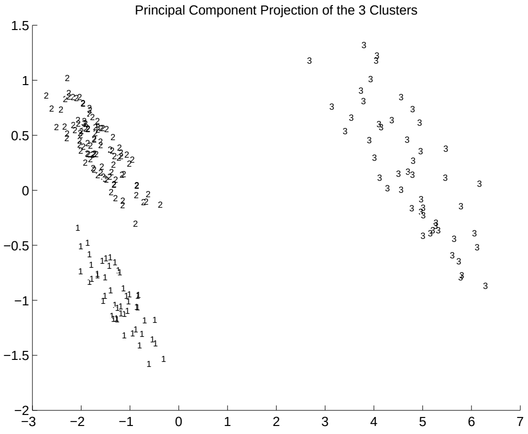


Figure 2.5: Projection onto the first two principal axes of the 3 clusters.

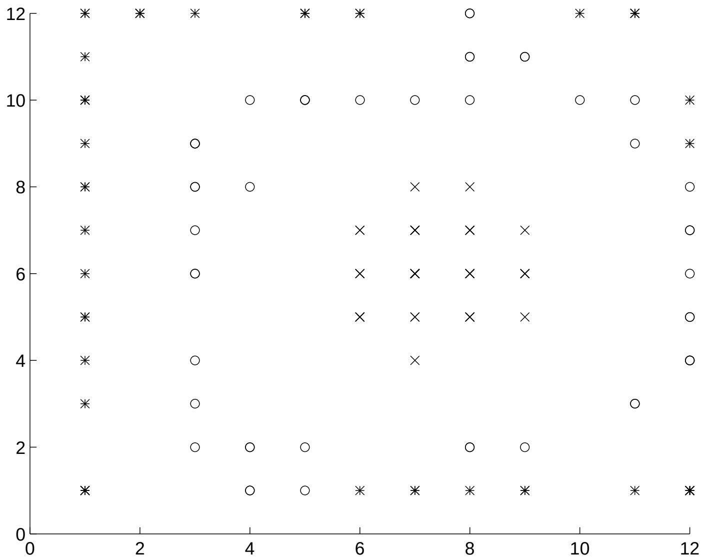


Figure 2.6: A Kohonen Mapping of data on 3 concentric spheres.

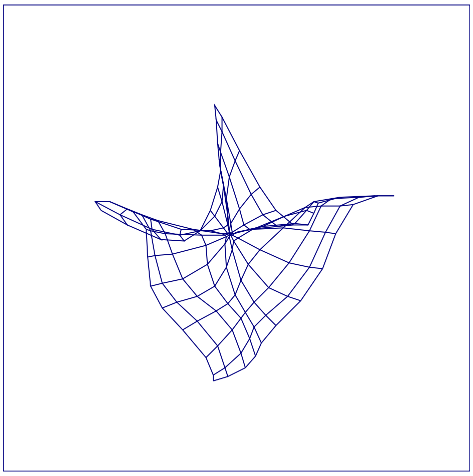


Figure 2.7: The Kohonen lattice embedded in the original space.

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.

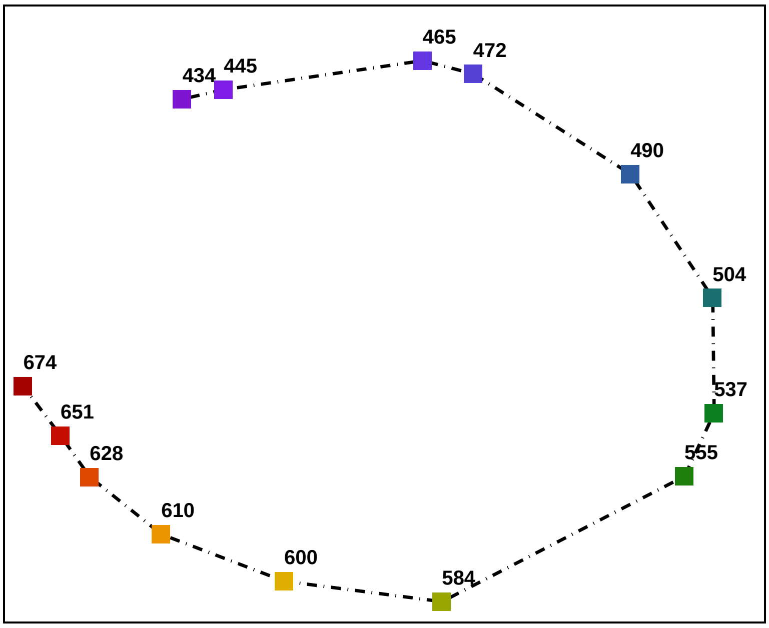


Figure 2.9: The resultant map, with wavelength shown for each sample, for Ekman's colour data.

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:

➊ 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:

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

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

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

❸ The matrix $\mathbf{Y} = \mathbf{U}\boldsymbol{\Lambda}^{1/2}$ is the configuration of points in $p = N$ dimensions that satisfies exactly the dissimilarity measures specified in $\boldsymbol{\Delta}$. 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 $\boldsymbol{\Lambda}^{1/2}$ 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 $\boldsymbol{\Delta}$ 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, $\mathbf{Y}^{\mathrm{T}}\mathbf{Y}$ is diagonal). Furthermore, if the elements of $\boldsymbol{\Delta}$ 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 $0 \le s_{ij} \le 1$ 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, $\boldsymbol{\Delta}$ 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

$$\delta_{i_1j_1} < \delta_{i_2j_2} < \ldots < \delta_{i_Mj_M} \tag{2.15}$$

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:

$$\hat{d}_{i_1j_1} \leq \hat{d}_{i_2j_2} \leq \ldots \leq \hat{d}_{i_Mj_M} \tag{2.16}$$

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.

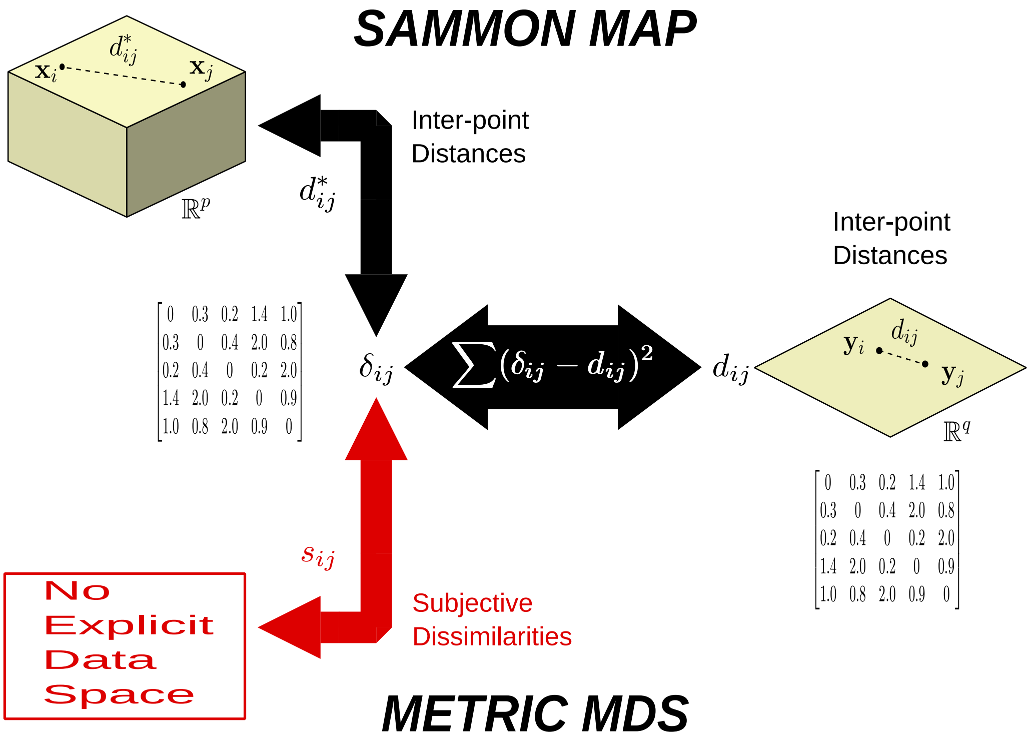


Figure 2.10: A schematic of the operation of the Sammon Mapping and MDS, emphasising the conceptual distinction between the two.

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

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

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

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

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

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

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

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

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

3.2.2 Calculating Weight Derivatives

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

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

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

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

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

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

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

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

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

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

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

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

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

3.3 Exploiting Additional Knowledge

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

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

3.3.1 Class Knowledge

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

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

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

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.

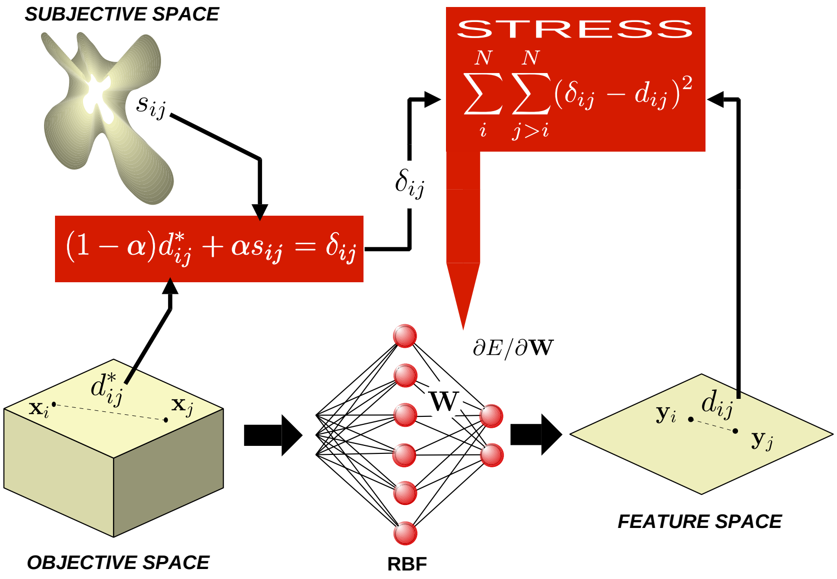


Figure 3.2: A schematic of the operation of NEUROSCALE.

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.

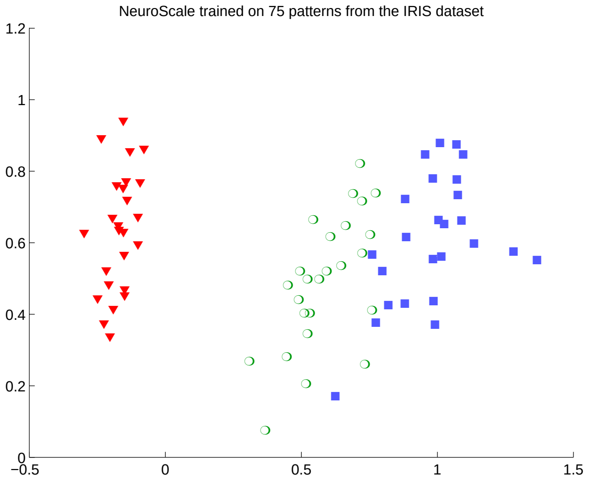


Figure 3.3: The resulting projection when NEUROSCALE is trained on 75 patterns of the Iris dataset. The STRESS for this configuration is 0.00275.

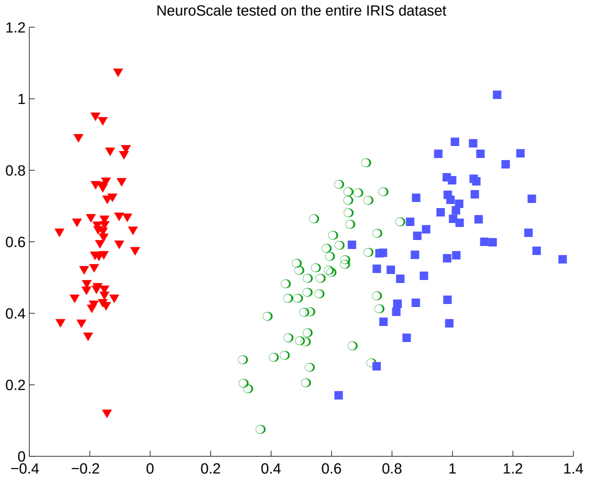


Figure 3.4: The projection when the trained NEUROSCALE network of the previous figure is tested on all 150 patterns of the Iris dataset. The STRESS for this configuration is 0.00325.

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.

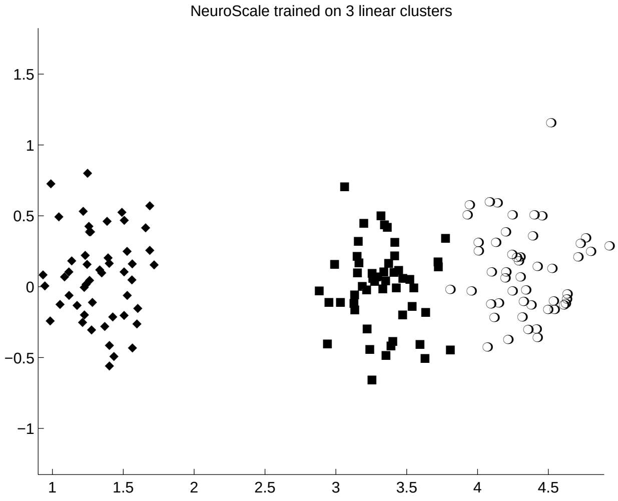


Figure 3.5: The resulting projection when NEUROSCALE is trained on 3 of 4 linear clusters. The STRESS for this configuration is 0.00515.

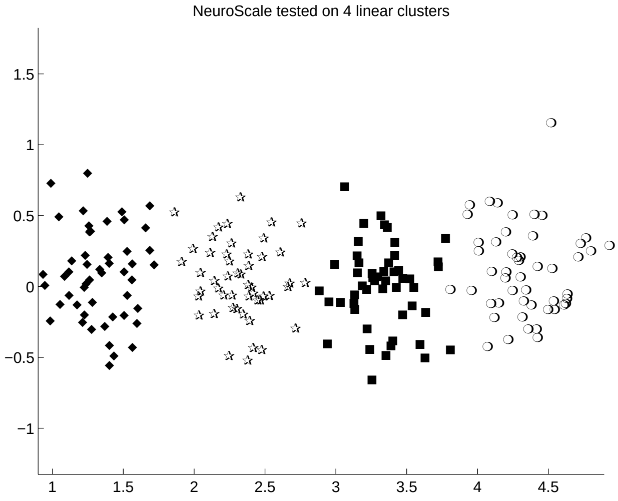


Figure 3.6: The projection when the trained NEUROSCALE network of the previous figure is tested on all 4 clusters. The STRESS for this configuration is 0.00532.

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.

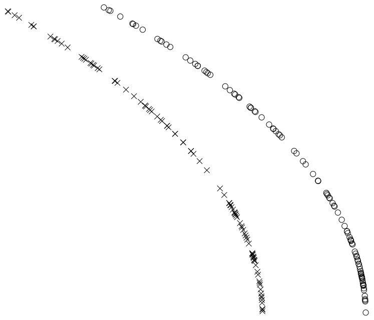


Figure 3.7: Cross section of the two adjacent surfaces.

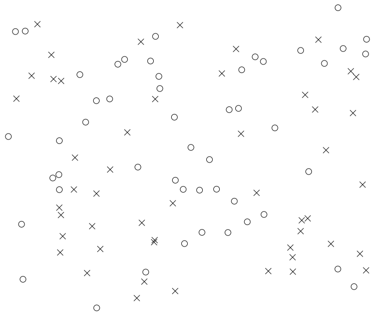


Figure 3.8: An RBF topographic projection of two adjacent surfaces with $\alpha = 0$.

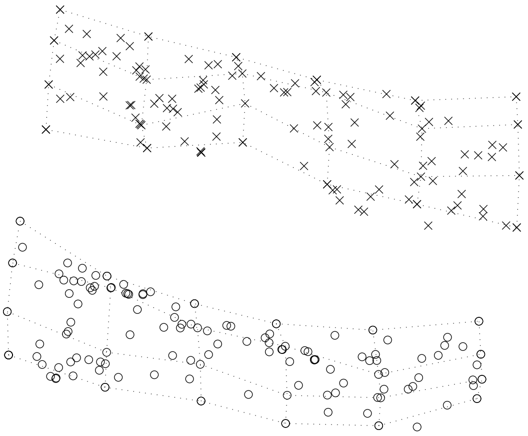


Figure 3.9: An RBF topographic projection of two adjacent surfaces with $\alpha = 0.5$. A grid indicating lines of constant 'height' and 'width' is superimposed.

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.

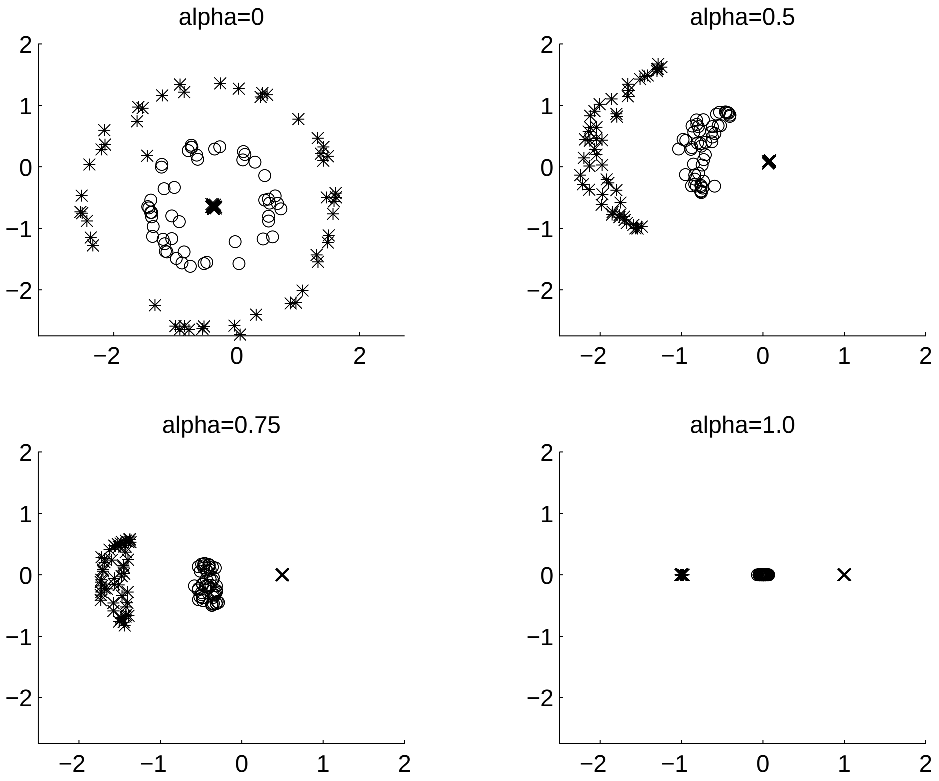


Figure 3.10: Projections of the 3-Spheres data for subjective matrix $\mathbf{C}_1$.

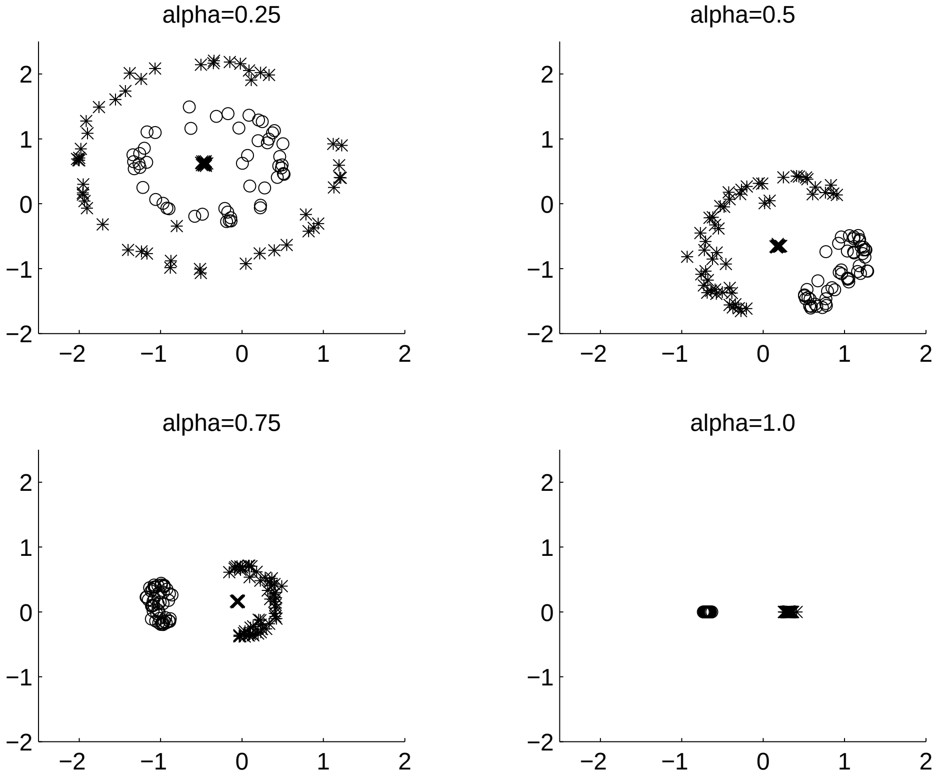


Figure 3.11: Projections of the 3-Spheres data for subjective matrix $\mathbf{C}_2$.

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 benefits of generalisation to new data were alluded to, no illustration of this capability was given.

Tattersall and Limb [1994]

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

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

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

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

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

3.5.2 Simple 'Binary' Mapping of Class-Labelled Data

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

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

Koontz and Fukunaga [1972]

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

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

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

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

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

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

The separability term is

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

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

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

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

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

Cox and Ferry [1993]

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

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

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

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

Webb [1995]

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

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

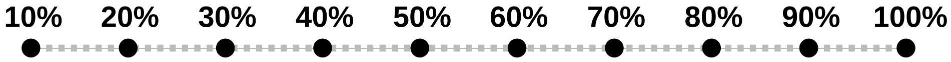
where the two criteria J_{SE} and J_{SP} were those as used by Koontz and Fukunaga. The parameter λ $(0 \leq \lambda \leq 1)$ allows a mixing of the two criteria. The hybrid STRESS measure, *J*, was then minimised via the *iterative majorisation* method, as employed in the popular ALSCAL procedure [Young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

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

3.5.3 General Mapping of Class-Labelled Data

Lowe [1993]

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



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

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

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

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

3.5.4 Relationships with NEUROSCALE

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

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

3.6 Conclusions

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

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

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

To summarise the advantageous features of NEUROSCALE:

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

There nevertheless remain some limitations:

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

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

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

Chapter 4

Feature Extraction and the 1992 Research Assessment Exercise

4.1 Introduction

This chapter is a study of the application of NeuroScale to a subset of data from the 1992 United Kingdom Research Assessment Exercise (Rae), which has recently been made publicly available [Higher Education Funding Council for England 1994]. This is a 'difficult', high-dimensional, real-world database and offers a good illustration of the potential for visualisation and exploratory analysis provided by the feed-forward neural network topographic approach introduced in the previous chapter.

In the next section some background and details concerning the Rae database are given, followed by a consideration of the NeuroScale technique from the perspective of *feature extraction*. This will then be the basis for a comparison of various projections of the dataset obtained both by NeuroScale, and by more established methods. To accompany this analysis, there is a description and discussion of classification experiments undertaken on the data presented in Sections 4.4.2 and 4.5.

4.2 The Rae Dataset

4.2.1 Background to the Research Assessment Exercise

One of the major factors determining centralised research funding for university departments[1], through the higher education funding councils in the United Kingdom, is the *research rating*, awarded to each such department as a result of a Research Assessment Exercise (Rae). Such an exercise has been held in 1989 and 1992, with the next one taking place in 1996. In the last exercise, in 1992, there were five integer research ratings, ranging from 1 (lowest quality) to 5 (highest quality). (For the 1996 exercise there will be an expansion to 7 ratings.) Funding varies across subject areas as, generally, the humanities receive less financial support than science or engineering, and is also in proportion to the

[1] The word 'department' is used here as a convenience. The actual individual entities that are assessed are termed 'Units of Assessment', and may, at some institutions, encompass more than one department or alternatively, a single department may have more than one relevant Unit of Assessment.

department size. These factors being equal, research funding is then allocated according to that department's research rating. A department with a research rating of '1' receives no funding at all, a rating of '5' is rewarded with four times the funding of a '2'. So, taking these influences into account, for a given subject area, approximately, $\textit{funding} \propto (\textit{rating} - 1) \times \textit{size}$.

For the purposes of the 1992 RAE, each higher educational institution supplied numerous quantitative measures of their research activity for 72 different subject areas, and each such department was required to supply the variables listed in table 4.1. The submitted data was then intended to be a comprehensive record of the research activity subsequent to the previous exercise in 1989.

```
Institution No.
Unit of assessment No./No.+letter
No. of selected staff
No. of staff not selected
No. of postdocs
No. of postgrads
No. of technicians
No. of scientific officers
No. of experimental officers
No. of other staff
No. of selected staff
FTE of selected staff
No. of staff funded from general income
No. of staff funded from other income
No. of staff in post throughout
No. of category A staff
No. of category B staff
No. of category C staff
No. of category D staff
No. of publications
No. of research assistants
No. of postgrad research students
No. of cited authored books
No. of cited edited books
No. of cited short works
No. of cited refereed conferences
No. of cited other conferences
No. of cited editorships
No. of cited articles for academic journals
No. of cited articles for professional journals
No. of cited articles for popular journals
No. of cited reviews of academic books
No. of cited other publications
No. of cited misc publications
No. of cited pubs classified as applied research
No. in FT research
No. in PT research
No. of Doctorates
No. of Masters
No. of FTE postgrad students
No. of student[ship]s from ABRC res. counc.s etc
No. of student[ship]s from UK based charities
No. of student[ship]s from UK central government
No. of student[ship]s from UK local authorities
No. of student[ship]s from UK public corporations
No. of student[ship]s from UK industry & commerce
No. of student[ship]s from UK health & HAs
No. of student[ship]s from other overseas
No. of student[ship]s from others
No. of grants from ABRC res. councs et al
No. of grants from UK based charities
No. of grants from UK central government
No. of grants from UK local government
No. of grants from UK public corporations
No. of grants from UK industry & commerce
No. of grants from UK health & HAs
No. of grants from EC
No. of grants from other overseas
No. of grants from PCFC/NAB initiatives
No. of grants from TC schemes - RCs
No. of grants from TC schemes - company
No. of grants from other
Value of grants from ABRC res. councs et al
Value of grants from UK based charities
Value of grants from UK central government
Value of grants from UK local government
Value of grants from UK public corporations
Value of grants from UK industry & commerce
Value of grants from UK health & HAs
Value of grants from EC
Value of grants from other overseas
Value of grants from PCFC/NAB initiatives
Value of grants from TC schemes - RCs
Value of grants from TC schemes - company
Value of grants from other
Usage of reseach council facilities (pounds)
No. of published authored books
No. of published edited books
No. of published short works
No. of published refereed conferences
No. of published other conferences
No. of published editorships
No. of published art.s for academic journals
No. of published art.s for profess. journals
No. of published art.s for popular journals
No. of published reviews of academic books
No. of published other publications
No. of published misc publications
No. of staff producing publications
No. of selected staff on payroll
No. of staff on payroll
Proportion of staff submitted (letter)
FTE staff submitted
Research rating for applied research
Research rating

[ Note: FT=full-time; PT=part-time; FTE=full-time-equivalent; TC=Teaching Company ]
```

Table 4.1: The variables supplied by each department for the 1992 RAE. Note that some of the fields had to be supplied for individual years.

There are in the order of 95 distinct data types, and with some types (such as the numbers of publications) being supplied for individual years, there is a total of nearly 150 fields in the database. The research rating for each specific subject area at each institution is also supplied with the data.

The award of the research rating is partly dependent on this quantitative data, but critically, the final decision depends on the judgement of an assessment panel, composed of a number of experts in the relevant fields. In addition to the numerical data, each institution supplied two specimen publications per academic for consideration. This peer assessment component may be regarded as adding considerable noise to the data, with respect to the prediction of research ratings from the quantitative indicators alone.

Some statistical studies of this data have been previously published, both for the 1989 exercise [Johnes, Taylor, and Francis 1993] and for that of 1992 [Taylor 1994; Taylor 1995]. The data available from the 1989 exercise is less comprehensive and reliable than that from 1992. Johnes, Taylor, and Francis [1993] nevertheless observe that size has a positive influence on research quality, and also note a ‘halo’ effect, where a particular department at an institution is awarded a higher research rating than might otherwise be justified by the quantitative data alone and where other departments at that same institution have obtained justifiably high ratings. The analysis of Taylor [1994] was based upon the 1992 data and was restricted to Business and Management schools. This was later extended to all subject areas [Taylor 1995]. Again, size was found to be a statistically significant explainer of rating, along with three other variables: the number of articles in refereed academic journals per capita, the number of full-time equivalent postgraduates per capita and the amount of research council grants per capita.

It should be emphasised that the research rating is intended as a measure of the *quality* of research, and not its *quantity*. For a given research rating, departmental funding is proportional to the number of active research staff, so *funding* will be related to the size of the department even though the award of the *research rating* should not be. There is therefore no inherent reason for a large department to score more highly than a smaller one. Nevertheless, the observations of the previous paragraph still apply, that “research quality improves with size” [Taylor 1994] and suggested explanations for this effect are offered by Taylor [1995]. Further evidence for this assertion was also derived through a factor analytic study by Barrett [1995].

From an academic perspective, the RAE has aroused great interest, debate and not inconsiderable controversy, with much comment passed that it is wasteful of resources, unfair, will ultimately erode quality through its quantitative emphasis, and encourages short-termism. Indeed, there has been considerable discussion in the national press, for example [Crace 1995; MacLeod 1995], on its implications. This chapter is not able to address these wider issues, but can question the value of the quantitative data and whether it is truly informative given the various panels’ ultimate qualitative assessments through the ratings awarded.

4.2.2 Extracting Experimental Data from the RAE Database

It is of natural interest to investigate the relationships and structure within the RAE database. The prospect of being able to predict the value of research rating from the quantitative data alone, and thus infer which indicators are salient, has been the prime motivation for its statistical study. Furthermore, it is in the interest of the assessment bodies to confirm that the data that is being collected (at the cost of considerable time and expense) is both valid and a useful contribution to the exercise. Indeed, the funding councils themselves have commissioned research on the data.

The challenge of elucidating structure from within the raw RAE dataset is actually quite demanding. If all the departments are included together, there are over 4000 data points in close to 150 dimensions. However, treating these points as one large monolithic dataset is not a sensible approach to data analysis. Firstly, the distribution of the data will vary significantly across subject areas, which is evident from simple examination of the raw data values. For example, consider the likely contrast between “number of published authored books” and “value of grants from health authorities” for subjects such as Celtic Studies and Biological Sciences. Secondly, the importance of these individual variables in determining ratings will also exhibit considerable differences from department to department, as a separate assessment panel sits in each case. Statistical evidence for these effects has been offered by Taylor [1995].

It is thus appropriate to study the database on a department-by-department basis. Unfortunately, for each subject are there are limited exemplars (typically 50-90 in most cases), which represents fewer data points than dimensions. This is a major disadvantage, particularly if it is desired to build a classifier of the data. A sensible approach, therefore, would be to seek a number of subject areas which exhibited similar structure in the data, and where the relationships between the variables and the awarded research ratings were fairly constant. Correlation analysis by the Joint Performance Indicators Working Group [1994] indicates that one such appropriate grouping would be to combine the data from Physics, Chemistry and Biological Sciences, which gives a dataset of 217 examples that has been chosen for analysis within this chapter. In addition, data from Applied Maths is also well correlated (although to a lesser degree), and these 67 examples will also be utilised as a test sample for the purposes of assessing generalisation of various models.

Although it was desired that no prejudice be exerted over the choice of variables to be retained for any subsequent projection, it was possible to remove several redundant and repeated indicators. It was also determined to agglomerate those variables given for individual years over the period they encompassed. This approach reduced the dimensionality of the input data to 80, which is nevertheless high, particularly in comparison with only 217 available examples.

The data was standardised for size by dividing throughout by the number of staff in the department (specifically CATEGORY_A + CATEGORY_B staff). This is intuitively sensible and has been the approach adopted in other studies, including that of the Joint Performance Indicators Working Group [1994]. The variables thus become of the form 'number of published short works per researcher', for example. This is an imposition of the prior knowledge that research ratings are quality-based. Although empirical evidence exists that research rating is related to size, this is likely to be an indirect effect, as hinted at by Taylor [1995]. For example, larger departments may be more productive due to economies of scale, or may have expanded as a result of research success attracting new funding and boosting staff levels.

Finally, because of the inhomogeneity of the input variables, and the wide ranges thereof, each such variable was normalised to zero mean and unit variance.

4.3 Feature Extraction, Neural Networks and NEUROSCALE

In practical application neural networks can fulfil a variety of rôles. Common examples are for classification, nonlinear regression, function approximation, auto-encoding, topographic mapping (as in the case of NEUROSCALE) or time series prediction. In all of these diverse incarnations, the neural network can be viewed as performing some variety of *feature extraction* process. This interpretation provides a helpful unifying perspective when comparing neural network approaches with more classical techniques.

'Feature extraction' is a term for the process of deriving alternative (and usually lower-dimensional) representations of data that are more appropriate for a given specific application. While it might appear intuitive that access to more input features should improve performance, the inclusion of noisy or redundant features in a model can easily lead to a degradation thereof [Weiss and Kulikowski 1991]. For example, in the case of limited data, an additional feature that is irrelevant (in terms of classification, say) will simply exacerbate over-fitting as it can be interpreted as acting as a key to a look-up table.

One example of feature extraction is in a classification scenario where data may be projected linearly onto those axes that best separate the classes according to some criterion. This is a case of a *linear, supervised* extraction process, and it is convenient to group feature extraction techniques in general according to their linear–nonlinear and supervised–unsupervised nature. This categorisation has been adopted in three recent studies [Mao and Jain 1995; Lowe and Tipping 1995; Lowe and Tipping 1996].

The classical linear, unsupervised feature extraction technique is *principal components analysis* (PCA) [Jolliffe 1986]. The first q principal components of a set of data points are those obtained by the projection of the p-dimensional data onto the q orthogonal axes that retain the maximum variance. The projection vectors may be easily found as the dominant eigenvectors of the data (sample) covariance matrix, or alternatively, there are numerous neural network architectures specifically designed to extract principal components and subspaces (see Chapter 5 for some examples). There are also nonlinear extensions of PCA, both within the neural network domain [Saund 1989; Kramer 1991] and from a statistical viewpoint [Hastie and Stuetzle 1989; LeBlanc and Tibshirani 1994], as well as a hybrid of the two [Dong and McAvoy 1996]. A more complex class of unsupervised methods are those incorporated in *projection pursuit* [Friedman and Tukey 1974; Huber 1985]. These schemes look to maximise alternative criteria under projection, for example, higher-order statistics such as skew and kurtosis.

All the techniques mentioned in the above paragraph utilise the data in isolation, without any explicit class information, and are thus considered to be *unsupervised.* If alternative information is available, most commonly in the form of class labels, then this may be exploited in the extraction process to produce *supervised* feature spaces more appropriate for subsequent classification. This concept was originally considered by Fisher [1936] in the search for a linear projection of the measured variables that maximised the separation of two classes, or groups, from within the Iris dataset. His empirical approach was to maximise the quotient of the squared group-mean difference and within-group variance in the projection. This may be formulated as $(\mathbf{a}^{\mathrm{T}}\mathbf{S}_b\mathbf{a})/(\mathbf{a}^{\mathrm{T}}\mathbf{S}_w\mathbf{a})$, where $\mathbf{S}_b$ and $\mathbf{S}_w$ are the *weighted between-group covariance matrix* and *within-group pooled covariance matrix* respectively, with $\mathbf{a}$ the projection vector. This ratio may be maximised by setting $\mathbf{a}$ to the dominant eigenvector of $\mathbf{S}_w^{-1}\mathbf{S}_b$. The resultant projection is known *Fisher's linear discriminant function*, and although it was designed simply to be "sensible" and makes no explicit assumptions of the form of the class conditional densities, it may be shown to be equivalent to a *maximum likelihood* discriminant rule for two classes characterised by normal distributions with identical covariance matrices [Mardia, Kent, and Bibby 1979].

Fisher's approach was generalised to the discrimination of more than two groups by Rao [1948] and Bryan [1951]. In the case of generalised *linear discriminant analysis* (LDA), an appropriate criterion for discrimination is $\mathrm{tr}\left[\hat{\mathbf{S}}_w^{-1}\hat{\mathbf{S}}_b\right]$, where $\hat{\mathbf{S}}_w$ is now the within-group pooled covariance in the *transformed* space, and likewise, $\hat{\mathbf{S}}_b$ is the transformed counterpart of $\mathbf{S}_b$. Such a criterion naturally increases as the within-group variances become smaller and the between group variances become larger. For this measure, the projection vectors $\mathbf{a}_i$ may be found by solution of the generalised eigenvector equation $\hat{\mathbf{S}}_b\mathbf{a}_i = \lambda\hat{\mathbf{S}}_w\mathbf{a}_i$, and the resultant discriminant axes are often referred to as the *canonical variates*. For a k-class problem, all the discriminatory information, as measured by $\mathrm{tr}\left[\hat{\mathbf{S}}_w^{-1}\hat{\mathbf{S}}_b\right]$, may be retained by projection onto $\min(p, k-1)$ eigenvectors.

In the neural network domain, it has been demonstrated by Gallinari et al. [1991] that the hidden layer space of a *linear* neural network trained as a classifier maximises the criterion $|\hat{\mathbf{S}}_b|/|\hat{\mathbf{S}}_t|$, where $\hat{\mathbf{S}}_t = \hat{\mathbf{S}}_w + \hat{\mathbf{S}}_b$ is the *total covariance matrix* measured in the feature space, which generates an equivalent projection to $\mathrm{tr}\left[\hat{\mathbf{S}}_w^{-1}\hat{\mathbf{S}}_b\right]$ [Fukunaga 1990]. Again, there are nonlinear extensions of this approach, with the hidden unit space of a linear output-layer multi-layer perceptron having been shown to maximise the *network discriminant function* $\mathrm{tr}\left[\hat{\mathbf{S}}_t^{+}\hat{\mathbf{S}}_b\right]$ for a particular target coding scheme [Webb and Lowe 1990]. This again gives an equivalent projection to the $\mathrm{tr}\left[\hat{\mathbf{S}}_w^{-1}\hat{\mathbf{S}}_b\right]$ criterion [Fukunaga 1990].

Topographic mappings, such as those generated by the Kohonen or Sammon Maps, may be viewed as nonlinear, unsupervised feature extraction processes. Here the criterion for selection of features is not to maximise variance or class separability, but rather that the topology, or geometric structure, of the data be preserved in the feature space. Naturally, NeuroScale with $\alpha = 0$ also fits into this category. Alternatively, as discussed in Chapter 3, NeuroScale with $\alpha = 1$ may be considered a supervised technique, as full subjective (or class) information is exploited in the mapping. However, only with $\alpha = 1$, and in certain special cases, may it be interpreted as a form of discriminant analysis. For example, consider a mapping with subjective dissimilarities set to zero for data points in the same class ($\delta_{ij} = 0 : \omega_i = \omega_j$) and set to some constant value for data points in different classes ($\delta_{ij} = \lambda :

$\omega_i \neq \omega_j$). In this case, the STRESS measure is

$$\sum_{ij}(\delta_{ij} - d_{ij})^2 = \sum_{\omega_i=\omega_j} d_{ij}^2 + \sum_{\omega_i\neq\omega_j} (\lambda - d_{ij})^2, \tag{4.1}$$

$$= \sum_{\omega_i=\omega_j} d_{ij}^2 + \sum_{\omega_i\neq\omega_j} (\lambda^2 - 2\lambda d_{ij} + d_{ij}^2), \tag{4.2}$$

$$= \sum_{ij} d_{ij}^2 - 2\lambda \sum_{\omega_i\neq\omega_j} (d_{ij} - k), \tag{4.3}$$

$$= 2N\mathrm{tr}\,[\mathbf{S}_t] - 2\lambda \left(\sum_{\omega_i\neq\omega_j} [(\mathbf{y}_i - \mathbf{y}_j)^{\mathrm{T}}(\mathbf{y}_i - \mathbf{y}_j)]^{1/2} - k \right), \tag{4.4}$$

where $k = \frac{1}{2}\sum_{\omega_i\neq\omega_j} \lambda$ and is constant. Equation (4.4) is very similar to one particular discriminant criterion, $\mathrm{tr}\,[\mathbf{S}_t] - \mu(\mathrm{tr}\,[\mathbf{S}_b] - c)$, where μ and c are constants [Fukunaga 1990, pp447]. The presence of the square root in equation (4.4) unfortunately prevents any further simplification with respect to this standard measure, but it may be heuristically interpreted as minimising the total covariance while retaining some measure of the between-group spread at a constant value.

However, in general even for $\alpha = 1$, there is not such a close relationship to discriminant analysis, as the subjective dissimilarities will not be as simplistic as in the above example.

It can therefore be intuitive to view the parameter α as interpolating in some manner between an unsupervised mapping and a supervised variant — a variant which is related closely to the philosophy of multidimensional scaling because the topography of the feature space is then determined by the subjective information, and not explicitly by the spatial distribution of the data.

The collection of techniques discussed above may be related by the taxonomy in figure 4.1, which categorises some of the aforementioned feature extraction schemes and places the NEUROSCALE approach in that context also. The techniques illustrated in that diagram will be applied to the analysis of the RAE dataset in the following section.

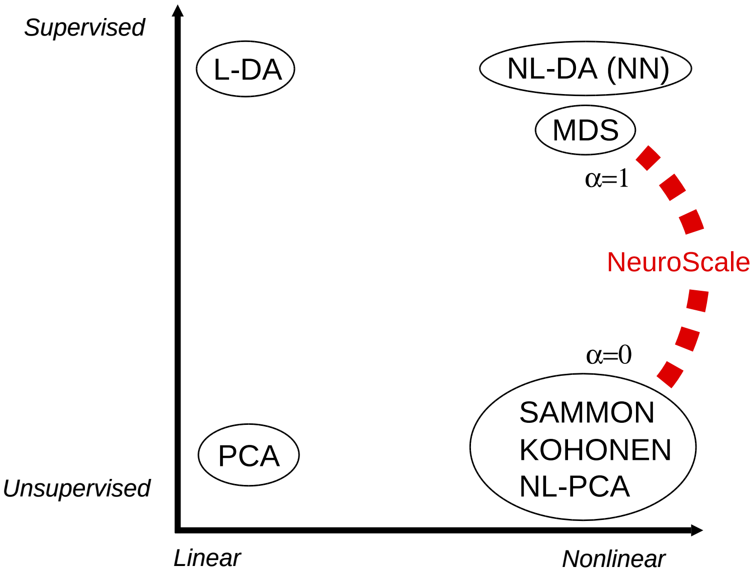


Figure 4.1: A schematic of feature extraction approaches.

4.4 Experiments on the RAE Data

For the purposes of brevity in this section, the combined Physics, Chemistry and Biological Science set of 217 data points will be referred to as the RAE_PCB dataset, with the 67 example data points from Applied Maths referred to as RAE_AM.

4.4.1 Principal Components Analysis

As the data in RAE_PCB is 80-dimensional, a sensible first step is to perform a principal components analysis (PCA). If the data is actually residing for the most part in a lower dimensional linear subspace, then this will be exposed by PCA, and the data may be projected down onto a reduced number of principal axes, without significant loss of information. These axes are the principal eigenvectors of the sample covariance matrix of the data, with the variance projected onto each axis given by the corresponding eigenvalue. Axes along which the projected variance is deemed negligible, where the eigenvalues are small compared with the principal eigenvalues, may be judiciously discarded.

The eigenvalues of the covariance matrix of the RAE_PCB dataset are given in figure 4.2.

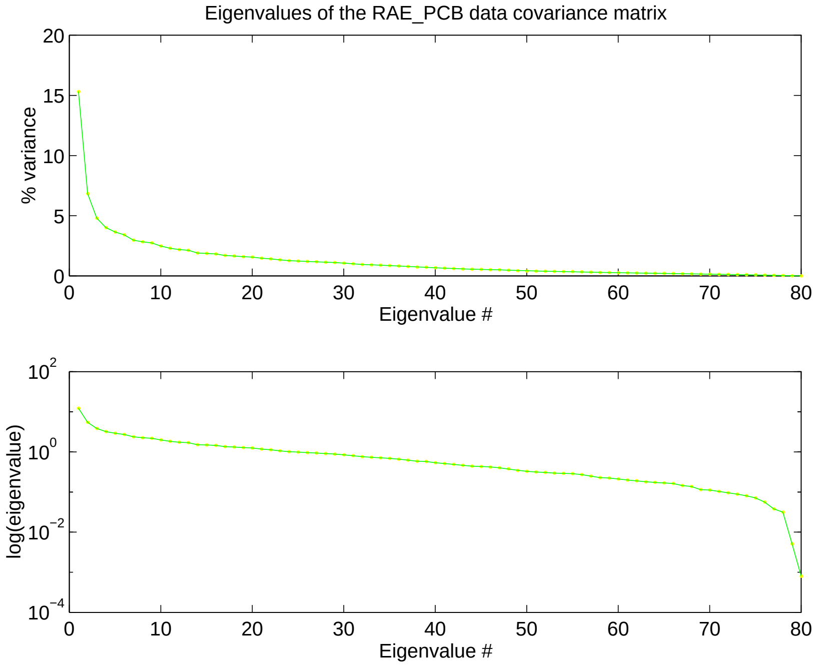


Figure 4.2: The eigenvalues of the RAE_PCB covariance matrix.

After the first two principal eigenvalues, there is a gradual decay in magnitude, with even the 32-nd dimension contributing greater than 1% of the overall variance. The large principal eigenvalue is characteristic of covariance matrices where the majority of variables are positively correlated, and can be considered a measure of the *size* of those variables [Chatfield and Collins 1980]. This is the case for the RAE_PCB dataset, as can be seen by the illustration of the sample covariance matrix in figure 4.3. This effect aside, in both the linear and logarithmic eigenvalue plots, the data does not exhibit the characteristic fall-off associated with inherently lower-dimensional data and there is no suggested cut-off point for discarding dimensions.

The feature space defined by a projection onto the two principal axes is given in figure 4.4. Very little

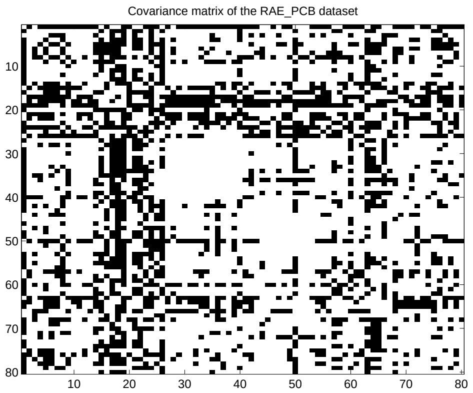


Figure 4.3: A schematic of the covariance matrix of the RAE_PCB dataset. White squares are positive elements, black squares negative.

information concerning the data can be elucidated from this plot, apart from the existence of some outliers with rating '1' at one extreme of the distribution. The linear nature of the projection implies that most of the structure in the data will be lost, particularly as over 78% of the variance is present in the remaining components. This is the major restriction of PCA, particularly when applied to such high-dimensional data, and in general it is best employed to seek degeneracies in the data as a preliminary to other techniques.

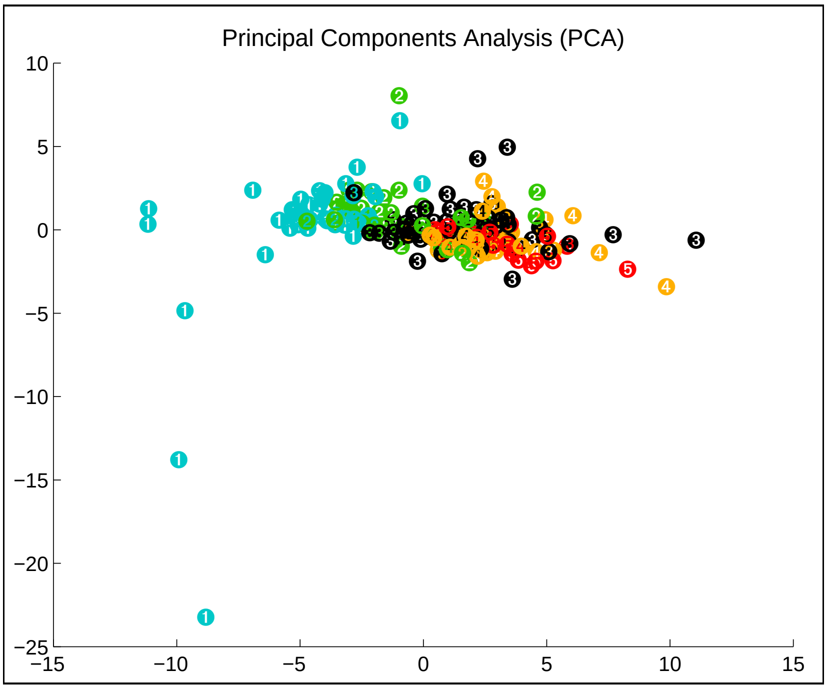


Figure 4.4: Projection onto first two principal axes of RAE_PCB.

4.4.2 Classification of the Data

Before considering alternative spatial representations of the data, it is of interest to consider a classification of RAE_PCB with respect to the research rating. Is it possible to make reliable predictions of the research rating awarded by the panel from the quantitative data alone? Even more importantly, is there any relationship at all between research rating and the gathered data?

Three different prediction models were considered, all fully connected to the 80 inputs and trained to produce a 1-of-5 output coding:

① Linear Model,

② Multilayer Perceptron Neural Network (MLP),

③ Radial Basis Function Neural Network (RBF).

The results for each model are given below, in terms of prediction error and misclassification matrices for both training and test sets. Each row of a misclassification matrix represents the actual class, and each column the prediction. So in the first such matrix below, the '1' in row 2 column 4 indicates a university that actually received a research rating of 2 but was predicted to gain a 4. Note that, for accurate estimating of prediction error of a model, the test set should be sampled from the same distribution as the training set [Lowe and Webb 1990; Lippmann 1994]. (That is, the prior probabilities for the respective classes should be identical for each set.) This assumption will be made for the RAE_PCB and RAE_AM datasets here, although study shows that, compared to RAE_PCB, RAE_AM has an exaggerated number of rating '1's and a reduced proportion of '2's. The class sizes are, in order, for RAE_PCB: (42,42,62,44,27) and for RAE_AM: (6,21,18,14,8). However, to alter the model based on this knowledge (through a weighting of the error function) would be "cheating" in this instance.

① *Linear Model*

The classification results were:

□ Training Set : LINEAR MODEL □

```
MSE per output:          0.0360696
Patterns Misclassified: 33 (out of 217)
Percentage Correct:      84.8%

Classification Matrix
====================
 40  2  0  0  0
  4 35  2  1  0
  3  5 51  2  1
  0  2  3 36  3
  0  0  1  4 22
```

■ **Test Set : LINEAR MODEL** ■

```
MSE per output:          0.106999
Patterns Misclassified: 41 (out of 67)
Percentage Correct:      38.8%

Classification Matrix
=====================
  5  1  0  0  0
  6  6  7  1  1
  1  6  7  4  0
  1  2  5  3  3
  0  0  1  2  5
```

② *Multilayer Perceptron*

The architecture of the MLP was 80-*h*-5, where *h* is the number of hidden units in the network. The hidden-unit activation functions were hyperbolic tangents (tanh) and the output neurons were linear. To determine a near-optimal number of hidden units, training and test errors were evaluated for 1-12 units, with the errors averaged over 25 runs with different random initial weight configurations. Weights were optimised with a conjugate-gradient routine. Plots of sum-of-squared training errors and test misclassification rate against number of hidden units is given in figure 4.5 below. Standard deviation error-bars are also shown.

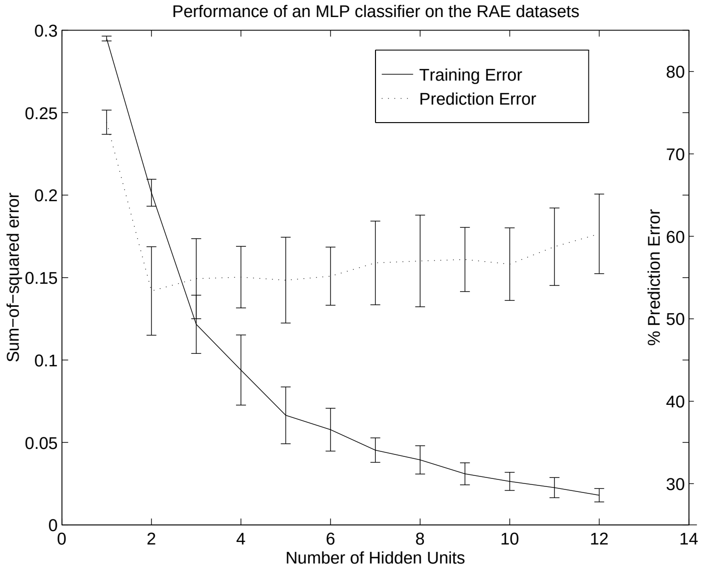


Figure 4.5: An MLP classifier trained on the RAE_PCB data and tested on the RAE_AM set.

In terms of misclassification rate, a network with 2 hidden units is optimal in this experiment:

□ **Training Set : MLP (2)** □

```
MSE per output:         0.0402512
Patterns Misclassified: 37 (out of 217)
Percentage Correct:     82.9%

Classification Matrix
=====================
 42  0  0  0  0
  3 39  0  0  0
  0  1 61  0  0
  0  0  6 38  0
  0  0  0 27  0
```

■ **Test Set : MLP (2)** ■

```
MSE per output:         0.0802902
Patterns Misclassified: 35 (out of 67)
Percentage Correct:     47.8%

Classification Matrix
=====================
  5  0  0  1  0
  7 12  0  2  0
  5  4  8  1  0
  0  1  6  7  0
  0  0  1  7  0
```

For a network with 12 hidden units, to illustrate over-fitting.

□ **Training Set : MLP (12)** □

```
MSE per output:         0.00207232
Patterns Misclassified: 0 (out of 217)
Percentage Correct:     100%

Classification Matrix
=====================
 42  0  0  0  0
  0 42  0  0  0
  0  0 62  0  0
  0  0  0 44  0
  0  0  0  0 27
```

■ **Test Set : MLP (12)** ■

```
MSE per output:         0.190793
Patterns Misclassified: 42 (out of 67)
Percentage Correct:     37.3%

Classification Matrix
=====================
  4  1  0  1  0
  5  6  8  0  2
  1  5  8  4  0
  0  0  7  4  3
  0  0  2  3  3
```

③ *Radial Basis Function Network*

For the radial basis function network, the linearity of the training problem implied by choosing fixed centres permits an efficient implementation of ‘leave-one-out’ cross-validation [Weiss and Kulikowski

1991] for model-order selection. In figure 4.6 below, the leave-one-out cross-validation error on the RAE_PCB dataset is calculated for networks with 1 to 20 basis functions. The error when generalising to the data in RAE_AM is also plotted. The basis functions used were 'thin-plate splines' ($r^2 \log r$), and they were positioned at random over points in the dataset and then adjusted by the batch version of the K-means algorithm [MacQueen 1967; Moody and Darken 1989]. In this algorithm, the centres are initially located at random on the data points. At each iteration, every data point is 'assigned' to its nearest centre, after which the centres are then adjusted such that each lies at the mean of the data points which were previously assigned to it. This procedure is then repeated until there is no change in the centre positions. K-means is often described as a "greedy" approach to vector quantisation, in that it is fast and generates a fairly good, although generally not globally optimal, solution, by performing an equivalent local Newton minimisation of the quantisation error [Bottou and Bengio 1995].

All error values were averaged over 100 runs. To reduce confusion, error-bars have not been plotted on the graph, but the standard deviations were in the order of 0.008 for the training set and 0.011 for the test set.

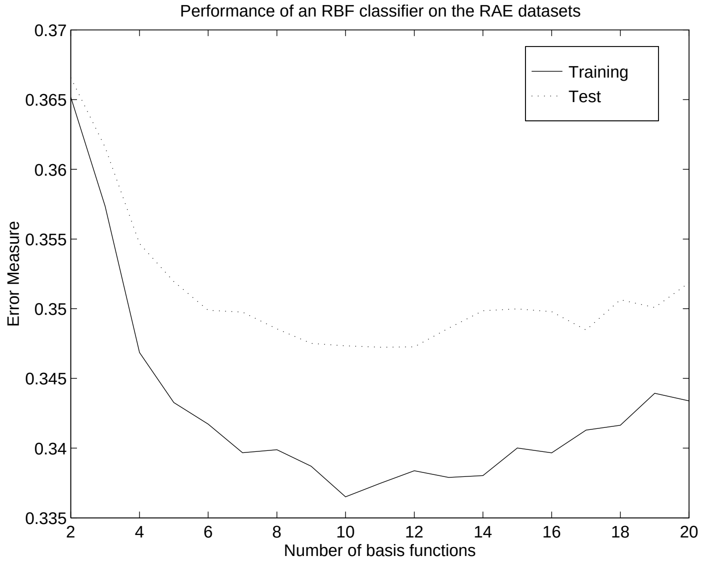
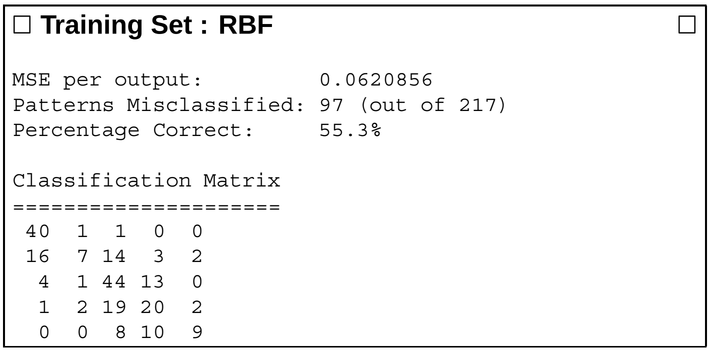


Figure 4.6: An RBF classifier trained on the RAE_PCB data and tested on the RAE_AM set. Training error is calculated using leave-one-out cross-validation.

The classification performance of a typical RBF network, with 10 basis functions, is:



```
MSE per output:          0.0620856
Patterns Misclassified: 97 (out of 217)
Percentage Correct:      55.3%

Classification Matrix
=====================
 40   1  1   0  0
 16   7 14   3  2
  4   1 44  13  0
  1   2 19  20  2
  0   0  8  10  9
```

■ **Test Set : RBF** ■

```
MSE per output:          0.0712626
Patterns Misclassified: 37 (out of 67)
Percentage Correct:      44.8%

Classification Matrix
=====================
  6  0  0  0  0
  9  2  8  2  0
  0  1 14  3  0
  0  0  7  5  2
  0  0  3  2  3
```

Discussion of Classification Results

The linear model appears to perform well on the training dataset, classifying nearly 85% of the patterns correctly. However, there are very few available examples compared to their dimensionality (217 against 80), so the linear problem is not well-constrained and an apparently ‘good’ classification might be expected regardless of the suitability of the model. The considerably degraded performance on the test set confirms this. Note that simply ‘guessing’ a rating of ‘3’ (the most prevalent class in the training set) on the test set would give 27% accuracy and is thus a good baseline for comparison.

A notable feature of the training / test error curves in figures 4.5 and 4.6 is the very low implied optimal classifier complexity. The number of hidden units in the MLP for lowest test set error was the minimum, 1, and the optimal number of basis functions appears to be 10. This apparently low number is a manifestation of a problem often referred to as *the curse of dimensionality* (a term originally coined by Bellman [1961], and an excellent exposition of its significance is given by Friedman [1994]). For genuinely high dimensional data (which the PCA results indicate is certainly the case for the RAE data considered here), the number of sample data points required to allow a faithful approximation of any underlying function is massive. In cases of inadequate data, such as for the RAE_PCB dataset, generalisation effectively becomes *extrapolation* (rather than *interpolation*), as it is highly improbable that any training data should be in the vicinity of any given test data point. In the absence of sufficient data, considerable smoothness constraints must be imposed on the model — hence the limited number of basis functions, and MLP hidden units, found to give optimal generalisation. This behaviour is an example of what is often termed the *bias-variance dilemma* [Geman, Bienenstock, and Doursat 1992]. It is necessary to bias the model (through the smoothness constraints), or the resulting function will be highly sensitive to the noise, and thus exhibit high variance over individual datasets.

The (nonlinear) neural network models perform better than their linear counterpart, classifying 45-48% of the test set correctly, compared to that of 39% of the linear classifier, which indicates that a more faithful model of the underlying relationship between data and research rating has been constructed. However, the generally poor observed performance may be seen as reinforcing the suspicion that the operation of the assessment panel would effectively introduce noise into the data, when viewed from a classification perspective. This will be considered further in Section 4.5, where an explicit feature extraction stage will be incorporated prior to classification.

4.4.3 Sammon Mapping

A Sammon mapping of the RAE_PCB dataset is illustrated in figure 4.7. The equivalent NEUROSCALE projection with $\alpha = 0$ is also illustrated (figure 4.8). The two plots should be identical if one basis function per point was used in the RBF projection; in this example, only 70 basis functions were utilised, and consequently, the STRESS of the NEUROSCALE projection is marginally greater. For this dataset, it was observed that ‘thin-plate-spline’ basis functions offered better (lower-STRESS) results than Gaussians, and these were used for all the NEUROSCALE projections within this chapter.

There is some notable similarity between the Sammon Mapping and the PCA projection, particularly with regard to the position of outliers. The nonlinearity of the topographic mapping algorithm enables significantly more structure to be visible, particularly along the direction of the vertical axis. Some coarse clustering of research ratings is now evident, although the classes overlap to the extent that prediction would be highly unreliable.

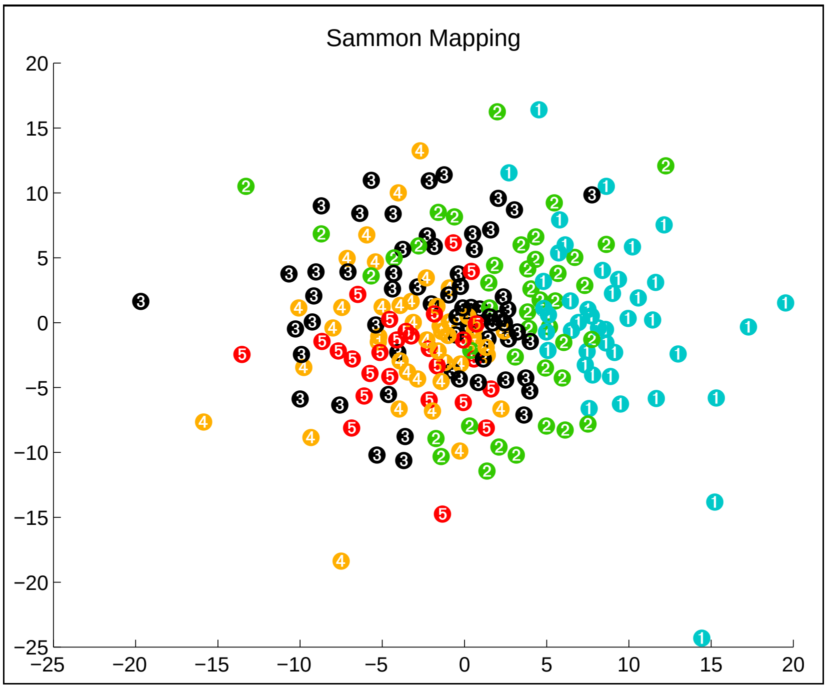


Figure 4.7: A Sammon Mapping of the RAE_PCB data. Final STRESS value = 1.118

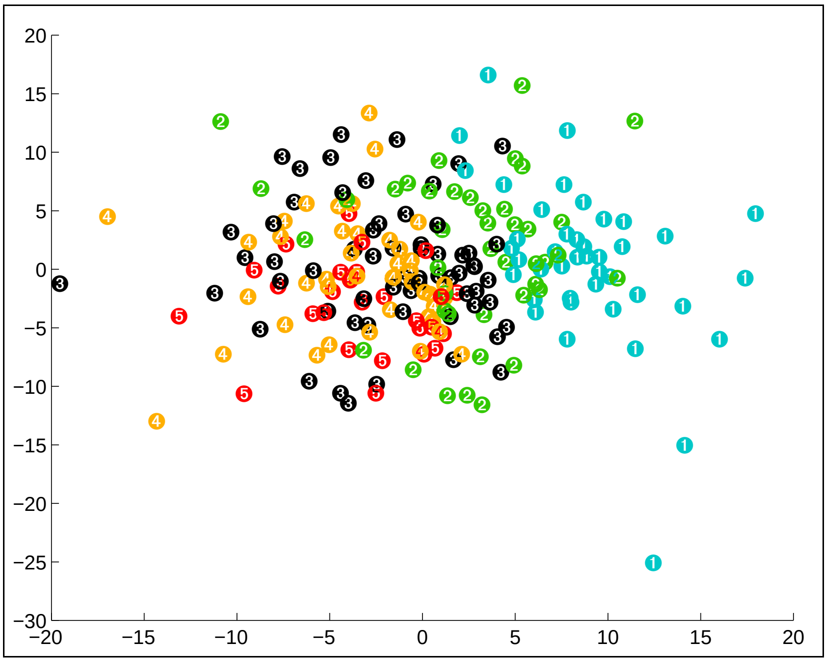


Figure 4.8: The NEUROSCALE equivalent projection for the Sammon mapping. Final STRESS value = 1.181

4.4.4 The Kohonen Self-Organising Feature Map

A 10x8 Kohonen map was applied to the RAE_PCB dataset, and the visualisation obtained using the SOM_PAK code [SOM Programming Team 1995] is displayed in figure 4.9. The nearest-neighbours to each data point in the two dimensional sheet are plotted, with multiple classes at any one node having an added displacement for the purposes of improving clarity.

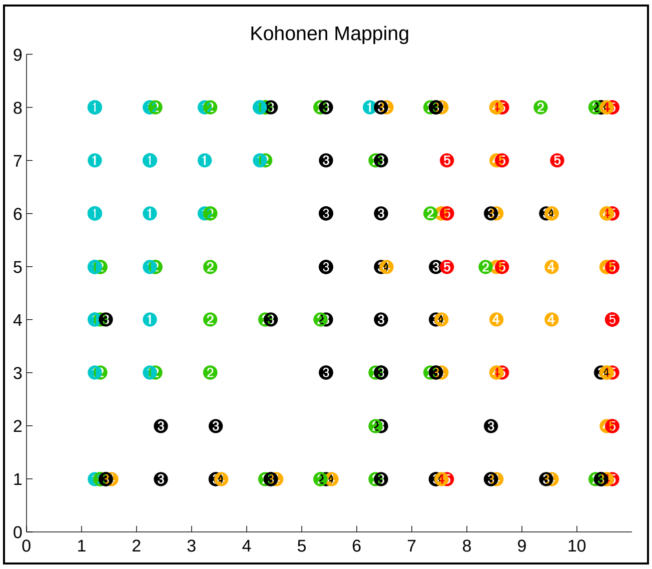


Figure 4.9: A visualisation of a Kohonen map applied to the RAE_PCB data.

The Kohonen map, like the Sammon mapping, displays little structure and considerable confusion of classes. At some nodes on the array, departments with four different research ratings are co-incident.

4.4.5 Discriminant Analysis

While the class labels of each data point are illustrated on both the principal component and Sammon projections, this information has not been exploited in the extraction of the feature spaces. If the separation of these classes is desired in feature space, then the methods of discriminant analysis may be applied. The plot in figure 4.10 is a plot onto the first two *linear* discriminant axes (these axes may also be termed the first two *canonical variates*). This represents a non-rigid linear projection which maximises the discriminant criterion tr $\left[\mathbf{S}_w^{-1}\mathbf{S}_b\right]$, as defined in Section 4.3 earlier.

A nonlinear extension of this technique may be illustrated by considering the hidden unit space of a multi-layer perceptron trained as a 1-of-5 classifier [Webb and Lowe 1990]. This plot is given in figure 4.11. Note that while the feature space was constructed to maximise the criterion tr $\left[\mathbf{S}_t^{+}\mathbf{S}_b\right]$, the matrix $\mathbf{S}_b$ is not the traditional weighted between-group covariance matrix for the target coding employed here. In fact, $\mathbf{S}_b = \sum_{k=1}^{5} n_k^2(\mathbf{m}_k - \bar{\mathbf{m}})(\mathbf{m}_k - \bar{\mathbf{m}})^{\mathrm{T}}$, where $\mathbf{m}_k$ is the mean of the hidden unit outputs for patterns from class k, $\bar{\mathbf{m}}$ is the overall mean of the hidden unit outputs, and n_k is the number of patterns in class (with research rating) k. The conventional $\mathbf{S}_b$ matrix is only weighted by the factor n_k, and not the squared value.

The linear discriminant plot reveals some considerable structure in the data. Significantly more distinct clustering is evident, when compared with previous unsupervised feature spaces, although there is still considerable mixing of research ratings at the borders. Furthermore, there is a visible structural *ordering* of classes within the plot.

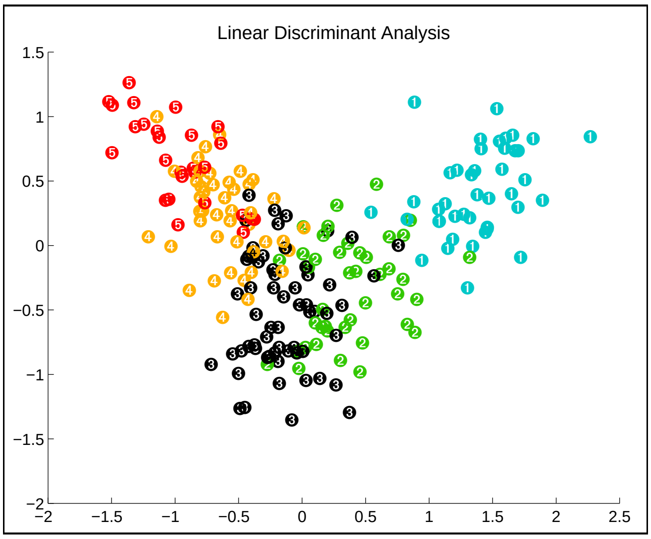


Figure 4.10: A plot of the first two canonical variates of the RAE_PCB data.

In the nonlinear case, to maximise the discriminant criterion the points have been approximately mapped to the corners of the -1/1 square. Some general misclassification is evident and points for research rating '4' and '5' occupy the same corner. The network discriminant function seeks to maximise the total covariance, while minimising a measure of the between-class covariance. Total covariance will be maximised by placing the classes as mutually distant as possible — in the corners — but, because there are only two hidden units in the MLP, only four such classes can be so placed without reducing the between-class covariance. The need to maximise the total covariance dominates in this case and hence the confusion of two classes. It is interesting to note that one of the pair is the least represented class, the research rating '5', because, according to Webb and Lowe [1990], "networks trained with a one-from-n coding bias strongly in favour of those classes with largest membership".

Note that the interpretation of this plot can be aided by referring to the equivalent misclassification matrix based upon the illustrated hidden unit space which is given in Section 4.4.2.

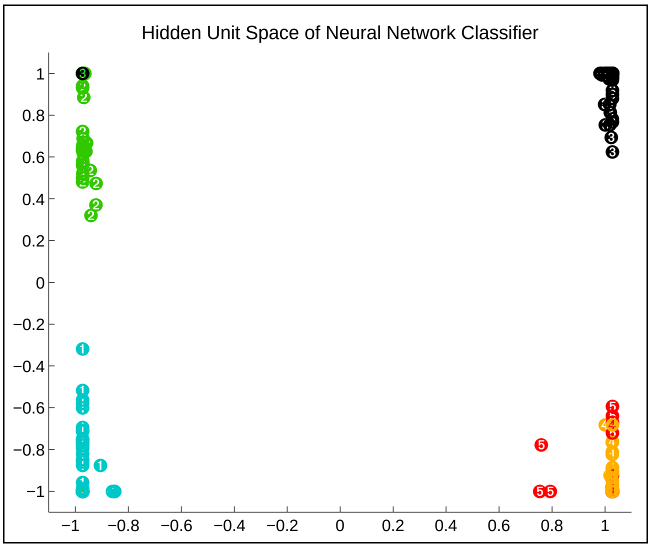
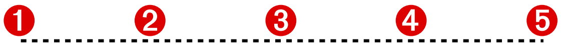


Figure 4.11: The hidden unit space of a MLP classifier trained on the RAE_PCB data.

4.4.6 NEUROSCALE with Supervisory Information

To incorporate some supervisorial, or preferential, input to the topographic projection, it is necessary to define the set of subjective dissimilarities for every pair of data points. These may in turn be derived from some measure of dissimilarity of the respective research ratings. The dissimilarities between research ratings are chosen to reflect both the prior knowledge concerning the assessment process and what is considered to be the preferred structure of the feature space. Based on the information available regarding the criteria for the assignment of ratings, it is intuitive to consider that a department with a rating of ‘2’ is as dissimilar from a ‘1’ as it is from a ‘3’, and furthermore, a ‘3’ is twice as dissimilar from a ’1’ as it is from a ’2’. These assumptions imply a geometrically linear ordering of research ratings thus:



which in addition, also reflects the resultant funding, since *funding* $\propto$ (*rating* $-$ 1). This in turn implies that a simple matrix may serve to characterise the subjective dissimilarities between research ratings:

$$\mathbf{C} = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 0 & 1 & 2 & 3 \\ 2 & 1 & 0 & 1 & 2 \\ 3 & 2 & 1 & 0 & 1 \\ 4 & 3 & 2 & 1 & 0 \end{bmatrix},$$

so that $c_{ij} = |i - j|$ is the subjective dissimilarity between points of rating i and j. Note that the scaling of this matrix is arbitrary, as it is only the *relative* differences that are of interest. Thus for example, entries $c_{24} = 2$ and $c_{25} = 3$ mean that the relative difference between departments with ratings ‘2’ and ‘4’ is 2, while the relative difference between departments with ratings ‘2’ and ‘5’ is 3. This considered, it is then sensible to scale the values in the matrix **C** such that the average inter-point subjective dissimilarity is equal to the average inter-point Euclidean distance. The matrix **C** then provides sufficient information to calculate a value of subjective dissimilarity, s_{ij}, between every pair of input data points, thus defining the subjective metric.

To illustrate a feature space influenced by both the objective (Euclidean distance in input space) and subjective metrics, figure 4.12 is a plot of the NEUROSCALE projection with $\alpha = 0.5$. Because of the scaling of the matrix **C**, a choice of $\alpha = 0.5$ implies a (very) approximate balance between the twin, objective and subjective, metrics.

The influence of the subjective metric is clearly evident by simple comparison with the Sammon Mapping. There is now a clear ordering of research ratings in a similar topology to that of the LDA projection. In contrast to that linear supervised feature space, careful examination shows that the inter-class boundaries are more pronounced in the NEUROSCALE plot. This may be expected, as the subjective metric element seeks to explicitly separate, nonlinearly, points with different research ratings.

The observations in the above paragraph concern the effect on the projection of the additional, supervisorial, knowledge. In addition to this subjective element, and because of the intermediate value of α (0.5), some of the geometric structure of the original data is retained in the feature space. This implies that useful information may be inferred from the locations of individual points in that space, as this structure reflects to some degree the topology of the input space. For example, in the plot of RAE_PCB in figure 4.12, it may illuminate potential anomalies in the awarding of research ratings. This considered, the plot in figure 4.13 highlights four particular departments in the projection. Each department appears to have received a rating incompatible with its position on the map, judged by consideration of the ratings awarded to its immediate neighbours in the feature space.

These departments are, from left to right on the plot:

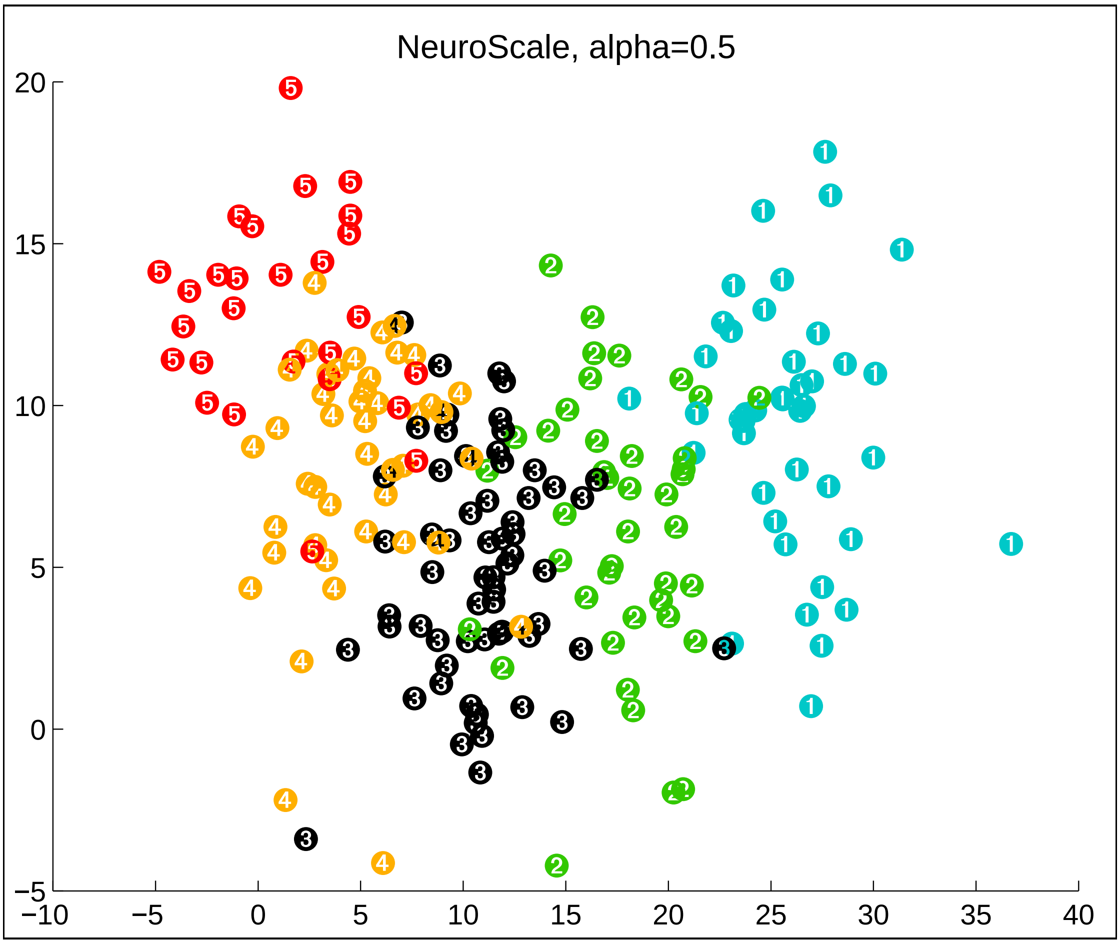


Figure 4.12: A feature space extracted by the NEUROSCALE technique with supervisorial influence.

① Physics at Heriot Watt University, Edinburgh, which has received a '5' while lying amongst a cluster of '4's.

② Physics at Queen's University, Belfast, which has also received a '5' while lying amongst departments awarded '4's and '3's.

③ Physics at Stirling University, which received a '4' while lying amongst a cluster of '3's.

④ Physics at the University of Westminster which was awarded a '3', while apparently located on the border between ratings '1' and '2'.

In the case of a purely supervised plot, for example the nonlinear discriminant analysis in figure 4.11, the location of individual points with respect to their neighbours is largely artefactual. Due to the topographic constraint upon the feature space of figure 4.12, there will be an element of structural information therein. In the cases of the individual points highlighted above, further evidence for the structural meaning can be elucidated by considering the classification results of section 4.4.2. Table 4.2 below shows the predicted ratings of the four departments above according to the best linear, MLP and RBF models.

In general, and particularly for the neural networks which demonstrated lower error on the test set, the classifier predictions support the evidence from the NEUROSCALE plot. In this example, the relative location of points in the feature space *has* proved informative. Although these four particular classifications may appear anomalous, there may well be good explanations. Firstly, it is noticeable

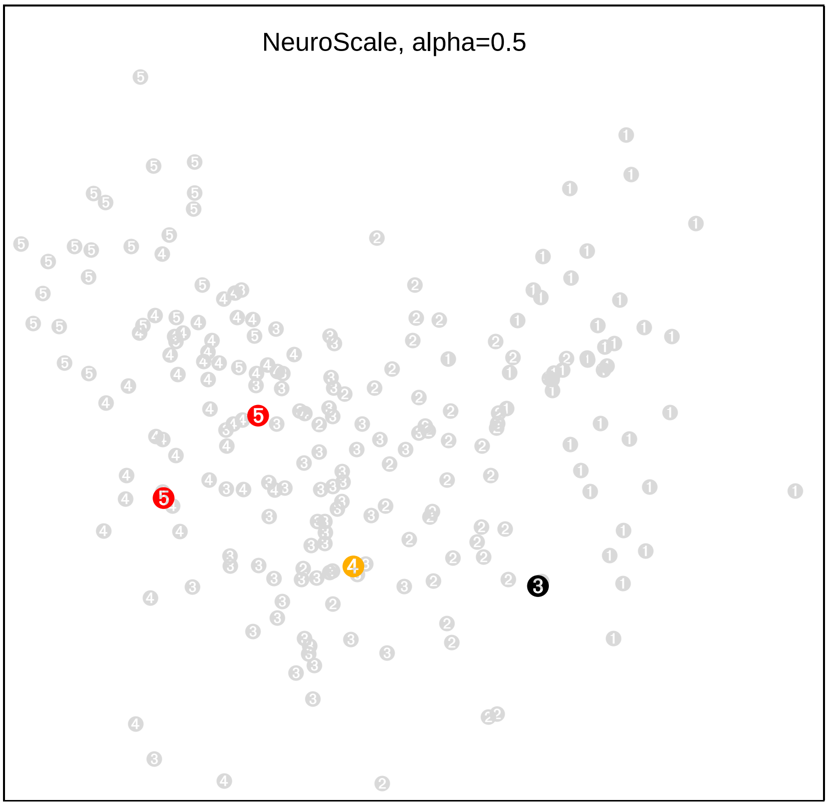


Figure 4.13: The $\alpha = 0.5$ feature space extracted by NEUROSCALE, with four departments highlighted whose awarded ratings appear anomalous.

that all four departments are of the Physics unit of assessment. The panel which awarded the ratings for this subject may have had slightly different criteria to those for Chemistry and Biological Sciences. Equally, the panel has access to additional qualitative information which, in the case of these four departments, may have influenced its judgement.

As a final example feature space generated by the NEUROSCALE technique, the illustration in figure 4.14 shows a plot for $\alpha = 1$. This feature space is no longer influenced (explicitly) by the spatial distribution of the input data, but is determined by the subjective metric alone. Thus the feature space should represent the preferential knowledge embodied in that metric, and should take the form of five point clusters distributed along a straight line.

The smearing out of the points along that line is a result of the RBF approximation to the Sammon mapping / scaling procedure. As the number of basis functions in the transformation is considerably fewer than the number of points, there is not sufficient flexibility in the model to precisely locate the points and satisfy the subjective metric constraint.

	Actual Rating	Linear	MLP	RBF
Physics, Heriot-Watt	*5*	5	4	3
Physics, Queen's	*5*	4	4	3
Physics, Stirling	*4*	4	4	3
Physics, Westminster	*3*	1	2	1

Table 4.2: Predicted and actual ratings for 'inconsistent' departments.

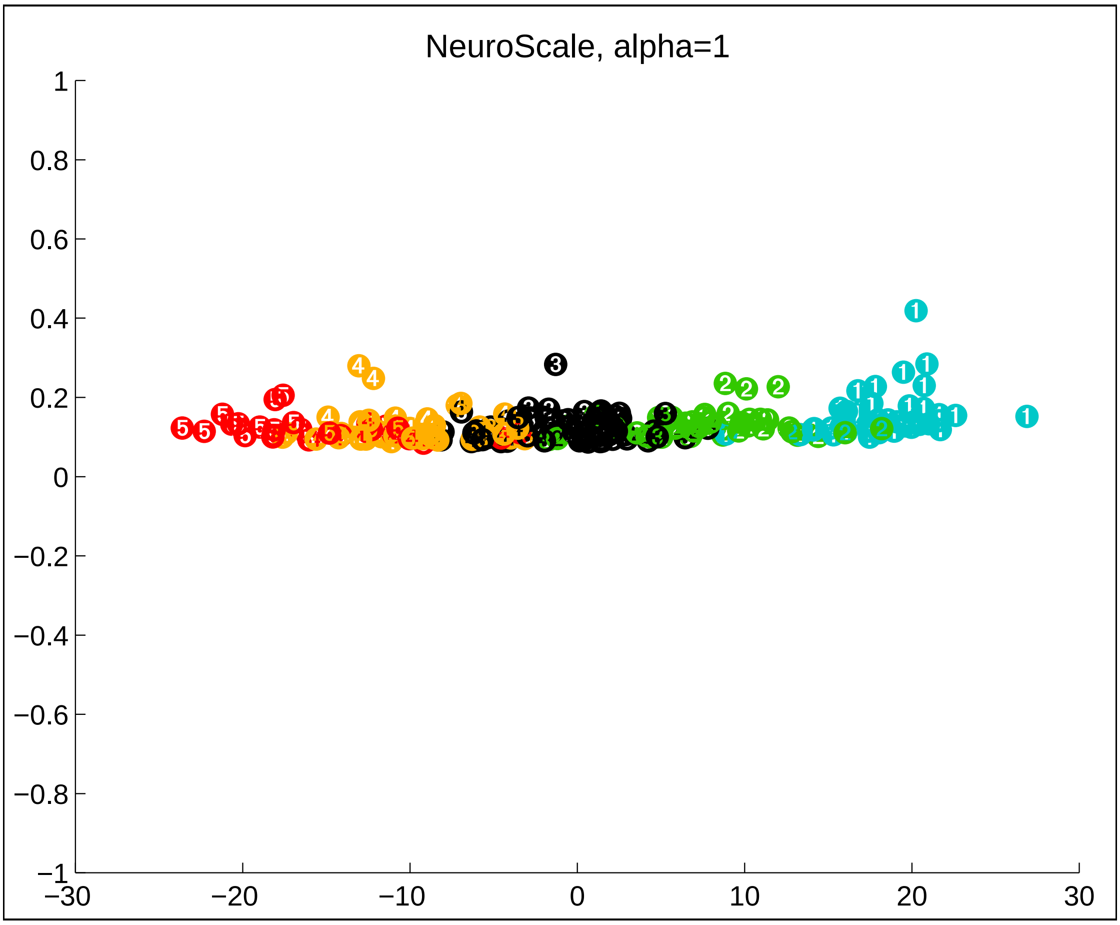


Figure 4.14: A feature space extracted by the NEUROSCALE technique determined by the subjective metric alone. This may be considered a fully supervised variant of the projection.

4.4.7 Generalisation to the Test Dataset

One of the key benefits of the NEUROSCALE approach to both topographic mapping and Sammon's projection is that the definition of the transformation, as effected by the RBF, permits new, unseen, data to be projected. Such a projection is illustrated in figure 4.15, for the $\alpha = 0.5$ experiment trained on the RAE_PCB dataset and tested on the RAE_AM set. As comparison, in figure 4.16, a projection of the test data onto the linear discriminant axes of figure 4.10 is also given.

As in the case of the training plot, for the RAE_PCB dataset, the NEUROSCALE projection exhibits better clustering and separation of research ratings. While there is still significant confusion at the borders, there is sufficient structure present in the plot to allow judicious inference of research ratings. Subjectively, the NEUROSCALE test projection appears to retain more of the structure of its respective training plot than the linear discriminant counterpart, indicating that this latter linear technique had not constructed such a good representation of the data. This contrast will be made more explicit in the next section.

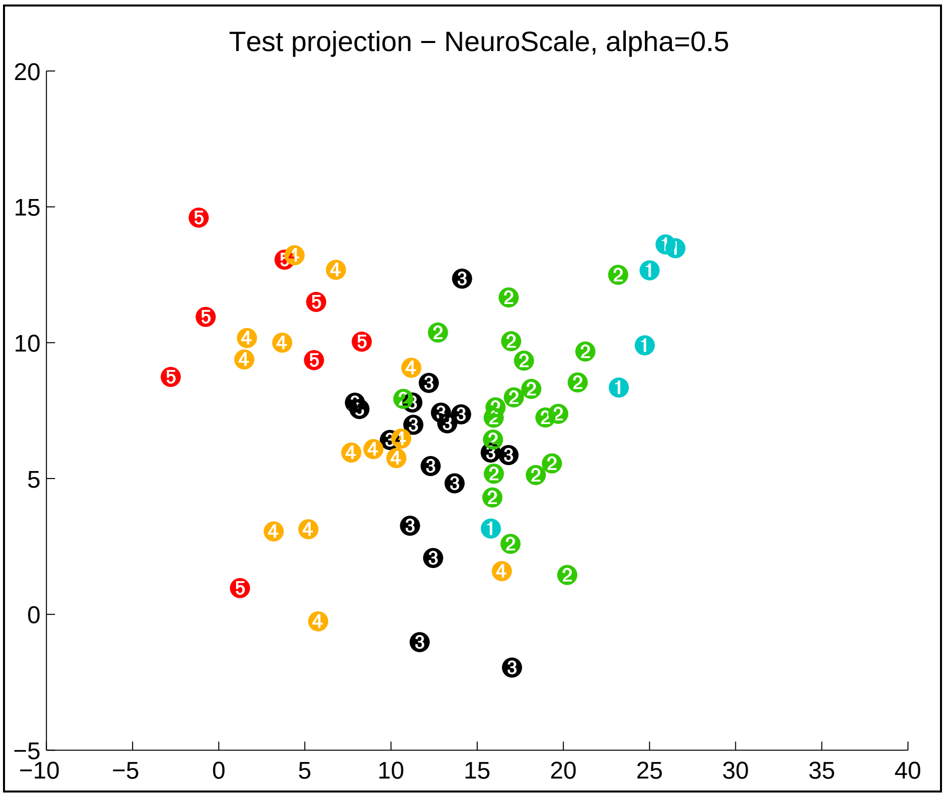


Figure 4.15: The RAE_AM test dataset transformed by the NEUROSCALE, $\alpha = 0.5$, technique previously trained on the RAE_PCB data.

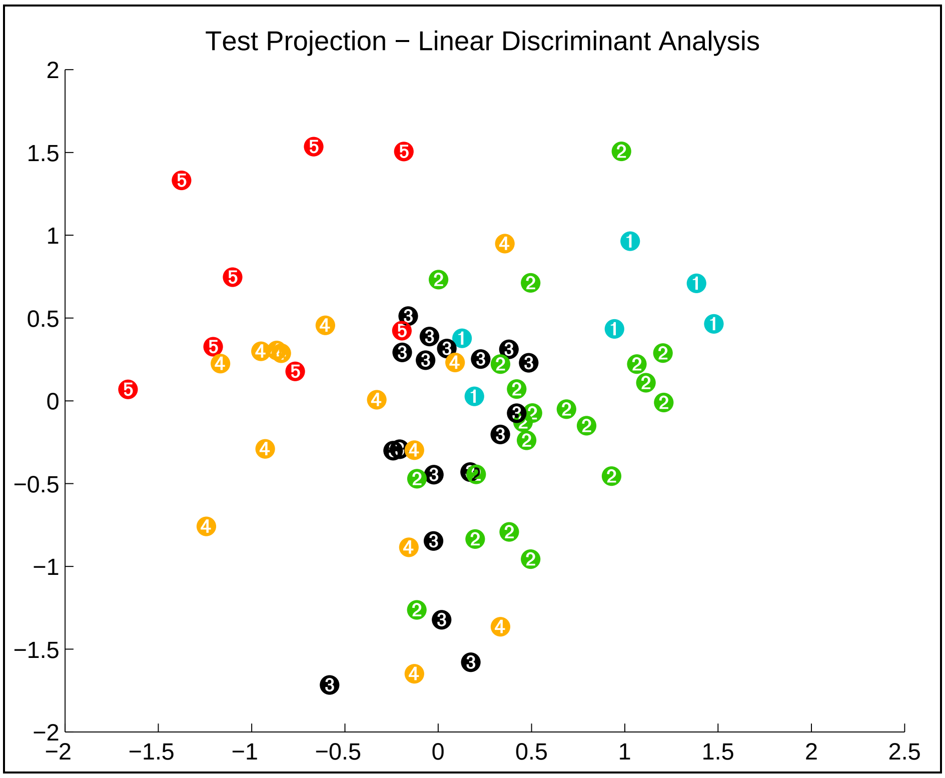


Figure 4.16: The RAE_AM test dataset projected onto the first two linear discriminant axes.

4.5 NEUROSCALE Pre-Processing for Classification

The emphasis of the previous two subsections was on the use of NEUROSCALE as a data analytic tool. However, in the discussion of Section 4.3, it was mentioned that in classification problems, for optimal performance, it is often better to extract, or select, an appropriate subset of features from the data prior to the classification stage. In fact, many texts include a schematic of a conventional prediction system similar to that shown in figure 4.17 below [Duda and Hart 1973; Fukunaga 1990; Weiss and Kulikowski 1991; Bishop 1995], which incorporates an explicit *data pre-processing* stage.

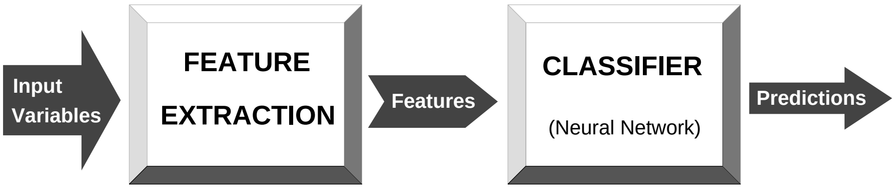


Figure 4.17: Schematic of a general classifier system.

This section considers the construction of an RBF classifier for the RAE data again, whose inputs are no longer taken from the complete set of raw variables as in Section 4.4.2, but instead from the feature spaces illustrated in the previous section as extracted by the following techniques:

① Generalised Linear discriminant analysis

② NEUROSCALE: $\alpha = 0$

③ NEUROSCALE: $\alpha = 0.5$

④ NEUROSCALE: $\alpha = 1$

4.5.1 Experimental Prediction Models

The neural network used for classification was an RBF with thin-plate spline basis functions ($r^2 \log r$), with the centres located within the data by the k-means algorithm, detailed in Section 4.4.2. The number of centres was chosen individually for each prediction model using leave-one-out cross-validation. For each pre-processing method, classification results are shown for 'typical' networks (specifically, the network that gave the median error over 10 runs of each model).

① Generalised Linear Discriminant Analysis

□ Training Set □

```
MSE per output:          0.0359165
Patterns Misclassified: 59 (out of 217)
Percentage Correct:      72.8%

Classification Matrix
=====================
 41  1  0  0  0
  4 27 11  0  0
  0  7 49  6  0
  0  1 11 28  4
  0  0  0 14 13
```

■ Test Set ■

```
MSE per output:          0.0641893
Patterns Misclassified: 32 (out of 67)
Percentage Correct:      52.2%

Classification Matrix
=====================
  4  2  0  0  0
  6 11  3  1  0
  0  6  8  4  0
  1  1  5  7  0
  0  0  0  3  5
```

② NEUROSCALE : $\alpha = 0$

□ Training Set □

```
MSE per output:          0.0592697
Patterns Misclassified: 107 (out of 217)
Percentage Correct:      50.7%

Classification Matrix
=====================
 32  8  2  0  0
 11 16 12  0  3
  4  3 42  8  5
  1  0 23 14  6
  0  0 11 10  6
```

■ Test Set ■

```
MSE per output:          0.0657081
Patterns Misclassified: 35 (out of 67)
Percentage Correct:      47.8%

Classification Matrix
=====================
  6  0  0  0  0
  5  9  6  1  0
  1  2 11  3  1
  0  0  9  4  1
  0  0  3  3  2
```

③ NEUROSCALE : $\alpha = 0.5$

□ Training Set □

```
MSE per output:         0.0293056
Patterns Misclassified: 38 (out of 217)
Percentage Correct:     82.5%

Classification Matrix
=====================
 38  4  0  0  0
  2 32  8  0  0
  0  4 52  6  0
  0  0  5 38  1
  0  0  0  8 19
```

■ Test Set ■

```
MSE per output:         0.0484208
Patterns Misclassified: 20 (out of 67)
Percentage Correct:     70.1%

Classification Matrix
=====================
  5  1  0  0  0
  1 18  2  0  0
  0  4 14  0  0
  0  1  7  6  0
  0  0  0  4  4
```

④ NEUROSCALE : $\alpha = 1.0$

□ Training Set □

```
MSE per output:         0.0323057
Patterns Misclassified: 51 (out of 217)
Percentage Correct:     76.5%

Classification Matrix
=====================
 39  3  0  0  0
  3 32  7  0  0
  0  6 49  7  0
  0  0  8 30  6
  0  0  2  9 16
```

■ Test Set ■

```
MSE per output:         0.0497819
Patterns Misclassified: 26 (out of 67)
Percentage Correct:     61.2%

Classification Matrix
=====================
  4  2  0  0  0
  0 18  3  0  0
  0  4 14  0  0
  0  1  7  3  3
  0  0  0  6  2
```

4.5.2 Discussion

The results of the various feature extraction approaches for pre-processing the data can be summarised by the histogram in figure 4.18 below, which illustrates the classification performance of the techniques ①-④ above and compares them with the direct methods of Section 4.4.2.

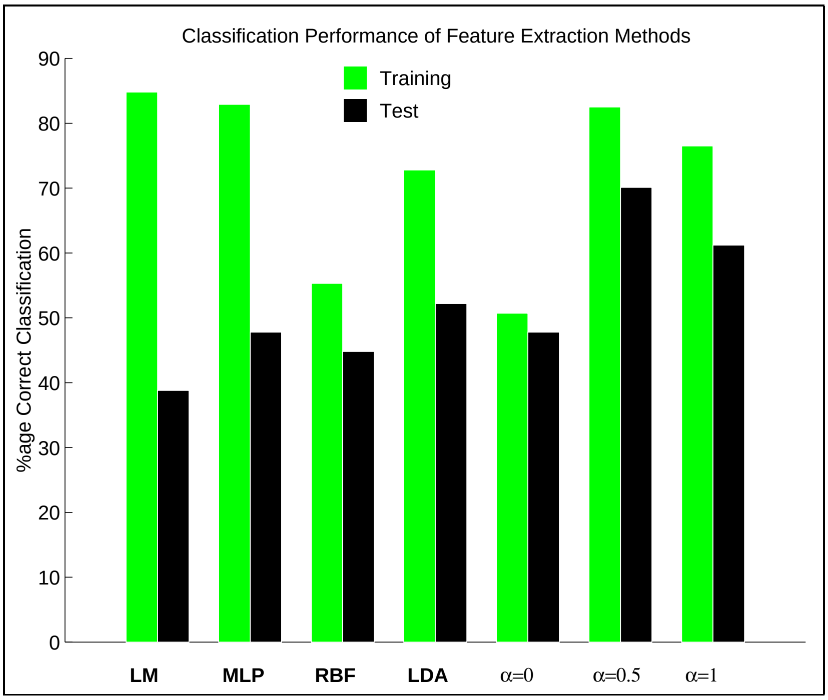


Figure 4.18: A histogram comparing the relative performance of various classifiers on the RAE data. The left-hand three models are those constructed directly from the data. The right-hand models are RBFs applied after a feature extraction stage.

The prediction rates illustrated by figure 4.18 confirm the proposition that additional input information can reduce classifier quality, particularly in the case of limited data. The classification rate on the test set for the worst predictor using pre-processed data (NEUROSCALE with $\alpha = 0$) is exactly equivalent to the best predictor applied to the entire set of input variables. Clearly, the feature extraction stage is highly appropriate for this particular task. Furthermore, effecting this pre-processing with the NEUROSCALE, $\alpha = 0.5$, network, produced the best results, classifying 70.1% of the test dataset.

That the hybrid supervised / unsupervised model ($\alpha = 0.5$) was optimal, of the three variants tested here, is particularly interesting. For the unsupervised version ($\alpha = 0$), it should be clear from studying the configuration of figure 4.8 earlier that considerable misclassification was inevitable based on those extracted features, both for the training and test sets. Note that the test error is only marginally greater than the training error for this model. Further insight into this phenomenon will be provided in Chapter 6.

The purely supervised NEUROSCALE variant ($\alpha = 1$), which might naively have been expected to generate the best results, is outperformed by a model that incorporates some element of topographic information ($\alpha = 0.5$). The solution configuration for $\alpha = 1$ is a straight line, which is closely approximated in figure 4.14, and the addition of some structural component evidently provides useful information in the second dimension. This further prompts the question of which dimensionality of feature space might be optimal in this pre-processing application, and is an obvious area for more detailed study.

Further insight into the dependence of classifier performance on the value of α is given in figure 4.19 below. This illustrates the classification rate of a typical classifier for values of α from 0 to 1 in steps of 0.1. Note that although these values are for a single run (averaging over many such runs of both the scaling and classification procedure would be highly computationally expensive), there is still an evident trend which implies that there is some mid-range value of α which leads to optimal performance.

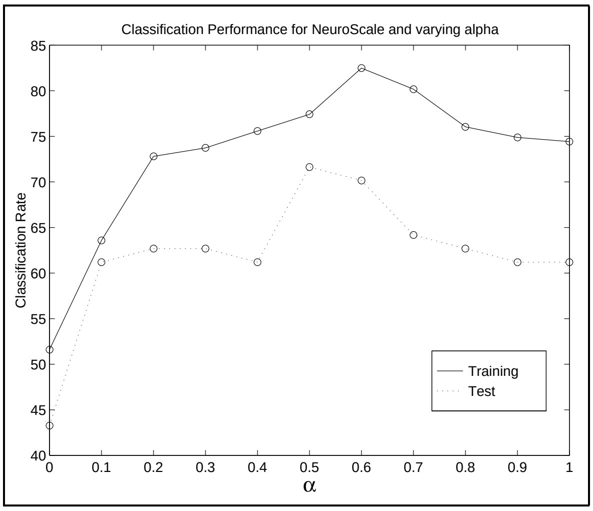


Figure 4.19: Variation of classification performance with α.

It is possible to generate a two-dimensional $\alpha = 1$ supervised feature space by altering the subjective dissimilarities. For example the matrix of inter-class (research-rating) dissimilarities

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

will produce such a space. The matrix $\mathbf{C}'$ simply attempts to cluster points from identical classes and separate those from different classes by the unit distance. There is therefore no implied *ordering*, as embodied in the matrix **C** used for earlier NEUROSCALE projections, and the resultant $\alpha = 1$ configuration is of a circular nature. However, in this experiment, there was no significant improvement in classification error observed when using such a matrix.

This implies that retention of some level of structure in the feature extraction phase is of benefit to later classification, and may in fact be more useful than a particular explicit discrimination criterion. This confirms a major advantage of a topographical transformation such as NEUROSCALE, since conventional approaches to structure-preservation, such as the Sammon mapping, do not possess a generalisation property, which precludes their inclusion in a predictive model.

The author is only aware of two examples in the literature where experiments have been conducted that exploit subjective class information within a parameterised topographic map before the discrimination process. Koontz and Fukunaga [1972] considered simple two-dimensional two-class problems, mapped also to two dimensions using their 'distance-difference' mappings. It was observed that classification performance improved as the discriminating element (J_{SE}) of the mapping was increased, for both separable and non-separable problems. Cox and Ferry [1993] also employed similar and "somewhat contrived" datasets to illustrate the efficacy of their MDS-based mapping. Additionally, they applied the method to a real 15-dimensional two-class dataset and showed that applying the MDS mapping prior to discrimination with Fisher's function improved classification accuracy compared to direct application of that function to the original data. This, however, should be expected as the pre-processing implies that additional nonlinear behaviour may be exploited in the discrimination process, whilst Fisher's discriminant is of linear character.

4.6 Conclusions

4.6.1 Exploratory Data Analysis

Analysing data taken from the 1992 Research Assessment Exercise is a particularly interesting, and difficult, problem. The data is of very high dimension, and this, coupled with the "noise" implied by the subjective peer assessment component of the exercise, makes the research ratings difficult to predict from the quantitative data alone. This was underlined in section 4.4.2, where, with the models investigated, it was found to be impossible to predict more than half of an unseen test set correctly.

With respect to visualising the data in two dimensions, several illustrative feature spaces, of linear–nonlinear and supervised–unsupervised character, produced by standard methods were generated. Of the linear techniques, the principal component projection failed to elucidate any useful structure from the data, while by exploiting the associated class information, the linear discriminant space separated the research ratings into approximate clusters. This latter technique could then project previously unseen data, although the clustering was less pronounced.

Of the nonlinear techniques, the Sammon Mapping is an improvement upon the unsupervised PCA projection, exhibiting more structure, but again offered little promise for exploratory analysis. Equally so is the feature space produced by the NEUROSCALE technique with $\alpha = 0$, although this latter approach would permit subsequent projection of the test data set. The established neural network approach to topographic mapping, the Kohonen map, proved to be ill-matched to the topology of the data and thus of limited utility in this application.

The most effective visualisation of the data presented in this chapter was that obtained from NEUROSCALE with an intermediate value of $\alpha = 0.5$, illustrated in section 4.4.6. The merit of combining both unsupervised and supervised — or structural and subjective — flavour in the feature space is made explicit. The subjective knowledge imposes a clustering and ordering constraint that facilitates analysis with respect to classification of departments. The generalisation capability of this approach was also clearly shown, illustrating a better projection than that achieved by LDA, and which showed potential (see below) for subsequent decision making or inference. Previously unseen data, which is transformed by the network, may be analysed in that space. The supervisorial content of the projection allows a prospective prediction to be made, and the structural content implies that the confidence of that prediction can be gauged according to the proximity of the test point to other clusters.

Equally, by retaining some of the structure of the original data, it is also possible to highlight apparent anomalies and inconsistencies in the assessment process, and this evidence was supported by the results of the previous classification experiment.

4.6.2 Data Pre-processing for Classification

The results from Section 4.5 confirm that improved classifier performance can be attained by extracting an appropriate set of features from the data *a priori*. Given this approach, it was possible to correctly predict the rating awarded to 70% of the Applied Mathematics departments (and 82% of those awarded ratings 1, 2 and 3). Note that these ratings were determined by a distinct assessment panel from the Physics, Chemistry and Biological Science departments (the data upon which the classifier was trained), and that it had access to additional qualitative information in making its decision. Despite this, and the large number of available input variables, it is evident that considerable structure *is* present in the data, in terms of the relationship to the research rating ultimately awarded. This conclusion supports the argument that, from the perspective of the funding councils, collection of the research data is, in part at least, a worthwhile exercise. However, as considered by Taylor [1995], this chapter has not addressed the question of *which* of the large number of variables are relevant.

It would be interesting to attempt, using the best model above, to predict the outcome of the 1996 exercise for those relevant departments. Unfortunately, changes in the assessment process and the supplied data would probably render this exercise somewhat difficult and ineffective.

Of particular interest from the pre-processing experiments, is the fact that the optimum feature extraction method, of those investigated, was the NEUROSCALE transformation with $\alpha = 0.5$. This approach outperformed its purely unsupervised and purely supervised relatives. Clearly, the structural influence upon the projection is beneficial to classification, and, counter-intuitively, more effective than supervisorial criteria alone, whether linear (LDA) or nonlinear (MLP and NEUROSCALE with $\alpha = 0$). The questions of exactly which value of α and output dimension q are optimal in this experiment remain open.

Two similar investigations into such topographic pre-processing in the literature are too simplistic for serious comparison, and the results for NEUROSCALE on the RAE data would appear significant. However, the primary focus of this thesis is concerned with topographic mappings for generic feature extraction, rather than tailored extraction for the purposes of any specific prediction or classification task. There thus remains much scope for further investigation of this aspect of NEUROSCALE.

Chapter 5

Classical Multidimensional Scaling and the Principal Subspace Network

5.1 Introduction

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

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

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

5.2 Classical Multidimensional Scaling Revisited

5.2.1 The CMDS Procedure

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

The procedure is summarised as follows:

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

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

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

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

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

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

5.2.2 Optimality Properties

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

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

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

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

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

and in corresponding form,

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

Thus E can be given by:

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

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

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

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

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

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

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

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

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

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

is minimised.[1]

Proof. Equation (5.2) may be minimised with respect to **Y** by differentiating to find the stationary points:

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

So at the stationary points,

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

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

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

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

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

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

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

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

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

5.3 A Neural Network CMDS Transformation

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

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

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

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

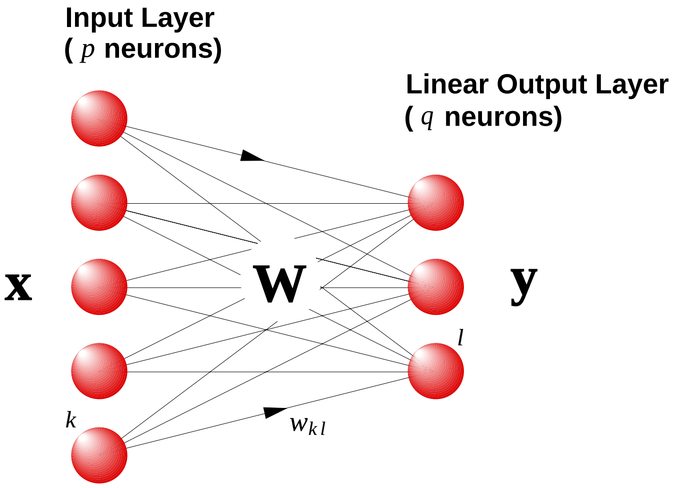


Figure 5.1: A simple linear neural network to perform CMDS.

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

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

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

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

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

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

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

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

where η is a small constant.

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

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

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

5.4 The Principal Subspace Network

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

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

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

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

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

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

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

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

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

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

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

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

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

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

with the constant term again introduced for convenience.

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

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

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

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

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

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

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

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

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

Equating the real parts gives:

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

Similarly,

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

and equating imaginary parts gives:

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

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

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

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

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

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

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

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

5.5 The Relationship between the PSN and CMDS Learning Rules

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

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

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

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

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

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

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

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

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

5.6 Conclusions

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

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

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

Chapter 6

The Form of Topographic Transformations

6.1 Introduction

In many conventional applications which exploit neural networks — function approximation, classification and time-series prediction for example — there are several important design and implementation issues that must be considered, and these are equally relevant to neural networks which are to be trained to effect topographic mappings.

In terms of network design, a fundamental decision is that of *model order selection*, which is related to the trade-off between bias and variance as remarked upon in Section 4.4.2. It is necessary to determine the sufficient network *complexity* that will permit good generalisation — that is, it is naturally desired that projections of new data, drawn from the same distribution as that used to train the network, will also have low STRESS.

With respect to implementation, a key problem is one of network weight optimisation. There are numerous optimisation schemes available, and consideration of these will be deferred to the next chapter. Furthermore, from the evidence of previous research with Sammon Mappings, sub-optimal local minima may be expected to be highly problematic.

This chapter will examine the NEUROSCALE approach in terms of the form of the network transformation, with the emphasis on purely topographic, or objective, mappings (those with $\alpha = 0$). Firstly, the problem of local minima in network optimisation will be considered. While this may appear to be more relevant to Chapter 7, it in fact proves to be a fruitful starting point for this chapter as the results of that investigation lead naturally on to a discussion concerning the *smoothness* of the network transformation, which is in turn closely related to the questions of model complexity and generalisation.

6.2 Local Minima

6.2.1 The Sammon Mapping

The nonlinear optimisation procedures conventionally employed in the generation of Sammon mappings are *local* schemes (see Klein and Dubes [1989] and Hofmann and Buhmann [1995] for exceptions). When applied to the optimisation of the Sammon STRESS, the final configuration will be a local minimum of the STRESS function. In such applications, it has been a general observation that many of these minima, and respective configurations, are considerably sub-optimal. An empirical study of local minima in two-dimensional scaling was undertaken by Mathar and Zilinskas [1993]. When applied to the (10×10) cola-testing data [Schiffman, Reynolds, and Young 1981], for example, the region of attraction of the global solution was estimated at only 4.8%. Other starting configurations outwith this region gave rise to sub-optimal minima with correspondingly higher values of STRESS.

A similar experiment may be undertaken for 45 patterns[1] from the Iris data set [Fisher 1936], and the results are given in figure 6.1. (This subset of the Iris data will be referred to as the IRIS_45 set.) This presents a histogram of the number of runs of the Sammon mapping algorithm, out of a total of 1000, that gave corresponding minimum STRESS values. The configurations were initialised at random, and a conjugate-gradient minimisation routine employed.

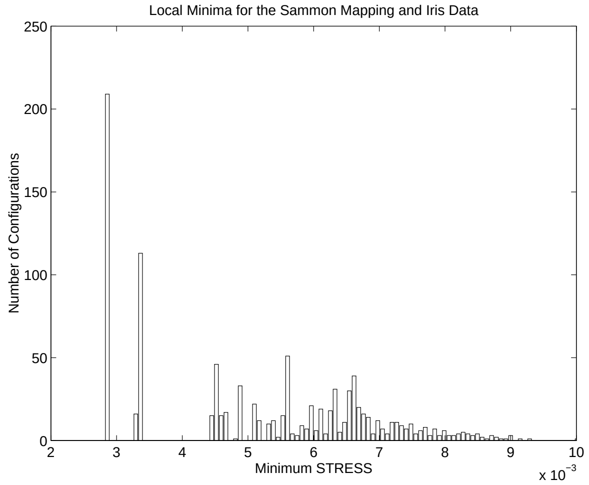


Figure 6.1: Histogram of number of final configurations with corresponding STRESS's for 1000 runs of the Sammon Mapping on 45 patterns from the Iris data set (IRIS_45).

It is clear that while the mode is at the global[2] minimum of 0.0028, there are many sub-optimal local minima. The proportion of runs which approached the global minimum was 21.3%, with the mean and standard deviation of the final STRESS being 0.0051 and 0.0017 respectively. It should be noted that the 'global' minimum found from random initialisation of the configuration was identical to that minimum obtained via PCA initialisation.

6.2.2 Parameterised Transformations

An equivalent experiment to figure 6.1 may also be conducted for NEUROSCALE (with $\alpha = 0$ to emulate the equivalent Sammon mapping). The results, for a radial basis function network with 10 Gaussian basis functions, are given in figure 6.2. The global width parameter, σ, of the basis functions was

[1] A representative subset of the full data was chosen in order to permit large numbers ($\approx$ 1000) of runs of the scaling algorithms.

[2] It is assumed, after many experiments, that this is indeed the *global* minimum, although this cannot be known for certain.

set to 3.0, where the k-th basis function is given by $\phi_k(\mathbf{x}_i) = \exp(- \parallel \mathbf{x}_i - \boldsymbol{\mu}_k \parallel^2 / \sigma^2)$, $\boldsymbol{\mu}_k$ being the respective centre.

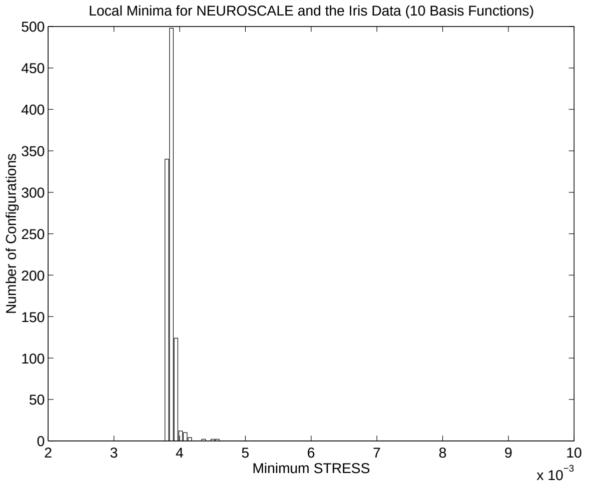


Figure 6.2: Histogram of number of final configurations with corresponding STRESS's for 1000 runs of NEUROSCALE, with $\alpha = 0$, on the IRIS_45 dataset. The width of the 10 Gaussian basis functions was 3.0.

The contrast between the standard Sammon Mapping and the NEUROSCALE RBF approach is striking. In the case of NEUROSCALE, the minimum STRESS is 0.0038. This is notably higher than that for the Sammon Mapping, which might be expected due to the reduced flexibility of the RBF model implied by the use of only 10 basis functions. However, the maximum STRESS is only 0.0046, with the mean and standard deviation 0.0039 and 0.000074 respectively. The underlying result is that there does not appear to be a significant problem with sub-optimal local minima in the case of the NEUROSCALE approach on this dataset. This supports the observation of Webb [1995], when performing MDS by iterative majorisation with radial basis functions for the entire 150-pattern Iris data set, that “the value of the loss [STRESS] to which the procedure converged was relatively insensitive to the initial weight configuration.”

It is natural to also consider a more flexible RBF, and figure 6.3 gives the minima obtained when optimising a model with 45 Gaussian basis functions, also of width 3.0.

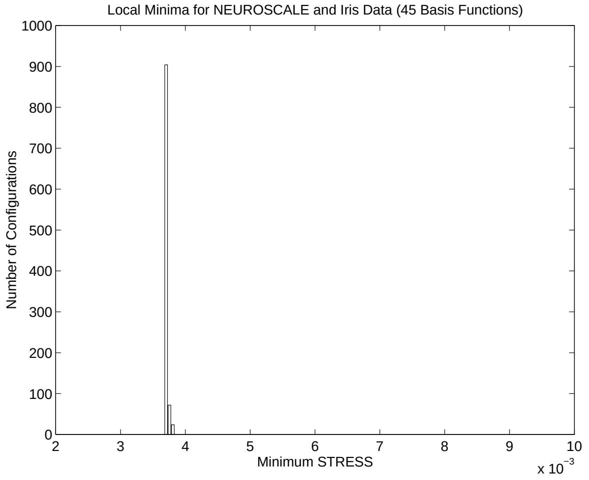


Figure 6.3: Histogram of number of final configurations with corresponding STRESS's for 1000 runs of NEUROSCALE, with $\alpha = 0$, on the IRIS_45 data set. The width of the 45 Gaussian basis functions was 3.0.

This gives a very similar distribution of minima to the 10 basis function plot, and with a similar minimum of 0.0037. In this case, however, the maximum value is only 0.0038 and the standard deviation 2×10^{-5}. Despite the fact that there are as many basis functions in the RBF as data points, the network has still not attained the potential minimum STRESS value of 0.0028. Also, by comparison with figure 6.1, it can be seen that the apparent minimum (and that exhibited in figure 6.2 for 10 basis functions) does not correspond to one of the sub-optimal configurations generated by Sammon's algorithm. From this experiment, it is also retrospectively apparent that the relatively high minimum value of STRESS observed in figure 6.2 was not a result of employing only 10 basis functions in that network.

Further understanding of this phenomenon can be obtained by realising that there are *two* levels of minima in the neural network implementation. The first level is the set of minima as generated by Sammon's method. The second level is the set of extra minima introduced by the mechanism of the RBF model. To clarify this distinction, consider the equations for calculating the derivatives of STRESS with respect to the weights in the NEUROSCALE model. The relevant equations are reproduced here from Section 3.2:

$$\frac{\partial E}{\partial w_k} = \sum_i^N \frac{\partial E}{\partial \mathbf{y}_i} \cdot \frac{\partial \mathbf{y}_i}{\partial w_k}, \quad \text{where} \tag{6.1}$$

$$\frac{\partial E}{\partial \mathbf{y}_i} = -2 \sum_{j \neq i} \left(\frac{d^*_{ij} - d_{ij}}{d_{ij}} \right) (\mathbf{y}_i - \mathbf{y}_j). \tag{6.2}$$

The derivatives $\partial E / \partial \mathbf{y}_i$ are exactly those determined in Sammon's algorithm, and when these are all zero, clearly so are all the $\partial E / \partial w_k$. This represents the first level of minima.

However, even if these derivative terms are non-zero, all the $\partial E / \partial w_k$ may still be zero, given appropriate values of $\partial \mathbf{y}_i / \partial w_k$. To determine which such values give rise to minima, firstly the weights $\mathbf{w}_l = (w_1, w_2, \ldots, w_h)^{\mathrm{T}}$ for each output dimension, l, of the network are considered in isolation, as each such weight vector only affects a single network output (and the subscript 'l' will be dropped where it is unambiguous to do so). Then, equation (6.1) may be expressed in matrix form as:

$$\nabla E = \mathbf{J}^{\mathrm{T}} \boldsymbol{\delta}, \tag{6.3}$$

where

$$\nabla E = (\frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2}, \ldots, \frac{\partial E}{\partial w_h})^{\mathrm{T}},$$

and $\mathbf{J}$ is the *Jacobian* matrix:

$$\mathbf{J} = \begin{bmatrix} \frac{\partial y_{1l}}{\partial w_1} & \frac{\partial y_{1l}}{\partial w_2} & \cdots & \frac{\partial y_{1l}}{\partial w_h} \\ \frac{\partial y_{2l}}{\partial w_1} & \frac{\partial y_{2l}}{\partial w_2} & \cdots & \frac{\partial y_{2l}}{\partial w_h} \\ \ldots & \ldots & \ldots & \ldots \\ \frac{\partial y_{Nl}}{\partial w_1} & \frac{\partial y_{Nl}}{\partial w_2} & \cdots & \frac{\partial y_{Nl}}{\partial w_h} \end{bmatrix},$$

where y_{il} is the value of output dimension l for pattern i, and

$$\boldsymbol{\delta} = (\frac{\partial E}{\partial y_{1l}}, \frac{\partial E}{\partial y_{2l}}, \ldots, \frac{\partial E}{\partial y_{Nl}})^{\mathrm{T}}.$$

Thus local minima occur, for non-zero $\boldsymbol{\delta}$, when $\boldsymbol{\delta}$ is orthogonal to the column-space of (or lies in the *null-space* of) $\mathbf{J}$. For an RBF with h basis functions, the Jacobian is given by

$$\mathbf{J} = \begin{bmatrix} \phi_1(\mathbf{x}_1) & \phi_2(\mathbf{x}_1) & \ldots & \phi_h(\mathbf{x}_1) \\ \phi_1(\mathbf{x}_2) & \phi_2(\mathbf{x}_2) & \ldots & \phi_h(\mathbf{x}_2) \\ \ldots & \ldots & \ldots & \ldots \\ \phi_1(\mathbf{x}_N) & \phi_2(\mathbf{x}_N) & \ldots & \phi_h(\mathbf{x}_N) \end{bmatrix}, \tag{6.4}$$

and remains identical whichever output l of the network is considered.

With 45 basis functions, this matrix should be full rank, and no such second level minima should be present. However, for larger values of width parameter σ, it is likely to be ill-conditioned. This is the case for the example above, where the condition number (ratio of largest to smallest singular values) is 9.45×10^8. For an ill-conditioned Jacobian, if not singular and thus a true local minimum does not exist, numerical round-off error may effectively generate an artefactual one. Even if not, should $\boldsymbol{\delta}$ ever approach the subspace spanned by the eigenvectors corresponding to the very small eigenvalues, the gradient will become extremely small and training will, for all intents and purposes, terminate.

To view this more explicitly, consider a single output dimension l of the network once more, for a simple gradient-descent error minimisation scheme. The vector of output neuron l for all the patterns will be denoted $\mathbf{z}_l = (y_{1l}, y_{2l}, \ldots, y_{Nl})^{\mathrm{T}}$, and therefore represents the l^{th} column of the output data matrix $\mathbf{Y}$. The weight update equation at time step t is then $\mathbf{w}^{t+1} = \mathbf{w}^t - \eta\nabla E$. This implies that

$$
\begin{aligned}
\mathbf{z}_l^{t+1} &= \mathbf{J}\mathbf{w}^{t+1}, && (6.5)\\
&= \mathbf{J}(\mathbf{w}^t - \eta\nabla E), && (6.6)\\
&= \mathbf{z}^t - \eta\mathbf{J}\mathbf{J}^{\mathrm{T}}\boldsymbol{\delta}^t. && (6.7)
\end{aligned}
$$

If there is a basis function located at each input point, then $\mathbf{J}$ is symmetric and may be decomposed as $\mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^{\mathrm{T}}$, where $\mathbf{U}$ is the matrix of eigenvectors of $\mathbf{J}$ and $\boldsymbol{\Lambda}$ the corresponding diagonal matrix of eigenvalues. From this:

$$
\mathbf{J}\mathbf{J}^{\mathrm{T}} = \mathbf{U}\boldsymbol{\Lambda}^2\mathbf{U}^{\mathrm{T}}. \qquad (6.8)
$$

If $\boldsymbol{\delta}$ is then expressed as a weighted sum of those eigenvectors, such that $\boldsymbol{\delta} = \sum_k \beta_k \mathbf{u}_k$, then the corresponding change in $\mathbf{z}$ is

$$
\begin{aligned}
\Delta\mathbf{z} &= \mathbf{z}^{t+1} - \mathbf{z}^t, && (6.9)\\
&= -\eta\sum_k \lambda_k^2\beta_k\mathbf{u}_k. && (6.10)
\end{aligned}
$$

Thus, $\Delta\mathbf{z}$ will be diverted from the direction $\boldsymbol{\delta}$, such that its components in the directions of the principal eigenvectors of $\mathbf{J}$ are accentuated in comparison to those of the minor eigenvectors. The magnitude of this effect is related to the condition number of the matrix $\mathbf{J}$, and may be illustrated by figure 6.4 below. This figure provides a simple example of a two-dimensional quadratic error surface, where the minimum lies at the origin. The evolution of the vectors $\mathbf{z}^t$ are shown for an equivalent NEUROSCALE gradient-descent scheme with (2×2) $\mathbf{J}$, whose eigenvectors and squared eigenvalues are also given in the figure. For comparison, the plot for the Sammon mapping is also illustrated, which is equivalent to $\mathbf{J}\mathbf{J}^{\mathrm{T}} = \mathbf{I}$.

In figure 6.4, the trajectory for the equivalent Sammon mapping converges on the minimum directly as expected. By contrast, the effect of the λ_1^2 term in equation (6.10) above is to initially cause the gradient-descent trajectory to approach the origin more rapidly than the Sammon trace. However, $\boldsymbol{\delta}$ soon becomes near-perpendicular to $\mathbf{u}_1$, and the trajectory is dominated by the effect of the smallest eigenvalue, $\lambda_2^2 = 0.0225$, causing convergence on the origin in the direction $\mathbf{u}_2$ to be dramatically retarded.

Although the global surface of the STRESS cost function will be very different from the quadratic example given above, in the vicinity of a minimum such an approximation may be quite reasonable. (This assumption is indeed made implicitly by BFGS and other optimisation algorithms.)

By considering each output in isolation, if $\mathbf{J}$ is not full rank (for example, when there are fewer basis functions than input data points), then the implication is that there will be a genuine local minimum

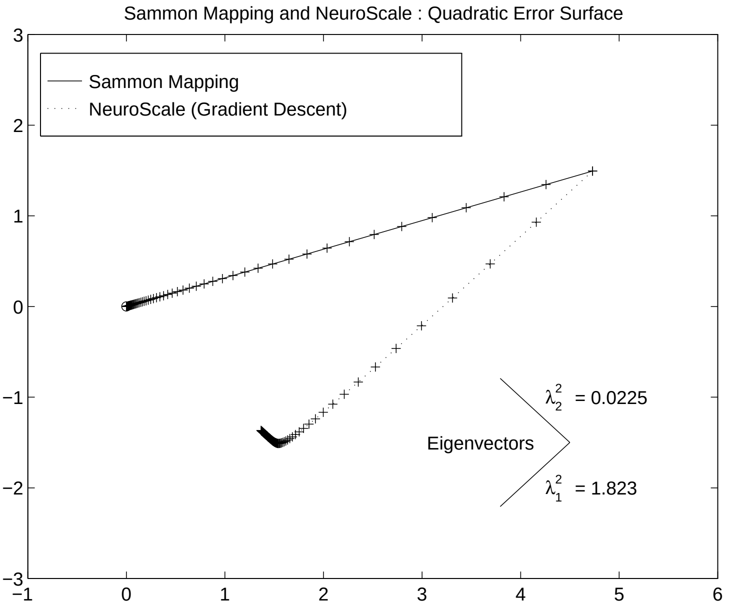


Figure 6.4: Comparison of gradient-descent and Sammon Mapping algorithms on a quadratic error surface.

due to the null-space of **J** where the corresponding eigenvalues are zero. However, for multiple outputs, the situation is more complex than this. If the error for a given output dimension l is such that $\mathbf{J}^{\mathrm{T}}\boldsymbol{\delta}_l = \mathbf{0}$, then in general, $\mathbf{J}^{\mathrm{T}}\boldsymbol{\delta}_{l'} \neq \mathbf{0}$, for $l' \neq l$. The output dimensions, l', for which the weight changes are non-zero will cause the the vector $\mathbf{z}_{l'}$, and thus the overall configuration **Y**, to change, which will in turn result in modification of the inter-point distances d_{ij}. These distances are related to the $\boldsymbol{\delta}$ for *all* output dimensions through equation (6.2), which implies that $\boldsymbol{\delta}_l$ will be perturbed from the null-space and there is thus an 'escape route' from the minimum. The capability of the algorithm to avoid the minimum in such a manner is a side-effect of the existence of a set of multiple solutions which are equivalent, with respect to STRESS, and may be generated by a arbitrary rotation and / or translation of any one single solution.

The effects of the eigenvalues of **J** upon the convergence of the relative supervision algorithm suggests that better convergence behaviour may be obtained through the selection of a relatively small value of basis function width, σ, and thus ensuring that **J** is well-conditioned. An identical plot to that of figure 6.3, but with $\sigma = 0.01$, is therefore given in figure 6.5.

The distribution of final minimum STRESS values is now very similar to the Sammon mapping given in figure 6.1 earlier. The minimum value is again 0.0028 and 19.8% of runs approach this value. The mean and standard deviation were 0.0052 and 0.0018 respectively. All the local minima observed also now correspond to those observed from Sammon's procedure, with the implication that there are no longer any second level minima effects. Indeed, the condition number of the Jacobian is now 1.0. With $\sigma = 0.1$ it is 53.4, and with $\sigma = 1$, it becomes 4.4×10^7.

The two previous figures illustrate that there is an evident trade-off between levels of local minima which may be controlled by the choice of basis function width. Decreasing σ improves the conditioning of the Jacobian and can moderate the effective attenuation of learning rate that results from the squared-eigenvalue terms in the update equation. However, this in turn permits the procedure to be caught in level-1 local minima, because for a full complement of basis functions, as $\sigma \rightarrow 0$, $\mathbf{J} \rightarrow \mathbf{I}$, the identity matrix, and gradient-descent NEUROSCALE becomes an exact equivalent of Sammon's algorithm.

It is therefore clear why *decreasing* σ should improve performance of the NEUROSCALE training algorithm with respect to the level-2 minima effects. In the next section, reasons why *increasing* σ improves

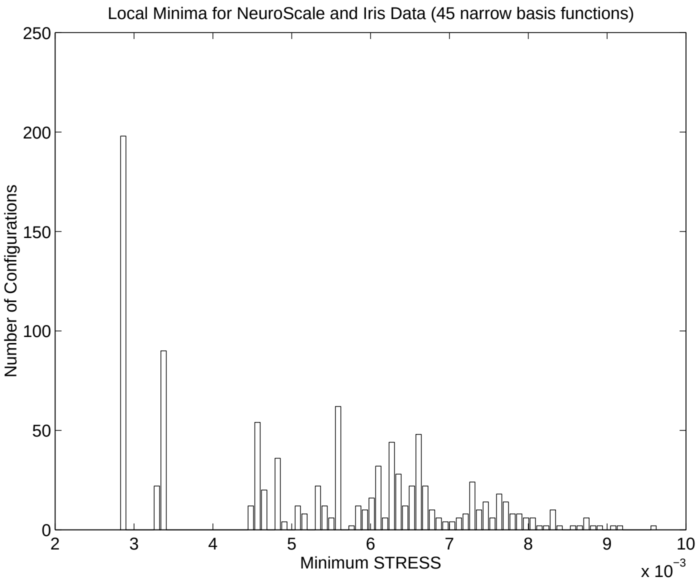


Figure 6.5: Histogram of number of final configurations with corresponding STRESS's for 1000 runs of NEUROSCALE, with $\alpha = 0$, on the IRIS_45 data set. The width of the 45 Gaussian basis functions was 0.01.

performance with respect to level-1 minima are considered.

6.3 Smoothness of Topographic Transformations

The experiments in the previous section indicated that, for moderate values of basis function width, the NEUROSCALE technique is less plagued by sub-optimal level-1 local minima — those inherent in the Sammon Mapping alone — than is the standard Sammon procedure itself. One potential explanation for this observed behaviour is that the vast majority of such minima found by Sammon's algorithm are highly *unsmooth* transformations of the input data. The basins of attraction for these poor minima in configuration space are likely to be very distant from the configurations attained by random initialisation of the RBF weights with significant values of basis function width.[3]

Further evidence for this hypothesis may be obtained by the following experiment. The standard Sammon Mapping algorithm was run, from random initialisations, 1000 times to generate an equal number of configurations of points with an associated minimised STRESS value. An RBF, with 15 Gaussian basis functions, of width 1.0, whose fixed centres were chosen at random from the data points, was then trained, in the conventional supervised (pseudo-inverse) manner, to generate each of the previous Sammon configurations. If high-STRESS configurations are not easily realisable by the RBF network, then they should give rise to significant residual sum-of-squares error. Figure 6.6 gives this error, plotted against the STRESS minimum, for these 1000 runs. The residual error was averaged over 25 runs of the RBF training process, as the centre selection was different on each run.

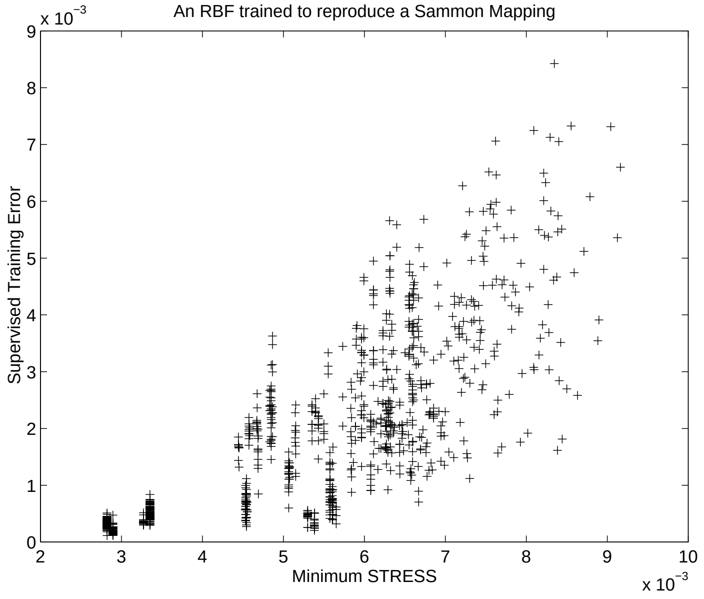


Figure 6.6: Residual error against minimum STRESS for an RBF trained to produce the final configuration of Sammon's procedure *a posteriori*. The RBF comprised 15 basis functions of width parameter 1.0.

The underlying trend in figure 6.6 supports this proposition, as, on average, higher value of minimised STRESS give higher values of residual error for an *a posteriori*[4] RBF transformation.

A more direct measure of smoothness can be obtained by calculating a measure of the *curvature* of the transformation effected by the RBF. That is, some measure of how much the gradient of **y** changes

[3] Gaussian basis functions have a width parameter, but, of course, there are other such functions utilised in radial basis function networks — cubics, thin-plate splines etc — that are not parameterised. In such cases there is an implicit global width parameter determined by the scaling of the data.

[4] This terminology will be henceforth adopted for any RBF network that effects a topographic transformation having been trained to reproduce a final configuration of a Sammon mapping, as distinct from being trained via a relative supervision procedure.

with **x**. One such measure is that given by the functional

$$E_C = \sum_i^N \sum_l^q \sum_m^p \left(\frac{\partial^2 y_{il}}{\partial x_m^2} \right)^2 , \tag{6.11}$$

where *i* ranges over the patterns, *m* over the input dimensions and *l* over the output dimensions. Thus y_{il} is the scalar value of the *l*-th output node for pattern *i*, and the overall expression is a measure of the total magnitude of the second derivatives of the output with respect to the input, evaluated at all the input points. Such a functional was used by Bishop [1991, 1993] for the regularisation of neural networks in supervised learning problems.

This measure is evaluated in the graph of figure 6.7, which is otherwise identical to 6.6 with the exception that the vertical axis indicates the value of E_C for each *a posteriori*-trained RBF, rather than the residual error.

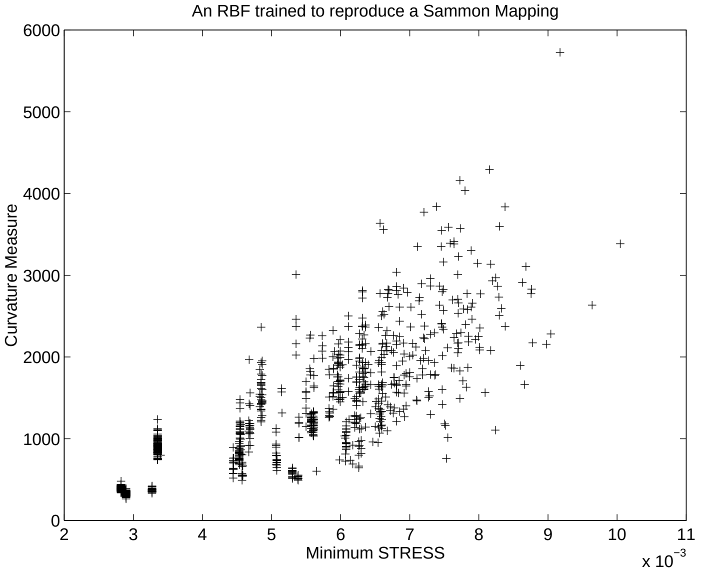


Figure 6.7: Curvature against minimum STRESS for an RBF trained to produce the final configuration of Sammon's procedure *a posteriori*. The RBF comprised 15 basis functions of width parameter 1.0.

Both these plots indicate that higher local minima represent more unsmooth transformations from the data space to the configuration space. While the spread of curvature values increases with increasing STRESS, it is nevertheless clear that low STRESS generally implies lower curvature on average, and with reduced variance in addition.

Further evidence that during the training of NEUROSCALE, curvature generally decreases with STRESS, is presented in figure 6.8. This graph plots the curvature measure E_C at each step of the training algorithm for three different RBF models with 15, 30 and 45 basis functions respectively. After an initial phase where the curvature increases over short sections of the training sequence, the measure decreases and stabilises.

Additional confirmation of this relationship is given for a single NEUROSCALE mapping of the RAE PCB dataset from Chapter 4 in figure 6.9. In this experiment, a network with 40 Gaussian basis functions was utilised.

The results of figure 6.8 are particularly interesting, as it is evident that the final curvature is not related to the number of basis functions in the transformation. This is contrary to naive expectation, as might arise from experience with conventional supervised training problems, which would assert that the more flexible model implied by a greater number of basis functions would exhibit more curvature. To

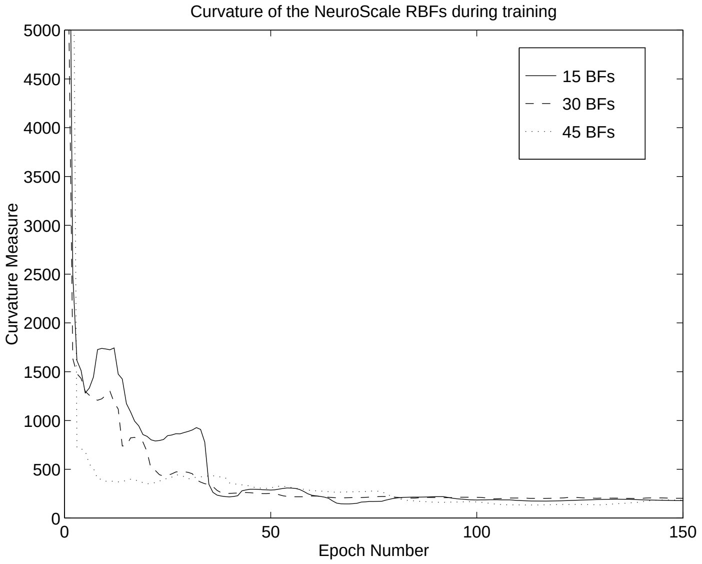


Figure 6.8: Curvature against time during the training of a NEUROSCALE mapping on the IRIS_45 data, for networks with 15, 30 and 45 basis functions.

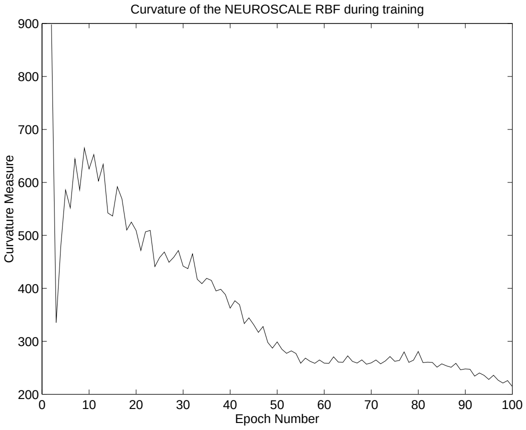


Figure 6.9: Curvature against time during the training of a NEUROSCALE mapping on the RAE_PCB dataset.

help understand this apparent anomaly, the relationship between STRESS and curvature is considered more formally in the next section.

6.4 The Relationship between STRESS and Curvature

The plots in figures 6.8 and 6.9 illustrate experimentally that the curvature of the RBF transformation from input space to configuration space generally decreases during the minimisation of STRESS.

To understand how the two quantities may be related, consider the diagram in figure 6.10, which illustrates a very simple example for a transformation from one-dimension to one-dimension. In practice, this is unrealistically trivial, but enables an underlying interaction between STRESS and curvature to be expressed mathematically.

The figure illustrates three input points $\{x_i, x_j, x_k\}$ mapped to three respective output points $\{y_i, y_j, y_k\}$. If any two points, for example the two extreme points $\{x_i, y_i\}$ and $\{x_k, y_k\}$, are correctly positioned, such that $y_k - y_i = x_k - x_i$, then it is intuitive that any error in the placement of the third point, for example $\{x_j, y_j\}$, must imply non-zero curvature at some point in the interval (x_i, x_k), whatever the form of the interpolating transformation.

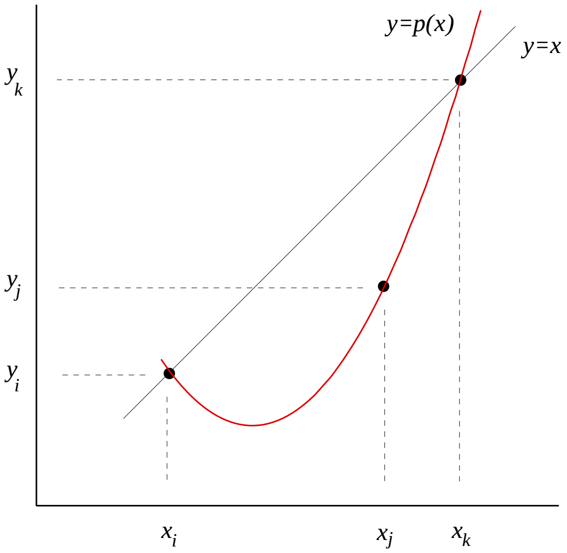


Figure 6.10: A simple example mapping of three points from one-dimension to one-dimension.

This relationship between STRESS and curvature may be demonstrated more formally. Let the mapping from x to y be effected by a RBF transformation, such that $y_i = f(x_i)$ etc. The three input points can be considered to be ordered, such that $x_k > x_j > x_i$, and are sufficiently local that the RBF function $f(x)$ can be approximated by a quadratic polynomial function $p(x) = ax^2 + bx + c$.

The Lagrange polynomial [Plybon 1992] that interpolates the three points is:

$$p(x) = y_i\frac{(x - x_j)(x - x_k)}{(x_i - x_j)(x_i - x_k)} + y_j\frac{(x - x_i)(x - x_k)}{(x_j - x_i)(x_j - x_k)} + y_k\frac{(x - x_i)(x - x_j)}{(x_k - x_i)(x_k - x_j)} \tag{6.12}$$

The curvature of the RBF transformation, as given by the measure of equation (6.11) is $[f''(x)]^2$, which for the quadratic approximation is identical evaluated at each data point and is simply $4a^2$, where a is the coefficient of the x^2 term:

$$a = \frac{y_i}{(x_i - x_j)(x_i - x_k)} + \frac{y_j}{(x_j - x_i)(x_j - x_k)} + \frac{y_k}{(x_k - x_i)(x_k - x_j)}. \tag{6.13}$$

Now, the inter-point distance relationships can be summarised by:

$$d^*_{ij} = x_j - x_i, \tag{6.14}$$

$$d^*_{jk} = x_k - x_j, \tag{6.15}$$

$$d^*_{ik} = x_k - x_i, \tag{6.16}$$

in the input space, and

$$d_{ij} = |y_j - y_i|, \tag{6.17}$$

$$d_{jk} = |y_k - y_j|, \tag{6.18}$$

$$d_{ik} = |y_k - y_i|, \tag{6.19}$$

in the output space.

Substituting the input space inter-point distances into equation (6.13) gives the simplification:

$$a = \frac{y_i(d^*_{jk} - d^*_{ik} + d^*_{ij}) - d^*_{ik}(y_j - y_i) + d^*_{ij}(y_k - y_i)}{d^*_{ij}d^*_{ik}d^*_{jk}}, \tag{6.20}$$

$$= \frac{d^*_{ij}(y_k - y_i) - d^*_{ik}(y_j - y_i)}{d^*_{ij}d^*_{ik}d^*_{jk}}, \tag{6.21}$$

since $d^*_{ij} + d^*_{jk} = d^*_{ik}$.

The above expression can be further simplified by considering that for a structure-preserving mapping, the scaling of the input points is effectively arbitrary, so d^*_{ik} may be set to 1. In addition, it is also necessary to fix two of the output points, or the curvature can be minimised by collapsing them to a single point. Therefore, consider that during optimisation of the overall configuration, it is the case that $(y_k - y_i) = d^*_{ik} = 1$, leaving y_j and thus d_{ij} (or d_{jk}) as free variables. Also, make the assumption that $y_j > y_i$, such that $d_{ij} = (y_j - y_i)$. While this assumption is unrealistic, it will be seen shortly that it is not significant. Given this:

$$a = \frac{d^*_{ij} - d_{ij}}{d^*_{ij}d^*_{jk}}, \tag{6.22}$$

and so

$$E_C = \frac{4}{(d^*_{ij}d^*_{jk})^2}(d^*_{ij} - d_{ij})^2. \tag{6.23}$$

If $y_j < y_i$, then the first expression is no longer valid as indicated previously. However, if $y_j < y_i$ then $y_j < y_k$, which gives $d_{jk} = y_k - y_j$ and so by alternative symmetric derivation:

$$E_C = \frac{4}{(d^*_{ij}d^*_{jk})^2}(d^*_{jk} - d_{jk})^2. \tag{6.24}$$

Equation (6.23) in this simplified and constrained example backs up the intuition that minimisation of STRESS implicitly reduces curvature, as it exactly corresponds to a term from the Sammon STRESS measure, scaled by some constant. As a mapping is generated, and the inter-point distances become approximately correct, the assumptions inherent in the above analysis become realistic, and it would be expected that on *local* scales, curvature would decrease. This notion is intuitively extended to higher dimensional, more realistic, mappings.

There is no such guarantee for *global* distance relationships, which may be exemplified by the SPHERES_3 dataset. At the point in input space where the outer two spheres are 'opened out', the curvature must be high as neighbouring points one one side of the 'tear' are still nearby in output space, but neighbours on the other side of the partition become relatively highly distant.

Even in the single dimension case considered, if the distances between points in the input space are sufficiently large such that the quadratic approximation to the RBF transformation is invalid, then it is possible for there to be zero STRESS (locally), whilst still non-zero curvature. A simple such example is given for a cubic interpolant in figure 6.11.

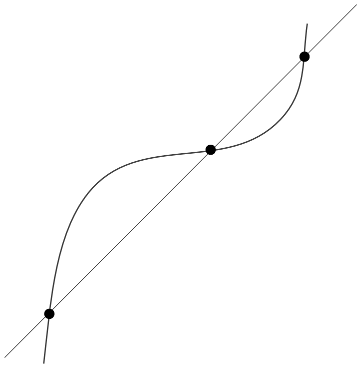


Figure 6.11: A correct mapping of three points from one-dimension to one-dimension, interpolated by a cubic.

The overall result from this analysis and discussion is that, in general, higher STRESS implies higher curvature, while lower STRESS implies lower curvature on local scales for sufficiently smooth interpolating functions. This summary has implications for the desired form of the neural network transformation, which will be considered in Section 6.6 shortly.

In fact, the reduction of curvature during training is an inherent property of networks trained by relative supervision algorithms which is not shared by their architecturally identical, but *a posteriori*-trained, counterparts. This distinction is investigated further in the following section, and its cause subsequently explained in Section 6.6.

6.5 Contrast Between NEUROSCALE and A Posteriori Fitting of an RBF Transformation

It is, of course, possible to produce a transformational Sammon Mapping by generating a configuration of points according to Sammon's standard procedure, and then fitting some model to this *a posteriori*. Indeed, this was the approach adopted by Cox and Ferry [1993] for discriminating future observations where a linear or quadratic model was fitted to a previously generated non-metric multidimensional scaling configuration. This approach could similarly be applied to the training of a neural network model, and it is this method that has been used to generate the previous plot of STRESS against curvature, for an RBF with 10 basis functions, in figure 6.7.

It is revealing to consider a similar *a posteriori* plot for an RBF with 45 basis functions, again with $\sigma = 1.0$, and superimpose a trace of the evolution of curvature of an identical RBF, during training by the relative supervision procedure of NEUROSCALE. This comparison of curvature values is given in figure 6.12. As the curvature of the transformation should be related to the generalisation performance of the networks, a plot of the test error, for the same models, is given in figure 6.13. The test dataset comprised a separate 45 patterns from the full Iris data.

The most significant feature of the two graphs in figures 6.12 and 6.13 is that configurations generated during the training of NEUROSCALE exhibit significantly lower curvature, and hence lower test error and better generalisation, than those configurations, with identical STRESS, generated by *a posteriori* fitting to Sammon mappings. It should be emphasised that that the RBF models in both cases have

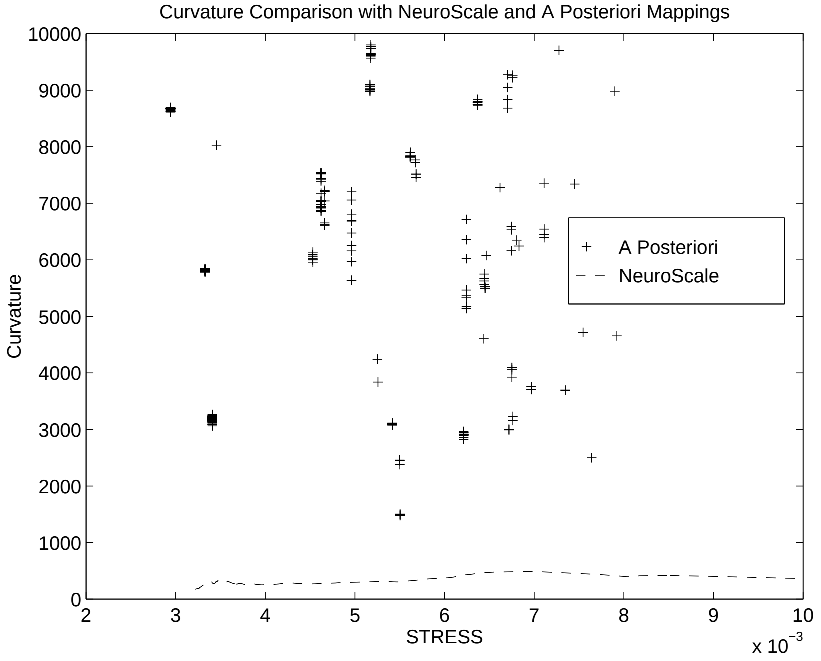


Figure 6.12: Curvature of *a posteriori*-fitted RBF networks compared to that during training of a NEUROSCALE model.

identical architectures and fixed basis functions; only the output layer weights are different.

Consider a NEUROSCALE configuration along the trace of figure 6.12, with STRESS 0.003325. The sum of the squared weights is 79.56. For a similar, *a posteriori* trained RBF, with STRESS 0.003327, this figure is 1.56×10^8 — very considerably greater. It is the smoothness constraint within the RBF model, as a result of the width parameter σ, which causes the ensuing trajectory through configuration space to be markedly different, in terms of curvature, to the corresponding trajectory of Sammon's algorithm — even though the STRESS values are comparable. The underlying cause of this alternative trajectory will be made more explicit in Section 6.6.2 later, where two apparently *identical* configurations, generated by NEUROSCALE and in *a posteriori* fashion, are shown to exhibit dramatically different generalisation errors.

The configurations that NEUROSCALE generates during training are not local minima for Sammon's algorithm, and this may be demonstrated by initialising a Sammon mapping with the final configuration from the above figures, and running that algorithm on the points alone. At each step of the mapping, the configuration is stored and an identical RBF to that of the NEUROSCALE model trained *a posteriori* to reproduce this configuration The trajectory of the networks trained during this Sammon procedure is thus superimposed on figure 6.14, and illustrates how curvature increases dramatically for minimal improvement in STRESS.

The results of this section, particularly figure 6.13, show clearly that training a neural network to perform a topographic transformation by a relative supervision method leads to significantly superior generalisation performance, on the IRIS_45 data set studied here, than that obtained by the *a posteriori* fitting of an identical network to a Sammon mapping. The reason for this behaviour is quite subtle, and will be revealed in the discussion of regularisation in the next section.

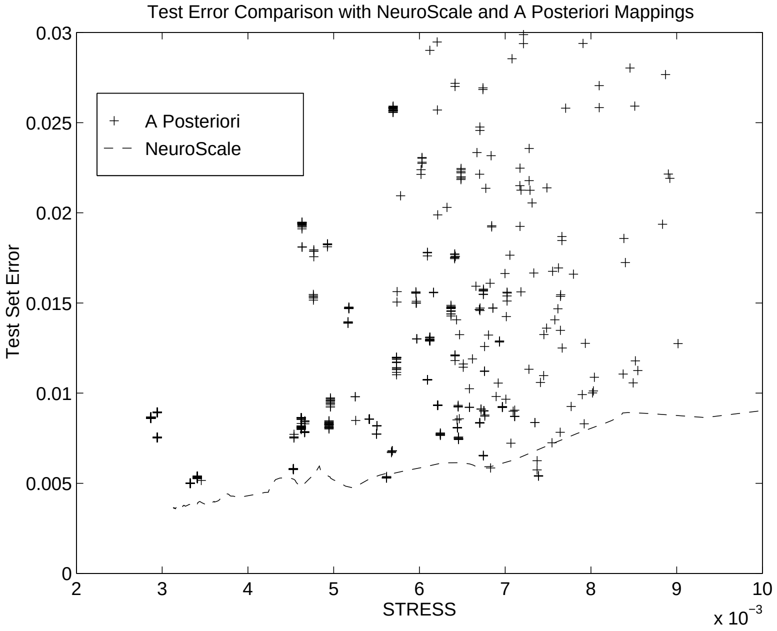


Figure 6.13: Test error of *a posteriori*-fitted RBF networks compared to that during training of a NEUROSCALE model.

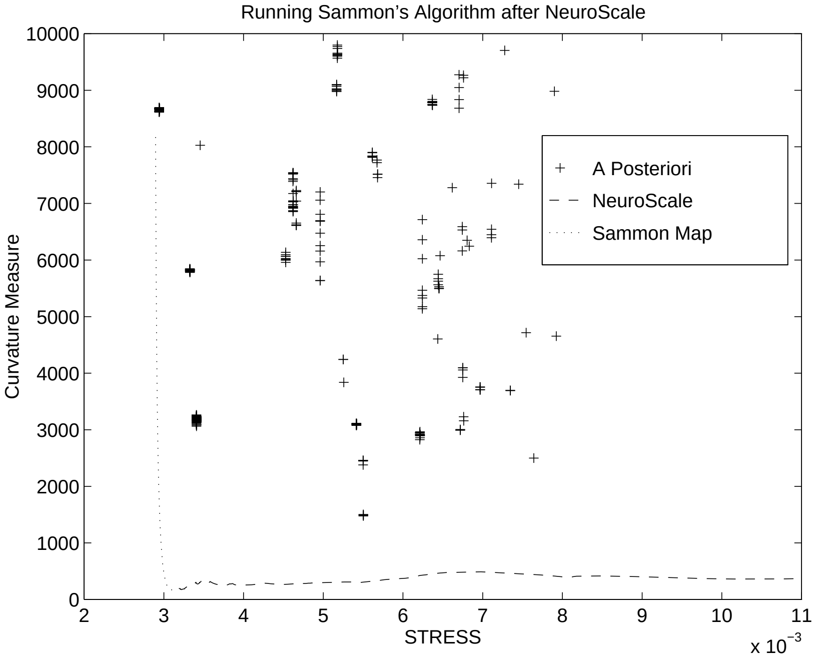


Figure 6.14: A Sammon Mapping, run after initialisation from the final configuration of the NEUROSCALE model. The dashed line illustrates the evolution of curvature and STRESS during NEUROSCALE training. At the minimum, training is stopped and the resultant configuration used as the initial step for a Sammon Mapping. For each step in the optimisation of the Sammon Map, the resulting configuration is reproduced by an RBF, and the network's curvature plotted along with the STRESS of the configuration itself.

6.6 Generalisation and Model Complexity

In many neural network applications, particularly function approximation and interpolation problems, the requirement of good *generalisation* is the salient design issue. In the general case of limited data, which has inherent additive noise, it is necessary to impose some form of *smoothness constraint* on the function. A model that fits all the training data in the presence of noise is likely to generalise, or interpolate, poorly. In the case of a model that is *over-fitted* (fits all the noise), the model's approximation of the function at a given test point will depend critically on the local noise in the training data. In this sense, the interpolation problem is *ill-posed* [Tikhonov and Arsenin 1977], due to this sensitivity to the noise.

There are two principal approaches to imposing smoothness, or limiting complexity, of a neural network model. Firstly, it is possible to reduce the complexity of the network by constraining the number of hidden nodes or basis functions, and this approach is known as *structural stabilisation*. Secondly, some form of penalty term can be introduced to the network error function, such as that given by the measure of equation (6.11) earlier, which discourages high-curvature models. This procedure is termed *regularisation*, and the degree of smoothness in this case is controlled by a hyperparameter. In either approach, the appropriate complexity of network transformation, and thus the number of basis functions or the value of the hyperparameter, will, in general, be unknown at the outset and must be determined from the data alone, often by employing some variety of cross-validation procedure.

6.6.1 Structural Stabilisation

With respect to structural stabilisation, the graph in figure 6.8, which illustrated the evolution of curvature during training for RBFs with 15, 30 and 45 basis functions, suggested that curvature was largely independent of the number of basis functions, indicating that generalisation performance might be likewise insensitive. Indeed, the generalisation error for RBFs trained on the IRIS_45 dataset and tested on another 45-pattern subset of the full Iris data is illustrated in figure 6.15. The RBFs comprised 5 to 45 basis functions, in steps of 5, and the error values were averaged over 25 runs.

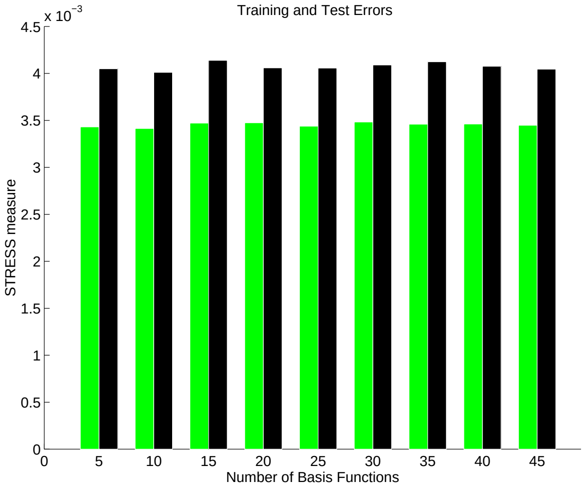


Figure 6.15: Training and test errors for NEUROSCALE RBFs with various numbers of basis functions. Training errors are on the left, test errors are on the right.

From figure 6.15 it can be seen that training and test error are roughly constant across the range of basis functions, 5 to the full complement of 45. There is no evidence of overfitting, and this is consistent with the previous evidence from figure 6.8. Reasons for this behaviour will be developed in the next subsection.

6.6.2 Regularisation

In contrast to the usual function approximation scenario where model complexity is unknown in advance, in the case of NEUROSCALE, and related models effecting such a topographic transformation, it may be reasoned that *the smoothness of the optimal network function is known a priori.*

While any particular arbitrary function with sufficient degrees of freedom may be fitted to a finite set of data and optimally[5] retain the topology (see the cubic in figure 6.11 for an example), it was shown in Section 6.4, in the one-dimensional case, that only a *perfectly smooth* function, with no curvature (or higher order terms), will exhibit optimal generalisation performance. This smoothness constraint must either persist through the entire space, or may be relaxed to only include the subspace containing the data, if this is known. This intuition may be extended to higher dimensions, where it should be apparent that non-zero curvature must imply structural distortion.

For the simple one-dimensional case given earlier in figure 6.10, it is evident that the ideal transformation is of the class:

$$y = \pm x + c, \tag{6.25}$$

where c is an arbitrary constant. The optimality property of this class of functions holds *regardless of the distribution of the data.* It is intuitive in this simple one-dimensional case that any transformation with gradient ±1 will retain the distance relationships between all possible pairs of data points.

Given that the optimal transforming function has no second-order or higher derivative terms, it is possible to more formally generalise the result for the simple one-dimensional example to higher dimensions, still with $p = q$. This permits an expression for the gradient, or first-order derivatives, of the network function to be obtained. It can be shown that, for exact structure preservation, the following relation must hold at each and every data point:

$$\forall x_m, m \in \{1 \dots p\} : \quad \sum_{l=1}^{q} \left(\frac{\partial y_{il}}{\partial x_m} \right)^2 = 1. \tag{6.26}$$

That is, considering a point $\mathbf{x}_i$ in input space and a point $\mathbf{x}' = \mathbf{x}_i + \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon}$ is an arbitrary vector, then the distance between the corresponding image points $\mathbf{y}_i$ and $\mathbf{y}'$ is $\| \boldsymbol{\epsilon} \|$ as required. This can be seen by referring to the Taylor expansions (which contain no second or higher order terms) around the point $\mathbf{x}_i$ of each of the output functions for each dimension:

$$\forall l, l \in \{1 \dots q\} : \quad y'_l = y_{il} + \boldsymbol{\epsilon}^{\mathrm{T}} \mathbf{g}_{il}, \tag{6.27}$$

where $\mathbf{g}_{il}$ is the gradient vector $(\partial y_{il}/\partial x_1, \dots, \partial y_{il}/\partial x_p)^{\mathrm{T}}$ evaluated at $\mathbf{x} = \mathbf{x}_i$, and from this the inter-point distances in the output space may be calculated thus:

$$\begin{aligned}
\| \mathbf{y}' - \mathbf{y}_i \|^2 &= \sum_{l=1}^{q} (y'_l - y_{il})^2, && (6.28)\\
&= \sum_{l=1}^{q} (\boldsymbol{\epsilon}^{\mathrm{T}} \mathbf{g}_{il})^2, && (6.29)\\
&= \boldsymbol{\epsilon}^{\mathrm{T}} \left(\sum_{l=1}^{q} \mathbf{g}_{il} \mathbf{g}_{il}^{\mathrm{T}} \right) \boldsymbol{\epsilon}, && (6.30)\\
&= \boldsymbol{\epsilon}^{\mathrm{T}} \mathbf{G}_i \boldsymbol{\epsilon}, && (6.31)
\end{aligned}$$

where the matrix $\mathbf{G}_i = \sum_{l=1}^{q} \mathbf{g}_{il} \mathbf{g}_{il}^{\mathrm{T}}$ and is distinct for every data point. For the corresponding distances to be retained, $\| \mathbf{y}' - \mathbf{y}_i \|^2 = \boldsymbol{\epsilon}^{\mathrm{T}} \boldsymbol{\epsilon}$, and so $\mathbf{G}_i = \mathbf{I}$, the identity matrix, which leads directly to the equalities of equation (6.26) above.

[5] The use of the term 'optimally' here is not intended to imply that all the distances are retained perfectly, but that the optimal Sammon configuration, given the constraint imposed by reducing dimension, can be reproduced by the network.

These gradient relationships can only hold exactly in the unrealistic case where there is no reduction in dimension. In practical applications, where $q < p$, $\mathrm{rank}(\mathbf{G}_i) = q$ and therefore $\mathbf{G}_i$ can never be equal to the identity matrix. In such instances, a low generalisation error implies that the magnitude of the residual $\epsilon^{\mathrm{T}}(\mathbf{I} - \mathbf{G}_i)\epsilon$ should be minimised. In the Sammon mapping, and the NEUROSCALE algorithm, the vectors ϵ of interest are the combinations of inter-point vectors $(\mathbf{x}_i - \mathbf{x}_j)$. It should therefore be the case that, for low STRESS on the training set, all potential inter-point vectors ϵ should lie in the range of $\mathbf{G}$. A simple example can be envisaged by considering data distributed over a 2-dimensional plane in a 3-dimensional space, with the neural network transforming the data points down to 2 dimensions. Here, the range of $\mathbf{G}_i$ at all the data points should be the linear subspace spanned by the plane. In figure 3.6 in Chapter 3, excellent generalisation was observed for a single cluster of datapoints taken from a set of four in a 'linear' configuration. Even though the significant Gaussian 'noise' component implies that the data lies in four dimensions, the inter-cluster distances dominate in the STRESS minimisation. The range of the $\mathbf{G}_i$ would thus be expected to include the axial direction along which the clusters are distributed, which would explain the low value of STRESS obtained when the test cluster is incorporated and the intuitively good visualisation evident in the figure.

However, for real data, it is likely to be the case that the minimum STRESS configuration will exhibit curvature at one or more points, with significantly different first derivatives, and thus $\mathbf{G}_i$s, in different regions of the data. The nature of these matrices may be informative, and further experimental inquiry might be appropriate for future research.

The following experiment included here, however, provides some insight into the above analysis and also offers further evidence of the distinction between NEUROSCALE and *a posteriori* mappings. Figures 6.16 and 6.17 illustrate the final configurations of a topographic mapping of the IRIS_45 dataset from the original four to two dimensions. The mappings have been aligned via a *Procrustes* rotation [Mardia, Kent, and Bibby 1979], and the goodness-of-fit (sum-of-squared distances between corresponding points) is 1.4492×10^{-5}, such that the two configurations may be considered to be identical. Superimposed on the plots is a colour scale which indicates a measure of the conformity of the mapping to the gradient constraint of equation (6.26), and is determined as follows.

For each point in the dataset, the unit vector to every other point, $(\mathbf{x}_i - \mathbf{x}_j)/d^*_{ij}$, was calculated, and the projections of the gradients of both network outputs in the direction of that vector were derived, then squared and summed. The squared deviation of this quantity from its ideal value of 1.0 was then accumulated over every such unit vector. This then gives a measure of the closeness of the gradient measure of equation (6.26) to unity along the directions of interest within the dataset. A simple interpolating surface[6] has been fitted to, and passes through, all the points. This surface is artificial (in that the interpolated values between the data points bear no real meaning) but it does permit a relatively clear visualisation of the expected generalisation accuracy in the output space.

For both figures, an identical radial basis function network comprising 45 fixed centres, located at the data points, was utilised. The basis functions were Gaussian with width $\sigma = 1$, and the results were generated by an identical segment of computer program, with the only difference between the two plots being the weight matrix loaded. Despite the fact that both configurations are similar (apart from reflection, rotation and translation), the NEUROSCALE transformation of figure 6.16 will clearly offer better generalisation performance as the gradient expression is much closer to unity (average error = 0.0027) than for the *a posteriori* mapping (average error = 0.0369). This is confirmed by the values of test error given in table 6.1.

The phenomenon that networks of identical architecture generating identical configurations (with respect to STRESS) exhibit significantly different generalisation performance is again related to the equivalence of solutions under rotation and translation. (This will be considered in more detail in Section 6.7 shortly.) The solution generated by the Sammon Mapping is generally arbitrary, in terms of rotation and translation, and the resulting RBF transformation fitted to this configuration can in turn be expected to exhibit arbitrary generalisation error. However, it can be demonstrated that the relative supervision learning algorithm should lead to better generalisation performance as it *tends to*

[6] This was automatically generated by the software package MATLAB using an inverse-distance method.

	NeuroScale	*A Posteriori*
Training Stress	0.00282294	0.00282294
Test Stress	0.0037395	0.00866962
Curvature	417.495	11620.4
$\|\mathbf{W}\|^2$	2.1315×10^7	1.0437×10^8

Table 6.1: Stress and curvature values for the two gradient mappings.

favour solutions with smaller weights.

Firstly, consider the expression for the update of a single dimension of the network output from Section 6.2.2:

$$\mathbf{z}^{t+1} = \mathbf{z}^t - \eta \mathbf{J}\mathbf{J}^{\mathrm{T}}\delta^t, \tag{6.32}$$

$$= \mathbf{z}^t - \sum_k \lambda_k^2 \beta_k \mathbf{u}_k. \tag{6.33}$$

Now, if initially $\mathbf{z}^0$ is close to the origin, then because of the effect of the λ_k^2 term, the final vector $\hat{\mathbf{z}}$ *will have generally larger projections onto the principal eigenvectors of* $\mathbf{J}\mathbf{J}^{\mathrm{T}}$. So, if $\hat{\mathbf{z}} = \sum_k \gamma_k \mathbf{u}_k$, in general, $\gamma_{k+1} > \gamma_k$. Thus, of all the equivalent solutions, NeuroScale favours those $\hat{\mathbf{z}}$ which lie mainly in the principal subspace of $\mathbf{J}\mathbf{J}^{\mathrm{T}}$. A very simple illustration of this effect is given in figure 6.18 for three output points in two dimensions (for which a Stress=0 solution exists), for full-rank, but ill-conditioned, $\mathbf{J}$. Plotted on the figure are the respective eigenvector components γ_k of $\mathbf{z}^t$ as it evolves during training. The smallest component does not visibly change, while the component in the direction of the principal eigenvector becomes dominant.

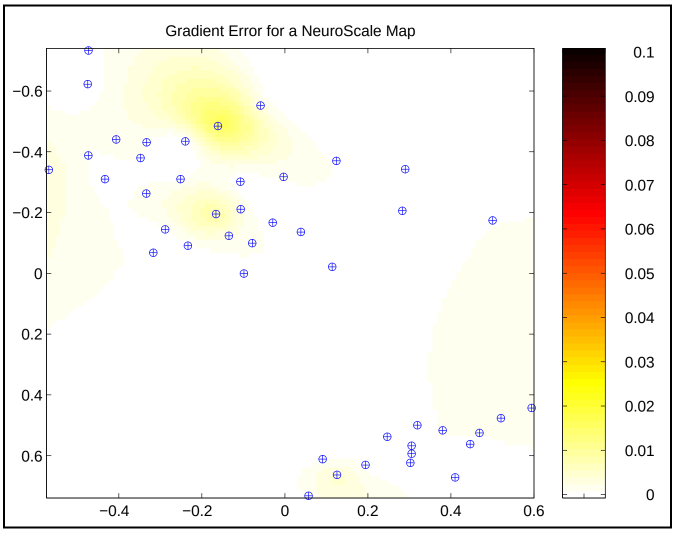


Figure 6.16: Two-dimensional topographic mapping of the IRIS_45 dataset, generated by NEUROSCALE. At each point, the squared deviation of the gradient measure from the ideal value of unity is superimposed with a colour scale.

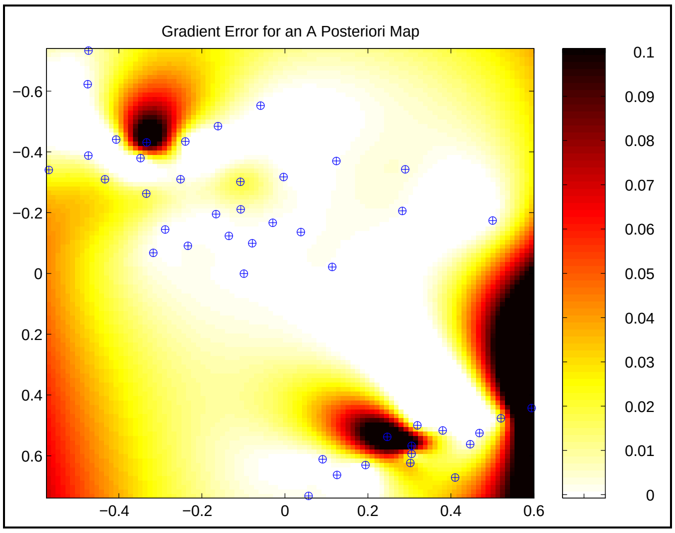


Figure 6.17: Two-dimensional topographic mapping of the IRIS_45 dataset, generated by *a posteriori* fitting of an RBF to a Sammon Mapping. At each point, the squared deviation of the gradient measure from the ideal value of unity is superimposed with a colour scale.

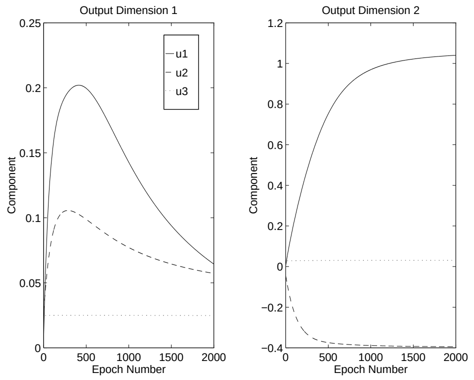


Figure 6.18: Evolution of the eigenvector components of a solution trained by NEUROSCALE.

For the two final configurations of figures 6.16 and 6.17, the values of γ_k^2 are given, for $20 \leq k \leq 45$, in figure 6.19 for both output dimensions. The relative supervision solution clearly exhibits lower values of γ_k for larger k (the minor eigenvectors).

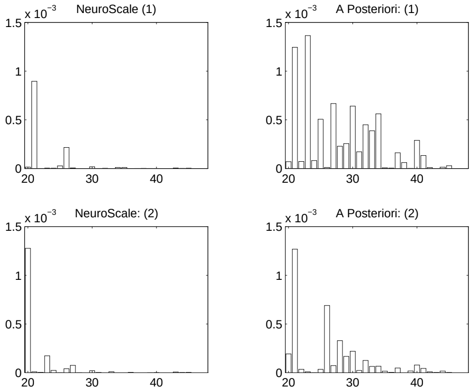


Figure 6.19: Final eigenvector components of the solutions from figures 6.17 and 6.16.

A more dynamic exposition of this effect is given in figures 6.20 and 6.21, again for the IRIS_45 dataset with full-rank **J**. For the first of the two outputs of the network, the evolution of the (absolute) direction cosines of the vector $\mathbf{z}_1$ with the 45 eigenvectors $\mathbf{u}_k$ is shown for both a Sammon Mapping and a gradient-descent NEUROSCALE network. For the former, no discernable pattern amongst the direction cosines is visible during the 40 training cycles. For NEUROSCALE, the vector $\mathbf{z}_1$ is clearly evolving such that it lies along the direction of the principal axes $\mathbf{u}_1$ and $\mathbf{u}_2$ of **J**, and is approximately orthogonal to all other eigenvectors.

Now, for the single network output solution vector $\hat{\mathbf{z}}$ generated after training with arbitrary rank $\mathbf{J}$,

$$\begin{aligned}
\mathbf{J}\mathbf{w} &= \hat{\mathbf{z}}, \quad \text{so} && (6.34)\\
\mathbf{w} &= \mathbf{J}^{+}\hat{\mathbf{z}}, && (6.35)\\
&= \mathbf{J}^{+}\sum_k \gamma_k \mathbf{u}_k, && (6.36)\\
&= \mathbf{V}\mathbf{S}^{-1}\mathbf{U}^{\mathrm{T}}\sum_k \gamma_k \mathbf{u}_k, \quad \text{by singular value decomposition}, && (6.37)\\
&= \mathbf{V}\mathbf{S}^{-1}\gamma, && (6.38)
\end{aligned}$$

where $\mathbf{S}$ is the diagonal matrix of singular values of $\mathbf{J}$ and $\gamma = (\gamma_1, \gamma_2, \ldots, \gamma_h)^{\mathrm{T}}$.

The sum-of-squared weights in $\mathbf{w}$ is:

$$\begin{aligned}
\| \mathbf{w} \|^2 &= \mathbf{w}^{\mathrm{T}}\mathbf{w}, && (6.39)\\
&= \gamma^{\mathrm{T}}\mathbf{S}^{-2}\gamma, && (6.40)\\
&= \sum_k \left(\frac{\gamma_k}{s_k}\right)^2, && (6.41)
\end{aligned}$$

and for the complete weight matrix $\mathbf{W}$,

$$\begin{aligned}
\| \mathbf{W} \|^2 &= \sum_l \| \mathbf{w}_l \|^2, && (6.42)\\
&= \sum_l \sum_k \left(\frac{\gamma_{kl}}{s_k}\right)^2. && (6.43)
\end{aligned}$$

So, for a given $\sum_{kl} \gamma_{kl}^2$, the sum-of-squared weights is minimised by distributing the ‘mass’ of γ_{kl}^2 over the largest singular values s_k. However, since $\sum_{kl} \gamma_{kl}^2 = \| \mathbf{Y} \|^2$, then $\sum_{kl} \gamma_{kl}^2$ is minimised when the points $\mathbf{Y}$ are centred at the origin, and this should also serve to reduce the norm on the weights. For the *a posteriori* configuration in figure 6.17, $\| \mathbf{Y} \|^2 = 13.3$, whereas for its NEUROSCALE counterpart in figure 6.16, this measure is 59.9. This implies, from the observations in this chapter, that the tendency of NEUROSCALE to distribute the γ_{kl} in the direction of the principal eigenvectors of $\mathbf{J}$ more than compensates for the overall increase in the squared-magnitudes of those factors.

Thus, the relative supervision algorithm within NEUROSCALE must seek, amongst all candidate solutions of the STRESS optimisation procedure, those with generally lower values of $\| \mathbf{W} \|^2$.

This norm on the weights is effectively a regularisation term, and in a supervised neural network setting, is known as *weight decay* [Hinton 1989]. Such a term, when used as a penalty function in network optimisation, generally reduces the curvature of the final transformation and improves generalisation. This reduction in both sum-of-squared weights and curvature was indeed observed for the example transformations given previously in this section, and listed in table 6.1.

The implication is that topographic transformations generated by the NEUROSCALE algorithms are effectively *self-regularising*, in that they incorporate an implicit weight decay element. Because this regularisation is a side-effect of the operation of relative supervision, there is no control over its magnitude. However, given that it was reasoned previously that topographic transformations should be as smooth as possible, any regularisation component in the algorithm should be beneficial in terms of generalisation performance.

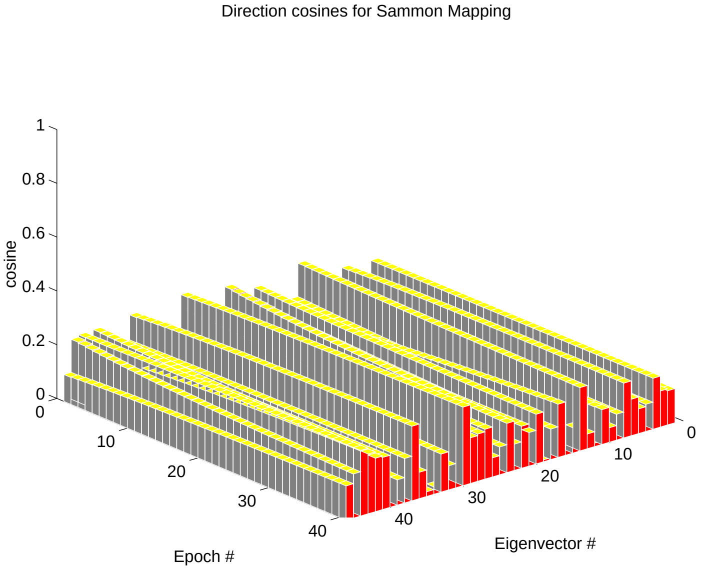


Figure 6.20: Evolution of the direction cosines of one output dimension for a Sammon mapping of the IRIS_45 dataset.

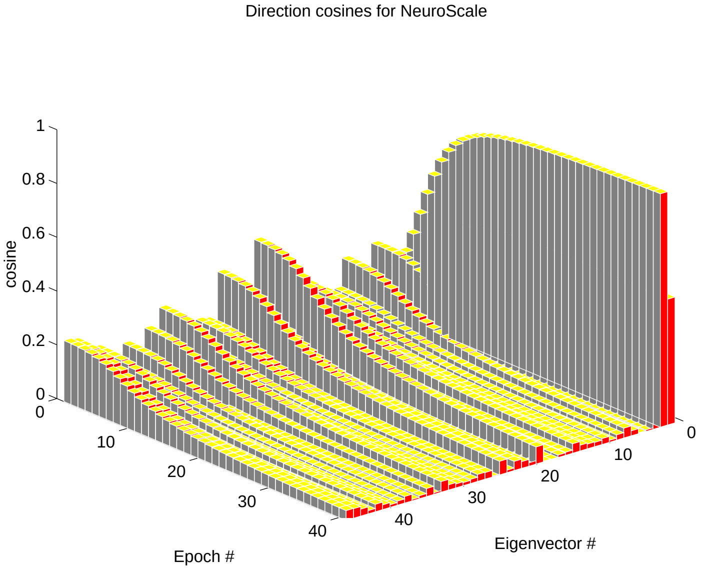


Figure 6.21: Evolution of the direction cosines of one output dimension for NEUROSCALE on the IRIS_45 dataset.

6.6.3 Effect of Width of Gaussian Basis Functions

In the case of RBF networks as incorporated in NEUROSCALE, the global width parameter, σ, employed by Gaussian basis functions is effectively a smoothing hyperparameter. The larger the value of σ, the smoother the resultant transformation will be. On the basis of the previous discussion in this section, it would be anticipated that optimum generalisation performance should be expected for larger values of σ. There is a caveat, however, in that in the pathological case where the basis function width is infinite, there is no discrimination between any of the data points. In addition, large values of σ would be expected to imply deteriorating performance due to the problems of finite numerical accuracy in the practical computation.

Plots of training and test STRESS against basis function width, for the IRIS_45 data, are given in figures 6.22 and 6.23. The network employed comprised as many basis functions, 45, as patterns. The error values given in these figures were obtained by the 'shadow-targets' training algorithm which is presented in Section 7.3 in the next chapter, as this produces better minima. As was demonstrated in Section 6.2.2, the final *training* STRESS using the standard relative supervision algorithm is dependent on the value of σ. The shadow-targets algorithm enables the minimum realisable STRESS value to be obtained for all (reasonable) values of σ and thus permits a fairer assessment of the effects of σ on *test* STRESS. The contrasts between these two approaches to training NEUROSCALE will be developed further in the next chapter.

In the first plot, the range of σ is from 0 to 100, while the latter plot is a larger scale version of the former, covering the range of values 0 to 20, and in addition, illustrating the values of σ where successive singular values of the matrix **J** become numerically zero. [7]

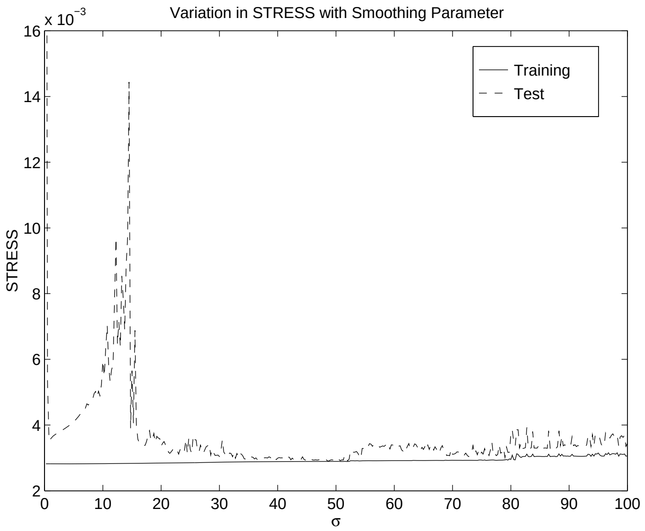


Figure 6.22: Training and test STRESS, for the IRIS_45 dataset, as a function of the basis function parameter σ.

The plot of training STRESS in figure 6.22 demonstrates that this quantity is almost insensitive to the choice of basis function width. There is a gentle increase in value with σ, due to the reduced accuracy of the computer arithmetic. In the case of test STRESS, it is apparent that after an intimal minimum at $\sigma = 1.00$, test STRESS actually *increases* and, after considerable oscillation, decreases to a value comparable with the training error. Figure 6.23 indicates that the test error is smooth up to the point where the smallest singular value of the Jacobian matrix is set to zero in the pseudo-inverse routine.

In general, the form of the test STRESS curve is consistent with the hypothesis that large width should imply better generalisation. It is also evident that test error begins to increase again, slowly, at about

[7] Machine accuracy, ϵ_m, for double precision arithmetic on a SUN Sparc10 workstation is $\approx 2.22 \times 10^{-16}$. Any singular value that is smaller than $\epsilon_m \times$ the largest singular value is discarded in the pseudo-inverse algorithm.

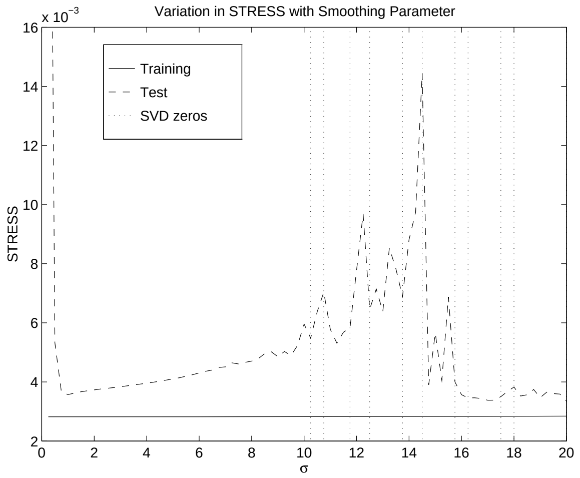


Figure 6.23: Training and test STRESS, for the IRIS_45 dataset, as a function of the basis function parameter σ. The dotted vertical lines indicate points at which successive singular values of the Jacobian become too small to be reliably computed and are thus set to zero.

$\sigma = 50$, and this is likely to be the result of degrading numerical accuracy.

The initial segment of the test error curve, however, exhibits a definite increase in generalisation error. One hypothesis to explain this shape is that this observed maximum may be due to the particular distribution of the Iris data used in the experiment. However, figure 6.24 illustrates similar curves for the topographic mapping of a simple spherical Gaussian cluster in three dimensions down to two.

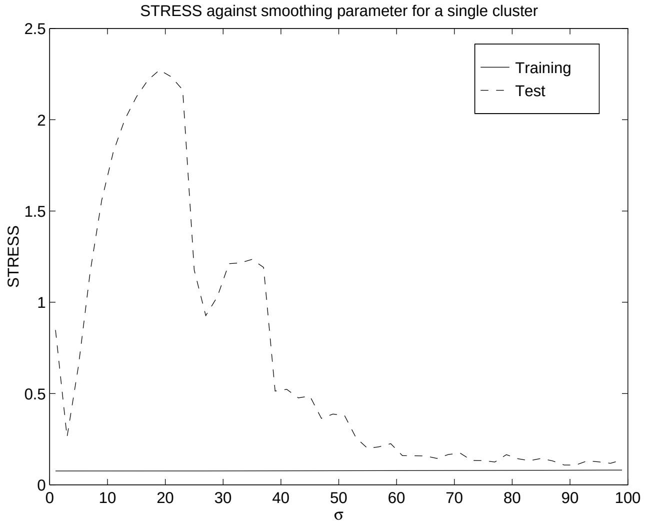


Figure 6.24: Training and test STRESS, for a spherical Gaussian cluster, as a function of the basis function parameter σ.

For the single cluster, best generalisation is again obtained with larger σ. Once more, however, there is a characteristic maximum at a relatively low value of width parameter. The cause of this apparent anomaly remains unknown and warrants further investigation.

6.7 Rotation and Translation of Solutions

It is clear that STRESS is invariant under arbitrary rotation, reflection and and translation of an output configuration $\mathbf{Y}$. What is less obvious, is how these perturbations to the solution effect the weights in the radial basis function network. As has already been noted, NEUROSCALE tends to select a solution which minimises the norm of the weight matrix, so this section will consider how this measure changes under rotation and translation. (Reflection only changes the sign of the weights.)

In both cases, let the base-line solution and associated weights be defined by the equation:

$$\mathbf{J}\hat{\mathbf{W}} = \hat{\mathbf{Y}}, \tag{6.44}$$

implying that $\hat{\mathbf{W}} = \mathbf{J}^{+}\hat{\mathbf{Y}}$.

Rotation

A rotated solution $\mathbf{Y}$ is given by $\mathbf{Y} = \hat{\mathbf{Y}}\mathbf{R}$, where $\mathbf{R}$ is an orthogonal rotation matrix. Then the new weights $\mathbf{W}$ are given by $\mathbf{W} = \mathbf{J}^{+}(\hat{\mathbf{Y}}\mathbf{R})$, and the sum-of-squared weights is:

$$\operatorname{tr}\left[\mathbf{W}\mathbf{W}^{\mathrm{T}}\right] = \operatorname{tr}\left[\mathbf{J}^{+}\hat{\mathbf{Y}}\mathbf{R}\mathbf{R}^{\mathrm{T}}\hat{\mathbf{Y}}^{\mathrm{T}}(\mathbf{J}^{+})^{\mathrm{T}}\right], \tag{6.45}$$

$$= \operatorname{tr}\left[\mathbf{J}^{+}\hat{\mathbf{Y}}\hat{\mathbf{Y}}^{\mathrm{T}}(\mathbf{J}^{+})^{\mathrm{T}}\right], \tag{6.46}$$

$$= \operatorname{tr}\left[\hat{\mathbf{W}}\hat{\mathbf{W}}^{\mathrm{T}}\right]. \tag{6.47}$$

Therefore, rotation of the solution does not affect the (Frobenius) norm of the weight matrix.

Translation

For translation, each dimension of the output can be considered in isolation, as shifts along each axis can be applied independently. Then for a single dimension, a translated solution $\mathbf{z}$ is given by $\mathbf{z} = \hat{\mathbf{z}} + k\mathbf{1}$, where k is a scalar determining the amount of translation and $\mathbf{1}$ is the N-vector of 1's. Then the weights $\mathbf{w}$ for that output dimension are:

$$\mathbf{w} = \mathbf{J}^{+}(\hat{\mathbf{z}} + k\mathbf{1}), \tag{6.48}$$

$$= \mathbf{J}^{+}\hat{\mathbf{z}} + k\mathbf{J}^{+}\mathbf{1}. \tag{6.49}$$

$$= \hat{\mathbf{w}} + k\mathbf{j}, \tag{6.50}$$

where $\mathbf{j} = \mathbf{J}^{+}\mathbf{1}$.

Then the norm-squared weights are:

$$\mathbf{w}^{\mathrm{T}}\mathbf{w} = (\hat{\mathbf{w}} + k\mathbf{j})^{\mathrm{T}}(\hat{\mathbf{w}} + k\mathbf{j}), \tag{6.51}$$

$$= \hat{\mathbf{w}}^{\mathrm{T}}\hat{\mathbf{w}} + k^2\mathbf{j}^{\mathrm{T}}\mathbf{j} + 2k\hat{\mathbf{w}}^{\mathrm{T}}\mathbf{j}. \tag{6.52}$$

To minimise this measure, differentiation with respect to k gives:

$$\frac{d}{dk}(\mathbf{w}^{\mathrm{T}}\mathbf{w}) = 2k\mathbf{j}^{\mathrm{T}}\mathbf{j} + 2\hat{\mathbf{w}}^{\mathrm{T}}\mathbf{j}. \tag{6.53}$$

and setting this equal to zero allows a value of k to be determined thus:

$$k = -\frac{\hat{\mathbf{w}}^{\mathrm{T}}\mathbf{j}}{\mathbf{j}^{\mathrm{T}}\mathbf{j}}, \tag{6.54}$$

that minimises the weight-norm, as the second derivative of equation (6.53) is positive. Substituting this value back into equation (6.51) gives the minimum value of the weight norm:

$$(\mathbf{w}^{\mathrm{T}}\mathbf{w})_{min} = \hat{\mathbf{w}}^{\mathrm{T}}\hat{\mathbf{w}} - \frac{\hat{\mathbf{w}}^{\mathrm{T}}\mathbf{j}}{\mathbf{j}^{\mathrm{T}}\mathbf{j}}. \tag{6.55}$$

It is therefore possible for a particular configuration, given the network Jacobian, to determine analytically the translated solution that minimises the norm of the weight matrix. (This is only possible with a linear model where **J** is equivalent to the design matrix, and is therefore not possible with an MLP where a nonlinear optimisation approach would be required.) However, it should be noted from equation (6.55) that this minimum value depends upon $\hat{\mathbf{w}}$, which *is* dependent upon rotation. So while an arbitrary rotation does not affect the weight norm of a given configuration, it does alter the class of translated solutions that may be obtained. Furthermore, there is no analytic solution for the optimal $\hat{\mathbf{w}}$, as the weight vectors for each output dimension are mutually dependent. To select the rotation and translation parameters that produce the smallest weight-norm is therefore a difficult nonlinear optimisation problem, but one which in theory could be tackled in order to produce an *a posteriori* mapping with low weight values and, by implication, good generalisation.

6.8 Conclusions

6.8.1 Local Minima

It is known that the generation of Sammon Mappings is considerably complicated by the tendency of the algorithm to be captured in sub-optimal local minima. This is true even for simple problems, which was illustrated for a subset of the Iris data in Section 6.2.1. These poor minima tend to represent *unsmooth* transformations of the input data, as demonstrated in Section 6.3, which is why Gaussian radial basis function networks with significant values of basis function width exhibit very few local minima problems.

In fact, when exploiting such networks to minimise STRESS, *two* distinct classes of weight local minima were discerned. The first class are those weights which give rise to configurations of points that are local minima of STRESS and thus identical to those of the Sammon mapping. The theoretical second class are those introduced by the network itself as a result of the null-space of the Jacobian matrix (the matrix of basis function outputs for every pattern). However, the relative supervision algorithm is not caught by this second class of minima, as there is an escape route offered by the set of multiple rotated / translated solutions.

Nevertheless, NEUROSCALE was observed to effectively cease training before attaining the 'global' minimum. This was shown to be caused by the dynamics of the weight update equation, which dramatically slows the training process when the error vectors are in the direction of the eigenvectors of the Jacobian matrix which have small eigenvalues relative to those of the principal eigenvectors. This behaviour is therefore exaggerated as the condition number of the Jacobian increases.

The conditioning of this matrix is dependent upon the width of the basis functions. A relatively small value of width, σ, will eliminate those problems discussed in the previous paragraph, which become more problematic as σ is increased. However, increasing σ additionally reduces the likelihood of capture in the first class of sub-optimal minima, as the majority of these are unsmooth and thus unrealisable by a network with high σ. There is an evident trade-off here.

6.8.2 Model Complexity

There is a fundamental point to be made concerning the functional form of a neural network effecting a topographic transformation. In Section 6.4 it was shown that in the case of a simple one-dimensional mapping, minimisation of STRESS also implicitly reduces curvature in regions of data where the network transformation can be usefully approximated by a quadratic. This notion can be generalised to higher dimensions, where it is intuitive that for minimum-STRESS configurations, improved generalisation error will be observed when curvature is minimised.

The issue of smoothness is highly relevant to the question of controlling model complexity. A significant result from Section 6.6.1 was that generalisation error for NEUROSCALE was independent of the number of basis functions in the network. In fact, it was argued in Section 6.6.2, that in contrast to many other application domains of neural networks, the smoothness of the desired model in a topographic context is known *a priori*. It was reasoned that the network function should ideally have zero second- and higher-order derivatives, and an expression for the optimal first-order, gradient, terms was derived. The relevance of this result was demonstrated for two apparently identical training-set configurations which nevertheless exhibited considerable differences in generalisation capacity for the test set.

These observations have important implications. In a regression setting, with noise on the target (response) variables, model complexity must be constrained through, for example, regularisation. The best model, in terms of predictions for future data, will generally not fit all the data points. In the topographic context, there are no explicit targets, but the *relative* targets that are generated from the input data itself, can be considered as being *noise-free*. These targets are still precise even in the presence of noise on the input variables.

Noise on the input variables does influence the mapping, and in certain circumstances, dramatically so. A good example of this is the SPHERES_3 dataset, where to minimise STRESS, the outer sphere is 'opened out' where the data is at its least dense on the sphere. Consider, then, data drawn from a near-uniform distribution, such that there is a small region where the data is marginally less dense and where the 'tear' in the sphere naturally forms. It can be seen that the location of the tear can be highly sensitive to added input noise, and the problem of determining the transformation can be interpreted as being *unstable* [Tikhonov and Arsenin 1977]. However, in contrast to a regression problem, this instability is not a result of too complex a model, as the form (curvature) of the topographic transformation will remain similar to the ideal (based on the known input distribution) wherever the tear occurs on the sphere; what is occurring is that the function is effectively 'shifted' in the input space. There is therefore no sense in which regularisation, or other forms of limiting model complexity, can satisfactorily stabilise the problem.

That considered, the SPHERES_3 data is a notably pathological distribution, as the merit of topographically reducing the dimension of data which is near-uniformly distributed in a higher-dimensional space is questionable. It is appropriate, therefore, that the ideal topographic transformation should strictly 'interpolate' all the 'target' data points while being of minimal complexity, or alternatively, of maximal smoothness. This implies that the use of as many basis functions as data points with large values of basis function width (σ) should give optimal performance, as seen in figure 6.22. Nevertheless, the use of fewer, but data-representative, basis functions may still be appropriate for computational reasons, although no significant investigation of this issue has been included in this thesis.

Experimentally, the curvature of the transformation produced by NEUROSCALE was observed to generally decrease during the training process. This is an inherent property of the weight-updating dynamics of the relative supervision algorithm which tends to implicitly reduce the sum-of-squared network weights, and is thus effectively incorporating a *weight decay* component into the training.

However, for supervised training of a neural network, weight decay trades-off the squared-weights penalty term for the error term. The net result of this is that the error on the training set increases from its potential minimum, and the network no longer fits the data points, with the motivation that generalisation to unseen data will be improved. This is *not* the case in NEUROSCALE, where the weight decay component enacts a different rôle. Remember that there is no single unique solution of the mapping process. Any rotation or translation of a particular solution is itself a solution, as it will exhibit the same measure of STRESS. The function of weight decay in the topographic context, therefore, is to select solutions with smaller weights, and by implication, generally lower curvature.

This is of particular relevance when considering *a posteriori* mappings, where a model is fitted, in the standard supervised manner, to the output configuration previously generated by a Sammon Mapping procedure. In Section 6.5, it was illustrated that models fitted in such a manner exhibited higher curvature and poorer generalisation than trained NEUROSCALE networks. The distinction was made very clear in table 6.1, where otherwise identical, but *rotated* and *translated*, configurations exhibited dramatic differences in STRESS on the test dataset. The NEUROSCALE approach was seen to be much superior in this respect. In fact, it was shown in Section 6.7 how the sum-of-squared weights, and by implication the curvature, depends on the rotation and translation of configurations. The preferred values of these two factors could potentially be determined using a nonlinear optimisation approach in order to produce a *a posteriori*-trained RBF that exhibited comparable generalisation to the equivalent NEUROSCALE network. However, the relative supervision algorithm combined with the smoothing of the RBF implicitly achieves a similar end via a single optimisation procedure, and offers the potential to avoid local minima because of the smoothing element.

6.8.3 Objective and Subjective Mappings

It should be emphasised that the above remarks concerning the model complexity are only relevant to purely topographic, or objective mappings, as implemented by NEUROSCALE with $\alpha = 0$. Of particular note is the earlier analysis concerning the gradient and curvature of topographic mappings, as the incorporation of supervisorial influence implies regions of higher curvature at, for example, class boundaries.

Nevertheless, the weight-decay implicit within NEUROSCALE will still be present in the extraction of supervised feature spaces, and thus implies the presence of a smoothing effect even when class information is allowed to influence the mapping. This phenomenon is one hypothesis for the good generalisation observed when using an $\alpha = 0.5$ NEUROSCALE projection as a pre-processing stage in a prediction model in Chapter 4. This possible factor should be considered as part of any future investigation in that direction.

Chapter 7

Optimising Topographic Transformations

7.1 Introduction

It has been previously noted that the storage and computational requirements of the Sammon mapping grow in the order of the square of the number of data points to be mapped. Any neural network implementation will also suffer from this undesirable $O(N^2)$ scaling behaviour.

For an efficient implementation of Sammon's algorithm, all the inter-point distances in the data space are calculated *a priori* and stored in a $(N \times N)$ matrix. For a thousand data points, and only single precision arithmetic, this would require 4 megabytes (MB) of storage. Even a machine with 64MB of dedicated memory could only store a 4000-pattern distance matrix.[1] For such large numbers of patterns, the inter-point distances would need to be calculated, repeatedly, on-the-fly.

This in turn will impact on the already considerable computational demands. Each calculation of the STRESS measure necessitates a loop of $N(N-1)/2$ distance calculations. Calculation of the gradients (for minimisation routines) requires a similar number of cycles, with additional vector operations.

There have been two characteristic approaches to alleviate the difficulties of training STRESS-based mappings. Firstly, various optimisation schemes have been considered for standard mapping procedures which will be covered in the next section. In Section 7.3, a new, and highly effective, algorithm is proposed for the training of topographic models whose outputs are linear in their train-able parameters, such as radial basis function networks.

A popular second approach has been the development of alternative heuristic strategies for structure preservation, based on Sammon's algorithm or MDS. These schemes are reviewed, with comment, in Section 7.4.

[1] Of course, virtual memory systems permit much larger matrices but the time spent retrieving the data from disc is prohibitive and defeats the purpose of storing it in the first place.

7.2 Optimisation Schemes for the Sammon Mapping

Sammon's original algorithm adopted a Newton-Raphson-based steepest-gradient method for minimising the STRESS, which required the setting of a "magic factor" parameter, "determined empirically to be 0.3 or 0.4". This was a 'batch' algorithm which calculated the derivatives over the entire pattern set before adjusting any weights.

Mao and Jain [1995] used gradient-descent for their MLP approach, but updated the weights on a pattern-by-pattern basis. This is an attempt to alleviate the computational demands associated with large data sets. See section 3.5.1 for earlier comment on this particular approach.

In the NEUROSCALE model, in addition to the above two schemes, both a conjugate-gradient and the quasi-Newton BFGS technique [Press, Teukolsky, Vetterling, and Flannery 1992] have been evaluated. The relative efficiency of these optimisation strategies is illustrated in the figures below. Comparative STRESS against time plots are given for an MLP trained according to Mao and Jain [1995] and by conjugate-gradient and BFGS techniques. Also plotted are the latter two optimisation schemes applied to the RBF NEUROSCALE model.[2] In figure 7.1, the data set used was the 150-point SPHERES_3 data, and the initial point configuration was the principal component projection. In figure 7.2, equivalent plots are given for 300 data points from the same spherical distribution. The MLP network contained 12 hidden units (tanh) with linear outputs while the RBF comprised 40 Gaussian basis functions centred at random. These values were chosen to provide similar asymptotic STRESS minima. For the on-line training of the MLP, some considerable time was required in hand-optimising the learning rate and momentum term in order to obtain even adequate performance.

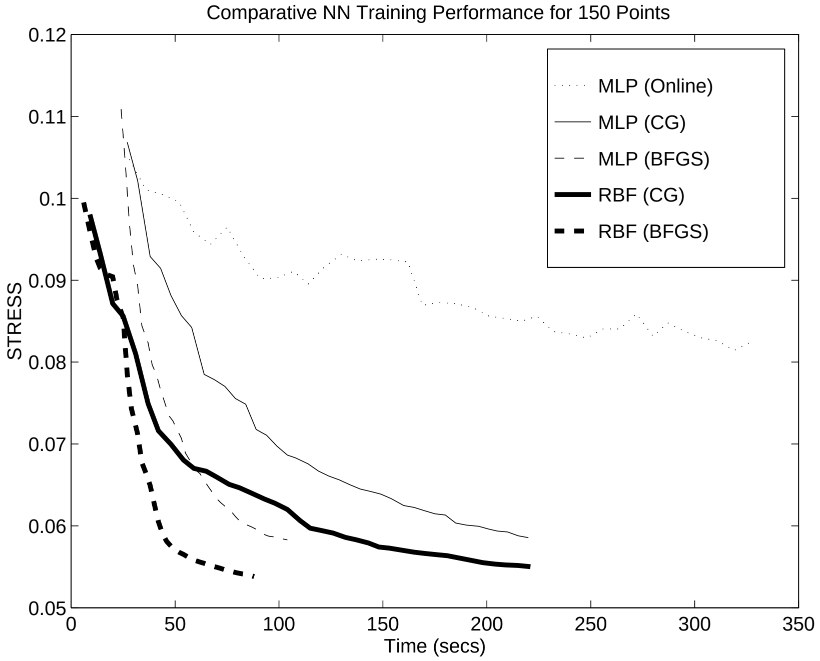


Figure 7.1: Comparison of training times for two models and three optimisation schemes for neural network implementations of the Sammon Mapping. The dataset is the 150-point SPHERES_3 set.

An example for some real, high-dimensional data is given in 7.3. The data comprise 217 points from the 1992 RAE database (see Chapter 4), and each is composed of 80 input variables. This precludes application of the BFGS technique in the case of the MLP, due to the large number of weight parameters resulting from this high input dimension. However, in the case of an RBF with fixed centres, this method may still be employed.

[2] The timings given were obtained on a 25-MFLOPS SUN Sparc-10/51 workstation.

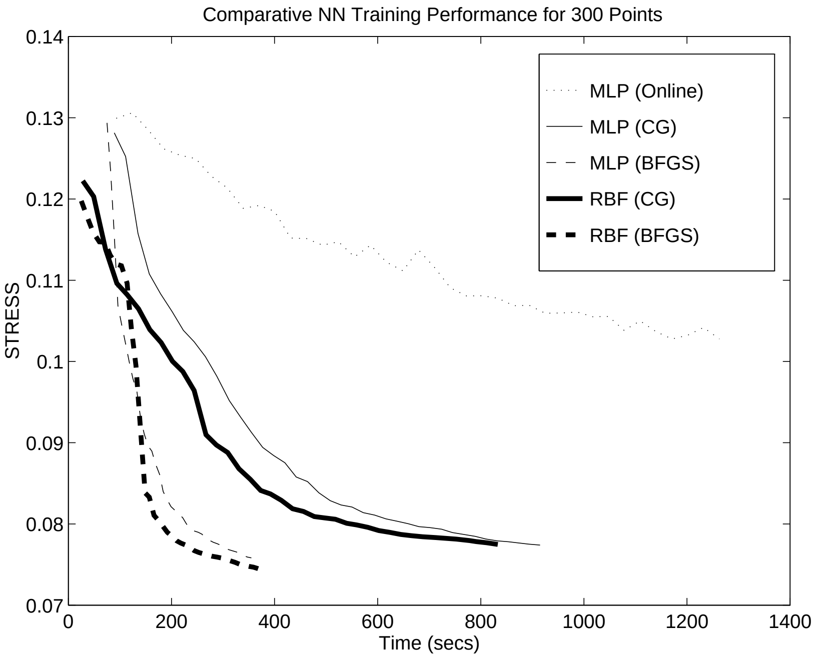


Figure 7.2: Comparison of training times for two models and three optimisation schemes for neural network implementations of the Sammon Mapping. The data is the 300-point SPHERES_3 set.

Webb [1995] utilised the *iterative majorisation* method for minimisation of the STRESS [Heiser 1991]. This technique exploits the Cauchy-Schwarz inequality in order to minimise an upper bound on the loss function. This still entails a double loop over the data points, but does not require any gradient calculations. However, previous studies by de Leeuw [1988], of convergence in MDS applications, concluded that the method "is reliable and very simple, but that it is generally slow, and sometimes intolerably slow."

Klein and Dubes [1989] applied the stochastic *simulated annealing* optimisation technique [Aarts and Korst 1989] to the Sammon mapping. The authors assessed the performance of the scheme on three data sets (real and synthetic) and their results corroborated earlier work by de Soete, Hubert, and Arabie [1987] — simulated annealing procedures generate good (low-STRESS) configurations but run times are very extended. Klein and Dubes thus concluded that the "computational cost [of simulated annealing] makes it impractical, especially for small problems."

This feature of exaggerated computational cost is often observed with the application of simulated annealing to many problem domains [Aarts and Korst 1989]. Simulated annealing is particularly appropriate for problems either characterised by many sub-optimal local minima or where the computation of the gradient of the cost function is either impossible or computationally prohibitive. In the case of a simplified structure-preserving mapping with cost function $\sum_i \sum_j (d^*_{ij} - d_{ij})^2$, the gradient may be calculated with little additional cost to that of calculating the STRESS. The number of floating-point operations required to calculate the gradient of the STRESS is approximately 22% greater than that of the STRESS alone, and some of the requisite distance calculations may be shared by both routines.

A statistical mechanics approach was also adopted by Hofmann and Buhmann [1995] for generating MDS configurations. They exploited mean-field and saddle-point techniques in order to derive an algorithm for approximation of the MDS problem based on the Expectation-Maximisation (EM) scheme. The authors indicated an interest in bench-marking this method against standard approaches, but have not yet done this [Hofmann 1995].

Again with respect to MDS, Tarazaga and Trosset [1993] considered the reformulation of the optimi-

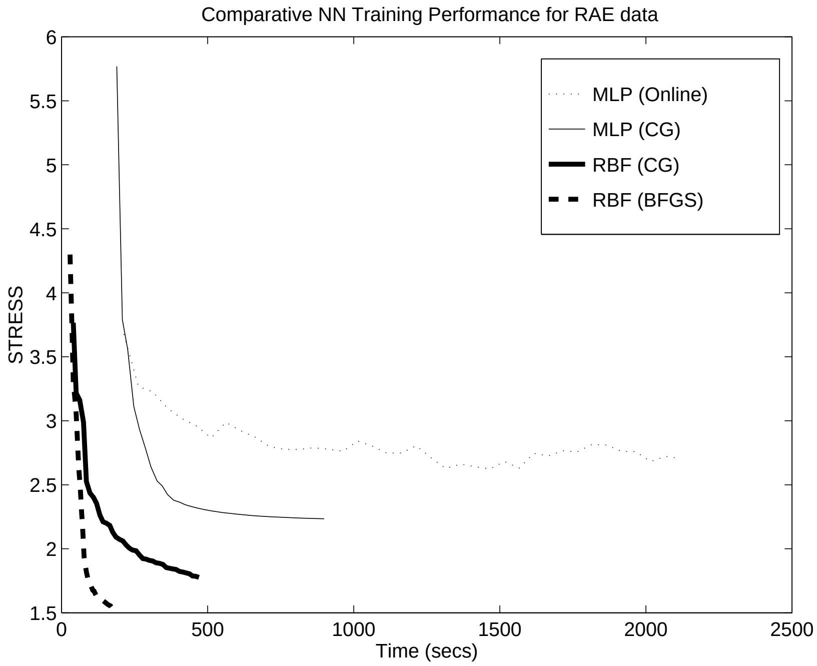


Figure 7.3: Comparison of training times for two models and three optimisation schemes for neural network implementations of the Sammon Mapping. The data comprises 217 points from the 1992 RAE database and has input dimension of 80.

sation problem in terms of functions of the matrix $\mathbf{B} = \mathbf{Y}\mathbf{Y}^{\mathrm{T}}$ (see Chapter 5 for the context of this). This is a non-trivial task, as it requires managing constraints on the rank of $\mathbf{B}$, and the authors do not propose any specific solutions.

7.3 An Improved Optimisation Algorithm for NEUROSCALE

Previous approaches to training neural networks to minimise STRESS-based measures have generally treated the task as one of standard nonlinear optimisation. Derivatives of the error with respect to the network weights were calculated and supplied to standard routines — gradient-descent, conjugate-gradient and BFGS for example.

This section details a new algorithm for the training of models that are linear in their weights, such as radial basis function neural networks. The advantage gained by using such networks in supervised problems with a sum-of-squares error function — that they could be trained via a single pass pseudo-inverse approach — is lost in the relative supervision case where the STRESS measure introduces quartic terms. The following algorithm effectively decomposes the training problem into a two-step procedure, one step of which is linear and can be computed efficiently.

7.3.1 The Algorithm

The equations for calculating the derivatives in the NEUROSCALE model are:

$$\frac{\partial E}{\partial w_k} = \sum_i^N \frac{\partial E}{\partial \mathbf{y}_i} \cdot \frac{\partial \mathbf{y}_i}{\partial w_k}, \quad \text{where} \tag{7.1}$$

$$\frac{\partial E}{\partial \mathbf{y}_i} = -2 \sum_{j \neq i} \left(\frac{d^*_{ij} - d_{ij}}{d_{ij}} \right) (\mathbf{y}_i - \mathbf{y}_j). \tag{7.2}$$

Equation (7.1) may be expressed in matrix form (for each separate output dimension) as:

$$\nabla E = \mathbf{J}^{\mathrm{T}} \delta, \tag{7.3}$$

as demonstrated in Section 6.2.2.

The set of linear equations implicit in equation (7.3) above, for the gradient of the STRESS measure with respect to the weights, is equivalent to the set of *normal equations* of a linear least-squares problem [Strang 1988]. In such a *supervised* problem, the error E is given by $E = \frac{1}{2} \sum_i \parallel \mathbf{y}_i - \mathbf{t}_i \parallel^2$, where the vectors $\mathbf{t}_i$ are the *targets*, such that

$$\frac{\partial E}{\partial \mathbf{y}_i} = (\mathbf{y}_i - \mathbf{t}_i). \tag{7.4}$$

In the relative supervision algorithm, there is an expression for $\partial E / \partial \mathbf{y}_i$ given by equation (7.2). This equation may be combined with equation (7.4) above to give a set of vectors that can be considered to represent *estimated targets* $\hat{\mathbf{t}}_i$:

$$\hat{\mathbf{t}}_i = \mathbf{y}_i - \frac{\partial E}{\partial \mathbf{y}_i}, \tag{7.5}$$

$$= \mathbf{y}_i + 2 \sum_{j \neq i} \left(\frac{d^*_{ij} - d_{ij}}{d_{ij}} \right) (\mathbf{y}_i - \mathbf{y}_j). \tag{7.6}$$

The vectors $\hat{\mathbf{t}}_i$ represent those *exact* targets for the network that would lead to an identical expression for the weight derivatives in the RBF, in the least-squares supervised case, as those obtained from the relative supervision approach. Of course, the problem cannot simply be solved in the one step as these targets $\hat{\mathbf{t}}_i$ are not fixed, but are dependent upon the current outputs of the network, $\mathbf{y}_i$, and thus the weights also.

For a fixed set of estimated targets $\hat{\mathbf{t}}_i$, the normal equations can be solved directly by

$$\mathbf{W} = \mathbf{J}^{+}\hat{\mathbf{T}}, \tag{7.7}$$

where $\hat{\mathbf{T}} = (\hat{\mathbf{t}}_1, \hat{\mathbf{t}}_2, \ldots, \hat{\mathbf{t}}_N)^{\mathrm{T}}$ and $\mathbf{J}^{+}$ denotes the *pseudo-inverse* [Golub and Kahan 1965] of the Jacobian.

An approach, therefore, to minimising E would be to repetitively estimate a set of targets and then for each successive set, solve the above least-squares problem directly. However, in the early stages of training when STRESS is high, the targets given by equation (7.5) are poor and often lead to an increase in STRESS. A more effective approach is to introduce a parameter η, which is initially small, and estimate the targets as

$$\hat{\mathbf{t}}_i = \mathbf{y}_i - \eta \frac{\partial E}{\partial \mathbf{y}_i}, \tag{7.8}$$

and increase η as STRESS decreases during the training procedure.

The training algorithm as implemented in NEUROSCALE then becomes:

❶ Initialise weights **W** to small random values.

❷ Initialise η to some small value.

❸ Calculate $\mathbf{J}^{+}$, where $\mathbf{J}_{ik} = \phi_k(\mathbf{x}_i)$.

❹ Calculate estimated targets $\hat{\mathbf{t}}_i$ from equation 7.8.

❺ Solve for weights using $\mathbf{W} = \mathbf{J}^{+}\hat{\mathbf{T}}$.

❻ Calculate STRESS.

❼
- If STRESS has increased, $\eta = \eta \times k_{down}$ where $0 < k_{down} < 1$ is a constant.
- If STRESS has decreased, $\eta = \eta \times k_{up}$ where $k_{up} > 1$ is also a constant.

❽ If not converged, return to Step ❹.

Note that, for fixed basis functions, the potentially computationally expensive pseudo-inverse calculation need only be performed once and the $(h \times N)$ matrix $\mathbf{J}^{+}$ stored. Appropriate values for k_{down} and k_{up} are, for example, 0.1 and 1.2 respectively. These values appeared acceptably robust for various data sets.

7.3.2 Convergence Behaviour

It can be shown that this new scheme converges on a minimum of E.

At time step t let

$$\mathbf{J}\mathbf{W}^t = \mathbf{Y}^t, \quad \text{and} \tag{7.9}$$

$$\hat{\mathbf{T}}^t = \mathbf{Y}^t - \eta \left.\frac{\partial E}{\partial \mathbf{Y}}\right|_{\mathbf{Y}=\mathbf{Y}^t}, \tag{7.10}$$

$$= \mathbf{Y}^t - \eta \boldsymbol{\Delta}^t, \quad \text{letting} \quad \boldsymbol{\Delta}^t = \left.\frac{\partial E}{\partial \mathbf{Y}}\right|_{\mathbf{Y}=\mathbf{Y}^t}. \tag{7.11}$$

Then,

$$\begin{aligned}
\mathbf{Y}^{t+1} &= \mathbf{J}\mathbf{W}^{t+1}, && (7.12)\\
&= \mathbf{J}(\mathbf{J}^{+}\hat{\mathbf{T}}^{t}), && (7.13)\\
&= \mathbf{J}\mathbf{J}^{+}(\mathbf{Y}^{t} - \eta\boldsymbol{\Delta}^{t}), && (7.14)\\
&= \mathbf{Y}^{t} - \eta\mathbf{J}\mathbf{J}^{+}\boldsymbol{\Delta}^{t}, && (7.15)
\end{aligned}$$

since $\mathbf{J}\mathbf{J}^{+}\mathbf{Y}^{t} = \mathbf{Y}^{t}$ from equation (7.9). (Note that in general, $\mathbf{J}\mathbf{J}^{+} \neq \mathbf{I}$.)

Now, $\boldsymbol{\Delta}^{t}$ is the direction of steepest descent, in $\mathbf{Y}$-space, on the STRESS surface. For the proposed scheme to minimise STRESS, it is required that the change in weights, and thus in $\mathbf{Y}$, implied by equation (7.15) represents a *descent direction* upon the cost surface. For this to be the case, the inner-product of the term $\mathbf{J}\mathbf{J}^{+}\boldsymbol{\Delta}^{t}$ and the direction of steepest descent $\boldsymbol{\Delta}^{t}$ must be *positive*. This inner-product is given by

$$\begin{aligned}
\mathrm{vec}[\boldsymbol{\Delta}^{t}]^{\mathrm{T}}\mathrm{vec}[\mathbf{J}\mathbf{J}^{+}\boldsymbol{\Delta}^{t}] &= \mathrm{tr}\left[(\boldsymbol{\Delta}^{t})^{\mathrm{T}}(\mathbf{J}\mathbf{J}^{+}\boldsymbol{\Delta}^{t})\right], && (7.16)\\
&= \sum_{l=1}^{q}(\delta_l^t)^{\mathrm{T}}(\mathbf{J}\mathbf{J}^{+})\delta_l^t, && (7.17)
\end{aligned}$$

where δ_l^t are the q component column vectors of the matrix $\boldsymbol{\Delta}^{t}$.

Since by considering its singular value decomposition as $\mathbf{J} = \mathbf{USV}^{\mathrm{T}}$, $(\mathbf{J}\mathbf{J}^{+})$ is equal to $\mathbf{UU}^{\mathrm{T}}$ and is thus positive semi-definite. The inner-product will then be non-negative for all $\boldsymbol{\Delta}^{t}$. Indeed, it will be positive excepting when the subspace of $\boldsymbol{\Delta}^{t}$ lies in the null-space of $(\mathbf{J}\mathbf{J}^{+})$, which is identical to the null-space of $\mathbf{J}\mathbf{J}^{\mathrm{T}}$. Thus the local minima of the cost surface descended by the shadow-targets algorithm are identical to those of E.

7.3.3 Interpretation

Firstly, consider equation (7.8). The estimated target $\hat{\mathbf{t}}_i$ is identical to the new point obtained after one step of a gradient-descent optimisation of the standard Sammon Mapping. In this sense, the proposed algorithm is effectively *shadowing*, step by step, the standard Sammon Mapping generation procedure. For this reason, the algorithm will be subsequently referred to as the *shadow-targets* algorithm.

However, in contrast to the Sammon mapping, by reference to equation (7.15), the combination of reduced basis functions and smoothing parameter(s) in the RBF network serves to transform this standard optimisation step $\boldsymbol{\Delta}$ to an alternative step, $(\mathbf{J}\mathbf{J}^{+})\boldsymbol{\Delta}$. The matrix $\mathbf{J}\mathbf{J}^{+}$ is effectively a projection matrix (it is clearly idempotent) that projects the columns of the error matrix $\boldsymbol{\Delta}$ onto the subspace spanned by the columns of $\mathbf{J}$. If there is a full complement of basis functions of sufficiently small σ (such that $\mathbf{J}$ is not ill-conditioned), then $\mathbf{J}\mathbf{J}^{+} = \mathbf{I}$, and the algorithm effectively defaults to a repetitive *a posteriori* fitting of an evolving Sammon map. Typically, however, either there will be fewer basis functions than patterns or, as demonstrated in Section 6.6.3, the width parameter σ of the basis functions will be chosen such that $\mathbf{J}\mathbf{J}^{+}$ is singular.

7.3.4 Performance Comparison

Speed of Training

Figure 7.4 illustrates typical plots of the evolution of STRESS during training for an RBF model trained on the 150-point SPHERES_3 data set. The RBF comprised 40 Gaussian basis functions ($\sigma = 2.0$), and training performance is shown for the best standard non-linear optimisation technique, BFGS, and for shadow-targets.

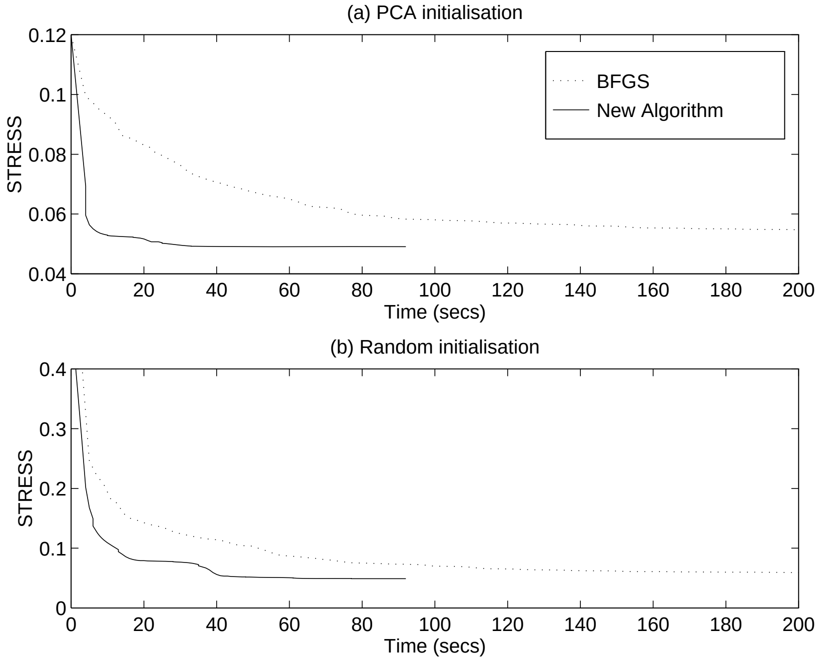


Figure 7.4: Evolution of STRESS during training for both BFGS optimisation, and shadow-targets, on the SPHERES_3 dataset. The upper plot (a) is for PCA initialisation, its partner (b) is for random initialisation.

For the high input dimension data set RAE_PCB, the illustration of STRESS minimisation is given in figure 7.5. In this example, a plot is only given for PCA initialisation. There were 70 basis functions in the network of type $r^2\log(r)$.

Local Minima

In terms of the algorithm's susceptibility to entrapment in sub-optimal local minima, an identical experiment to that given in figure 6.3 is given in figure 7.6. For the IRIS_45 dataset, 1000 mappings were generated with the shadow-targets algorithm, and a histogram of the final minimum STRESS values plotted.

Because the shadow-targets algorithm follows the Sammon mapping, it is unsurprising to observe that in figure 7.6, the minima in the plot correspond to those of the Sammon Mapping, the level-1 minima. In comparison to the standard BFGS optimisation technique, the algorithm exhibits a significant number of sub-optimal minima, although because of the apparent absence of level-2 minima, the lowest STRESS obtained is now 0.0028, and on nearly 50% of the runs. This represents a very considerable improvement when compared directly to the Sammon Mapping histogram of figure 6.1, and is as a result of the perturbing of the trajectory in **Y** space by the $(\mathbf{JJ}^{+})$ matrix, with its associated smoothing effect.

It was shown in Section 7.3.2 that the shadow-targets algorithm was susceptible to the same level-2 minima as other minimisation methods. However, as noted in the above paragraph, figure 7.6 indicates that the algorithm is not trapped by these minima, and the explanation given in Section 6.2.2 concerning invariance of STRESS under rotation and translation once again applies. The major distinction between the shadow-targets algorithm and those employed previously is that the effect of perturbing solutions within the principal eigenspace of the network Jacobian **J** is no longer evident. More direct evidence for this is given in figure 7.7, which plots the evolution of STRESS for two identical networks from identical starting configurations. One of the networks is trained by simple gradient-descent, and one by shadow-targets. In both cases, the learning rate η was identical and constant during training.

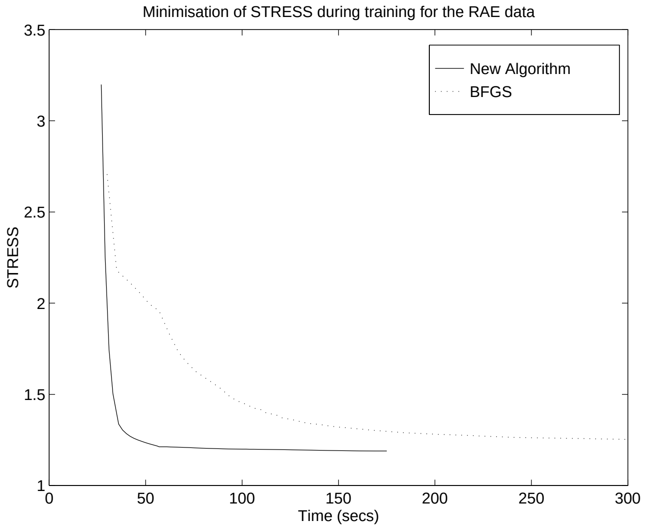


Figure 7.5: Evolution of STRESS during training for both BFGS optimisation, and shadow-targets, on the RAE_PCB dataset.

The gradient-descent trained network exhibits the behaviour illustrated in figure 6.4 earlier, where training is effectively terminated as the error vectors $\boldsymbol{\delta}$ are in the direction of the minor eigenvectors of **J**. That the gradient-descent scheme initially exhibits faster convergence than its shadow-targets counterpart is also consistent with the behaviour illustrated in that earlier figure. The trajectory of a single output dimension of the configuration for gradient-descent is defined by

$$\Delta \mathbf{z}_{gd} = -\eta \mathbf{J}\mathbf{J}^{\mathrm{T}}\boldsymbol{\delta} = -\eta \sum_{k} \lambda_k^2 \beta_k \mathbf{u}_k, \tag{7.18}$$

as shown in Section 6.2.2. For shadow-targets, this is

$$\Delta \mathbf{z}_{st} = -\eta \mathbf{J}\mathbf{J}^{+}\boldsymbol{\delta} = -\eta \sum_{k:\lambda_k \neq 0} \beta_k \mathbf{u}_k, \tag{7.19}$$

and the trajectory is no longer perturbed towards the direction of the principal eigenvectors of $\mathbf{J}\mathbf{J}^{\mathrm{T}}$.

Curvature and Generalisation

As the shadow-targets algorithm effectively repetitively fits a Sammon mapping, and does not seek a solution that implicitly reduces $\| \mathbf{W} \|^2$, it might be expected that values of curvature during training should be significantly greater than for the BFGS training method. By contrast with figure 6.8, figure 7.8 illustrates the curvature of the RBF transformation during shadow-targets training on the IRIS_45 data set. For comparison, the equivalent plot for BFGS training, in figure 6.8, is also plotted. The RBFs in both experiments had 45 Gaussian basis functions with width parameter $\sigma = 1.0$, and the results confirm the hypothesis that curvature is not significantly reduced. (Some reduction in curvature may often be observed if the initial weights tend to be too large.)

The plot of the evolution of STRESS and curvature for the new algorithm is superimposed on that for BFGS and *a posteriori* fitted RBFs and shown in figure 7.9. This illustrates that the curvature of a network, trained by shadow-targets to a given value of STRESS, is in most cases lower than that of an identical network, supervisorially trained to produce a standard Sammon configuration, although not to the same extent as the BFGS-trained version.

An indication of the generalisation performance that may be obtained from networks trained by the shadow-targets algorithm is given in figure 7.10. This illustrates, in a similar form to figure 6.13 on

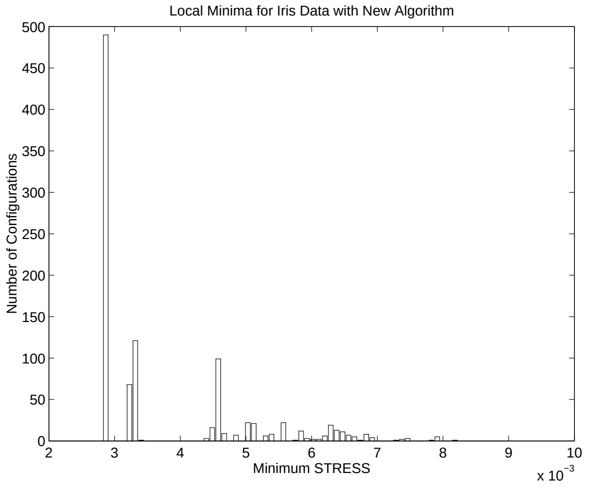


Figure 7.6: Histogram of number of final configurations with corresponding STRESS's for 1000 runs of NEUROSCALE, with $\alpha = 0$, on the IRIS_45 dataset. The width of the 45 Gaussian basis functions was 3.0. The NEUROSCALE model was trained with shadow-targets.

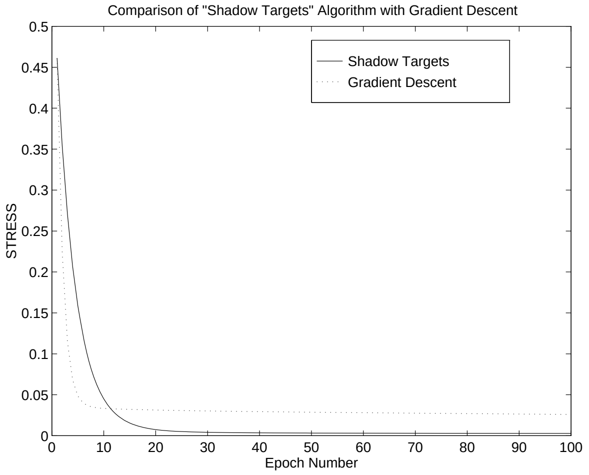


Figure 7.7: Evolution of STRESS for identically initialised networks for gradient descent and shadow-targets training.

page 99 of the previous chapter, the test STRESS that would be obtained by the network at each step of the training algorithm. This is shown for three separate networks with values of σ of 1,10 and 30, and the equivalent plots for BFGS (where $\sigma = 1$) and *a posteriori* networks are superimposed.

From figure 7.10, it is evident that the minimum training STRESS with shadow-targets is superior to that of BFGS in all cases. However, for equivalent values of basis function width parameter ($\sigma = 1$), test STRESS is lower for the relatively supervised version. This is a result of the implicit weight decay in the latter algorithm, and is consistent with the curvature plot of the previous figure. However, it can be seen that for a higher value of width parameter, 30, both training and test STRESS are much lower with shadow-targets. A network trained by BFGS with $\sigma = 30$ would exhibit very high STRESS due to the exceedingly poor conditioning of the matrix **J**. The plot for shadow-targets with $\sigma = 10$ reflects the unusual behaviour demonstrated in figure 6.22 in the previous chapter, where it was observed that generalisation performance was relatively poor for certain intermediate values of basis function width.

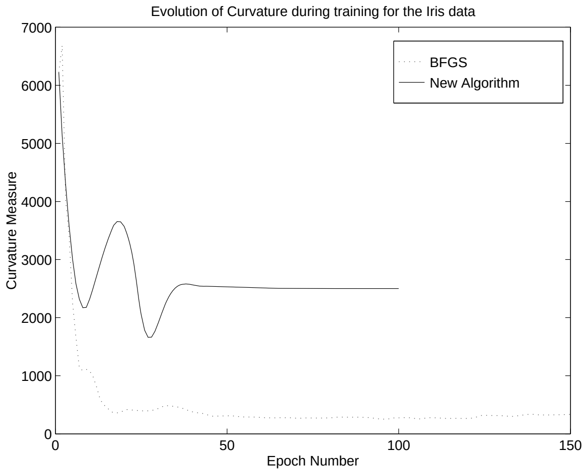


Figure 7.8: Curvature against time during the shadow-targets training of a NEUROSCALE mapping on the IRIS_45 data.

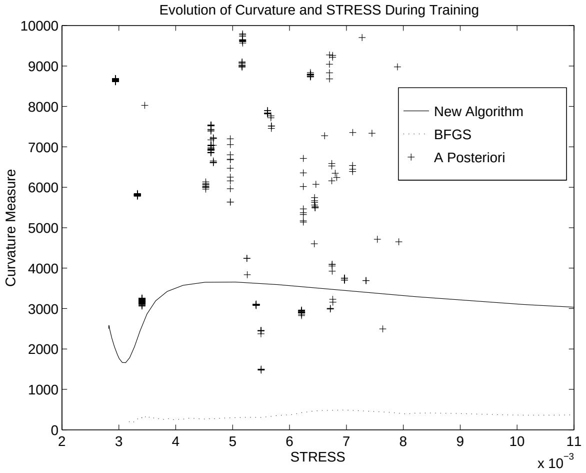


Figure 7.9: Curvature of a posteriori fitted RBF networks compared to that during training of a NEUROSCALE model by BFGS and shadow-targets methods.

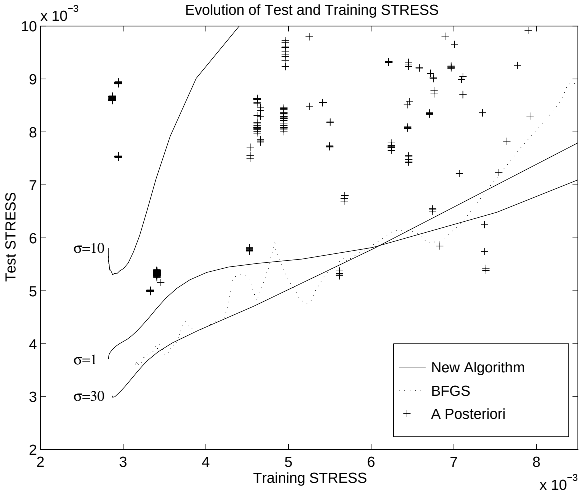


Figure 7.10: Test STRESS of a posteriori fitted RBF networks compared to that during training of a NEUROSCALE model by BFGS and shadow-targets methods.

7.4 Alternative Mapping Strategies

7.4.1 Review

Sammon [1969] himself remarked upon the undesirable scaling properties of his projection method and suggested that some sort of data compression pre-processing might be appropriate. Specifically, he utilised a clustering algorithm to determine a set (maximum ≈ 250) of data prototypes which may then be mapped using the standard algorithm. This approach was further refined by Pykett [1978], who mapped class centroids and superimposed circles on the map as an approximate indicator of class dispersion.

Chang and Lee [1973] proposed two alternative strategies for mapping large numbers of pattern vectors. The first, the "relaxation method" is really an optimisation technique and is designed to minimise the contribution to the overall STRESS of individual pairs of points. It is possible to directly perturb a pair of points $\mathbf{y}_i$ and $\mathbf{y}_j$ such that $\| \mathbf{y}_i - \mathbf{y}_j \| = d^*_{ij}$. This will naturally disrupt all other optimised distances, and Chang and Lee proposed reducing this disruption by introducing a term into the algorithm that attenuates the perturbation for increasing distance, and so making the mapping more local. Iterated application of this algorithm should then produce a stable solution. In more extensive analysis, however, Siedlecki, Siedlecka, and Sklansky [1988] observed the algorithm to be often unstable and to possess poor convergence properties.

Chang and Lee's second strategy was the "frame method". In this mapping, a subset of M of the original N points are selected and mapped according to Sammon's algorithm to generate a fixed *frame*. All the remaining $N - M$ points are then mapped, but are only constrained by the inter-point distances to the M points in the frame, as distinct from the entire set of N data points. This is equivalent to the mapping of a $N \times M$ distance matrix, and so clearly saves on storage and computational requirements. (However, there is no reason given for the fixing of the M points first, in preference to simply scaling the entire $N \times M$ distance matrix in one go.) The quality of this approximated mapping obviously depends on both the characteristics of the data and on the M points in the frame. An example is given of the frame method on a simple data set, but Siedlecki et al. [1988] question the evidence for the value of this approximate mapping technique. However, in a paper on the representation of rigid structures with minimal 'connected-ness' information, Levine and Kreifeldt [1994] discuss the potential of extending their work to the generation of MDS configurations from a partial set of distances and indicate future research in this area.

Another approximation approach was the *triangulation* mapping of Lee, Slagle, and Blum [1977]. This is based upon the fact that any three points may be placed in a plane such that all three inter-point distances are preserved exactly. Thus in a single pass of the dataset, $2N - 3$ distances (of the total $N(N-1)/2$) may be retained by locating each point sequentially such that the distances to two previously projected points are retained. There remains the choice of which of these distances to select for preservation — the authors propose that a good criterion for selection is that the *minimal spanning tree (MST)* of the data be maintained. (The minimal spanning tree [Gower and Ross 1969] is the tree connecting all points, with no loops, that has the smallest span, defined by the sum of the inter-point distances along the tree.) This entails the mapping of each point from the MST (in a *breadth-first* manner) so that the distance to its nearest neighbour is fixed, plus one additional distance which may either be selected as the second nearest-neighbour or some alternative reference point. This choice thus allows the emphasis of the mapping to be either local or global, or indeed, offers the user a choice of "perspective". Siedlecki et al. [1988] illustrate how this selection can affect the clarity of clusters in the final projection. Note that the construction of this mapping will be partially ambiguous as there are two alternative solutions for the placing of any point in the plane with respect to any other two points.

A modified form of the Sammon STRESS was adopted by Niemann and Weiss [1979]. The criterion to be minimised was of the form $\sum_{ij}(s^*_{ij})^r(s^*_{ij} - s_{ij})^2$, where s^*_{ij} and s_{ij} were the *squared* Euclidean distances in the data and map space respectively. Choice of parameter r allowed greater emphasis to be placed on local or global structure. The authors also utilised a co-ordinate descent technique, which involved adjusting a single parameter in the map at each step, by a fixed amount which was determined by the solution of a cubic equation. Siedlecki et al. [1988] observed this method to have very good convergence properties, but that it generated considerable distortion.

The frame method and the triangulation mapping were combined by Biswas, Jain, and Dubes [1981], to effect the *Sammon-Triangulation* technique. They proposed the generation of an M-point frame first, followed by the mapping of all other points, via the triangulation method, to retain the distances to the two nearest-neighbours from amongst the points within the frame. An additional step in the procedure is required to ensure that the triangle equality is maintained for the projected points. The authors claim that the frame method "leads to projections that greatly distort the data" while the Sammon-Triangulation scheme "produces better results than the triangulation method because more global information about the data set is retained". For a large number of patterns, Biswas et al. reiterate the suggestion of mapping only a prototypical subset of the patterns, or of performing some *a priori* clustering.

The approach to reducing the storage and computational cost of Kakusho and Mizoguchi [1983] is to only constrain the position of each point by the distances to the k nearest-neighbours (in data space). This is another form of a localised Sammon mapping, and exploits a basic idea from NMDS dating back to Shepard and Carroll [1966]. A similar concept is entertained by Schwartz, Shaw, and Wolfson [1989] for the mapping of the convoluted 3-dimensional primate visual cortex. The mapping of a given point is again restricted to a local "patch", with the remark that the distance matrix then becomes sparse and can be maintained as a collection of binary search trees. The patches were empirically chosen to represent around 10% of the total data area.

7.4.2 Comment

Apart from both the relaxation method of Chang and Lee [1973] and the co-ordinate descent technique of Niemann and Weiss [1979], all the other approaches discussed above are effectively *ad hoc* methods for *subset* mapping. Furthermore, the majority of these schemes are based on retention of *local* distances [Chang and Lee 1973; Lee, Slagle, and Blum 1977; Biswas, Jain, and Dubes 1981; Kakusho and Mizoguchi 1983; Schwartz, Shaw, and Wolfson 1989].

It has already been emphasised that in general visualisation applications, global structure can be highly informative — for example, the ordering of research ratings in the RAE data from Chapter 4. While local mappings seek to maintain clusterings under the dimension-reduction process, the inter-cluster structure is often equally, and sometimes more, important. Indeed, too much emphasis on local spatial information can lead to confusion of clusters, a phenomenon observed by Siedlecki et al. [1988]. In such cases, the original approach suggested by Sammon [1969], to select cluster prototypes for mapping, is actually the most sensible.

The requisite data pre-processing stage that this implies only transfers the scaling problem to one of selecting either the best set of prototypes or the best subset of the distance matrix, and this is a non-trivial question which is also likely to be computationally demanding. Of these two alternative strategies, selecting M_1 prototype points for individual mapping requires scaling a $(M_1 \times M_1)$ matrix, while selecting M_2 points as a basis for scaling a subset of the distance matrix requires scaling a $(N \times M_2)$ matrix. Clearly, for similar computation, more prototypes can be selected in the first method than in the second — $M_1 \gg M_2$ for large N. However, the second approach does permit the mapping of the entire set of data points, albeit constrained by only a partial set of distances.

The use of a parameterised transformation, as exploited in NEUROSCALE, is an advantage when scaling a subset of data points. Because it is now possible to project new data, all the remaining data points may still be mapped by the transformation once it is defined. The network is effectively training on the subset alone, and then generalising to the remaining data. Clearly, the subset of points must be representative of the distribution of the data as a whole if a good projection is to be expected.

One particular example approach might be to determine a set of *K* cluster centres, via the *K*-means procedure [MacQueen 1967], and then scale these with the standard NEUROSCALE algorithm. Subsequently, the entire dataset may be projected. Strictly, the STRESS calculation for the *K* means should be adjusted to account for the number of data points that each mean represents.

The above algorithm is only one of many potential approaches for coping with the demands of large datasets. Overall, this suggests, as intimated by Levine and Kreifeldt [1994], that the key issue is that of choosing the subset of distances that, when scaled to generate the mapping, best approximates the mapping obtained by scaling the complete distance matrix. This is clearly a very complex question, and one that demands considerably more study than there is opportunity to include in the remainder of this thesis. As such, it must remain an important area for future research.

7.5 Conclusions

The comparison of standard nonlinear optimisation procedures for both MLPs and RBFs revealed that the quasi-Newton BFGS technique exhibited best performance in terms of minimising training time. Furthermore, RBFs were better in this respect than MLPs, particularly where the input dimension was high and the number of weights in the MLP precluded the use of BFGS. On-line training of an MLP was significantly worse than all other optimisation approaches.

However, a new 'shadow-targets' training algorithm proposed in Section 7.3 for networks with outputs linear in the weights, such as RBFs, proved to be significantly better, in terms of speed of convergence, than even BFGS, attaining the region of the local minimum an order of magnitude more quickly. This algorithm was also shown to converge to the same classes of minima as the standard methods, and still exhibit generally lower curvature and test STRESS than similar *a posteriori* mappings, due to its inherent smoothing effect. However, for equivalent basis function smoothing parameter σ, networks trained with the shadow-targets algorithm exhibited greater curvature than those employing standard nonlinear optimisation techniques, as the former does not incorporate the implicit weight-decay of the latter methods in the relative supervision context.

Importantly, however, when employing the shadow-targets algorithm it is possible to use larger values of σ without suffering from the increased local minima effects discussed in Section 6.2.2 in the previous chapter. This enables the efficient generation of low-STRESS training *and* test transformations, which are superior to the standard nonlinear-optimised versions. In this respect, the method of choice for the training of a topographic transformation would be to utilise the shadow-targets algorithm with an appropriately large σ. This does necessitate, however, the determination of a suitable value for the basis function width parameter, although from the results of Section 6.6.3, this choice may be guided by the heuristic that "bigger is better".

It should be noted that the shadow-targets training algorithm is equally appropriate to mappings which incorporate some subjective element (i.e. $\alpha > 0$) and can be expected to offer significant reductions in the required training time for such applications. However, the argument that the transformation should be maximally smooth no longer applies and this can be expected to complicate the choice of σ, which is now likely to be significantly data-dependent. Further investigation would be appropriate in this direction.

Finally, in Section 7.4, some of the alternative approaches designed to ameliorate the computational problems associated with the mapping of large datasets were reviewed. These may be seen as generally *ad hoc* approaches to clustering or subset selection, and as such pose equally non-trivial problems which, for the purposes of topographic mappings, is still an open research issue.

Chapter 8

Conclusions

8.1 Overview

The introductory chapter of this thesis began by considering the generic problem of *information processing* and more specifically, how dimension-reduction techniques could ease the interpretation of complex data sets. Building on this theme, the objective of subsequent chapters has been two-fold — firstly to motivate the development of a particular neural network approach to dimension-reducing topographic mapping in the information-processing context, and secondly, to improve the theoretical understanding of the design, training and application of such models.

8.2 Why NEUROSCALE?

In Chapter 2 it was reasoned that in data-analytic contexts, the Sammon Mapping was the most effective strategy for topographic dimension reduction. This is an important conclusion in itself, considering the widespread and often inappropriate application of the Kohonen self-organising feature map. Established theoretical properties and experimental evidence, both in existing literature and presented within this thesis, combine to support this argument. The main advantage of Kohonen's approach is computational, because realistically, application of the Sammon Mapping is restricted to fewer than 1000 data points. However, the computational tractability of Kohonen's algorithm should not be allowed to disguise its flaws as an information processing paradigm.

The feed-forward neural network topographic mapping technique introduced in Chapter 3, NEUROSCALE, was thus based upon the Sammon Mapping and utilises a radial basis function neural network. Because of this neural network element, it offers the capability of generalisation to new data — a feature absent from Sammon's original algorithm which is effectively a look-up table approach.

An important extension embodied in NEUROSCALE is the capacity to exploit additional information in the mapping process. In standard approaches to topographic mapping, the geometry of the output space is determined solely according to some conventional metric (generally Euclidean) defined over the data space. If alternative information is available — such as class labels — then this may be allowed to influence the mapping (in order to emphasise clustering, for example). Previous implementations of this concept have been largely heuristic, whereas within NEUROSCALE, the extra information is embodied as an *additional metric*. This dual metric approach then allows the two classes of information to be combined variably, and importantly, in a *consistent* manner. At one extreme is the purely unsupervised mapping, based exclusively on the geometry of the data in the input space, while at the other extreme is a supervised variant based exclusively on the additional metric, which can be considered to represent the 'preference' of the mapping process. As a result of this mechanism, enhanced visualisation spaces may be derived which can be considerably more informative than those extracted utilising conventional topographic criteria.

That there is genuine merit in the generation of such hybrid supervised / unsupervised feature spaces was demonstrated in Chapter 4, which comprised a study of data taken from the 1992 Research Assessment Exercise. When compared with other established feature extraction approaches on this difficult high-dimensional real-world dataset, the NEUROSCALE mapping that included an element of class information gave the most useful visualisation. Instead of the confusion of the five classes of interest, the use of a metric which preferred a linear ordering permitted considerably more structure to be elucidated. However, because the mapping still contained a significant geometric element, it was possible to discover apparent inconsistencies in the awarding of research ratings from the NEUROSCALE projection of the training data — inconsistencies that were supported by independent classification experiments. The supervised component of the mapping also implied that the generalisation projection to previously unseen data offered better potential for subsequent prediction. Indeed, a classifier based on this feature space was able to correctly predict the rating awarded to 70% of the test set, which was superior to other illustrative classification examples.

On the basis of the results presented in Chapter 4 and elsewhere in the first half of this thesis, NEUROSCALE offers considerable potential as a tool for the visualisation and exploratory analysis of data. That this proposed topographic model utilises a feed-forward neural network naturally raises certain specific questions concerning its application; questions which relate closely to neural networks employed in more conventional rôles. Of particular relevance is the need to determine effective training techniques, appropriate model complexity and values of other model parameters, and these issues were considered in the second half of the thesis.

8.3 Theoretical Issues

While the presented theoretical analysis should apply to general topographic RBF transformations, all experiments utilised *Gaussian* radial basis functions. This is a reflection of the ease of differentiability of that function, rather than any property specific to its use in the NEUROSCALE model. Also, the emphasis of the final chapters was on exclusively objective mappings, although some results are applicable to the case where subjective information is incorporated and these are indicated in the text.

Local minima. In terms of training the NEUROSCALE model, it is already well known that Sammon's STRESS measure exhibits many sub-optimal local minima for even the simplest of problems. Further evidence of this effect was presented in Chapter 6, where it was seen that NEUROSCALE models with significant values of basis function width 'σ' were seemingly unaffected by these minima. In fact, it was shown that the local minima found by Sammon's algorithm were generally *unsmooth* transformations of the input data and unrealisable by the RBF within the NEUROSCALE architecture. This result is very significant as the proliferation of sub-optimal local minima is a major practical disadvantage of Sammon's technique.

However, by way of balance, NEUROSCALE was seen to introduce a new class of minima as a result of those very smoothness properties of the network. When adopting a standard nonlinear optimisation approach to network training, the attained final STRESS minimum occurred at some higher value than the best obtainable by the Sammon Mapping (although, in that latter approach, many runs might have been necessary to find that particular minimum). This effect is directly related to the smoothness of the RBF transformation, and as $\sigma \to 0$, it vanishes and NEUROSCALE tends to approximate the Sammon mapping more directly and so becomes susceptible to all its associated local minima also.

This reveals a clear trade-off. Large σ gives few local minima problems, but the STRESS of the ultimate solution is inferior to the best obtained by the Sammon Mapping. As σ decreases, this optimal STRESS value improves, but additional sub-optimal minima as exhibited by the Sammon Mapping are then introduced.

Model complexity. A significant feature of the NEUROSCALE model is its facility for generalising to previously unseen data. When applying neural network models in other domains, it is known that over-complex models generalise poorly in the presence of noise, as they tend to over-fit the data. By contrast, in Chapter 6, it was seen that test STRESS of NEUROSCALE models was relatively insensitive to the number of basis functions within the network — even to the point where as many basis functions as data points were utilised. There are two key underlying points which combine to explain this counter-intuitive behaviour. Firstly, it is possible to derive significant insight into the necessary form of the network function independent of the data. Secondly, there is a regularising component automatically incorporated in the NEUROSCALE training algorithm which, given this knowledge of the desired functional form, proves to be highly appropriate.

Taking the first point, for topographic mappings it was reasoned that for good generalisation the transformation effected by the neural network (or indeed, any functional model) should ideally be smooth (have zero second- or higher-order derivative terms) and an expression was determined for the necessary gradient of the network function. To underline this, experimental evidence was presented illustrating that for networks exhibiting identical training STRESS on a data set, those with higher curvature (second-order terms) gave correspondingly higher errors on unseen test data. Most compelling was an example where networks of differing curvature, producing apparently identical configurations from a training set, exhibited considerable difference in error on an identical test set. The key to this seemingly inconsistent behaviour is to understand that there are many configurations of points that give rise to identical measures of STRESS, and these simply correspond to arbitrary rotations and / or translations of any one particular configuration. Furthermore, the smoothness of any particular transformation depends, nonlinearly, on this rotation and translation.

This is a highly relevant observation, as one existing approach to producing a transformational topographic mapping is to generate a configuration using Sammon's standard procedure, and then fit a parameterised model to the resulting solution *a posteriori*. However, the smoothness, and therefore the quality of generalisation that might be expected, of such a network is generally arbitrary. By contrast, NEUROSCALE networks exhibited lower curvature and test STRESS than these *a posteriori* models in *all* examples. An important result therefore of Chapter 6 was to explain this behaviour in terms of the learning dynamics of the relative supervision algorithm. It was seen that NEUROSCALE models trained in this way automatically tend to generate output configurations that reduce the sum-of-squared weight values within the network. This effective *weight decay* or *self-regularisation* explains why NEUROSCALE models had appeared largely insensitive to their complexity and were observed to generalise better that identical *a posteriori*-trained networks. While the extent of this implicit regularisation is not explicitly controllable (although it increases with σ), it is important to underline that its effect is not to reduce the squared-weights at the expense of increasing the training error, but rather to minimise training error while at the same time seeking a rotation and translation of the final output configuration with associated lower values of the weights.

Training Algorithms. This feature of the NEUROSCALE training algorithm is highly effective, and explains the apparent ease with which test projections with low STRESS were obtained throughout the thesis. However, this regularising effect is a function of the smoothness of the network and as already observed, there is a penalty to be paid in having too large a value of σ as the minimum obtainable STRESS on the training set deteriorates. In Chapter 7, a new training method was presented — the "shadow-targets" algorithm. Its name refers to the fact that the algorithm effectively shadows Sammon's mapping algorithm, but with the smoothness of the RBF incorporated at each step. An unfortunate consequence of minimising a STRESS measure within NEUROSCALE is that the benefit of linearity in the RBF model is lost as the error measure introduces quartic terms. The shadow-targets algorithm effectively decomposes the training problem into two steps — one of which is linear. Because of this approach, the new algorithm was an order of magnitude more efficient at reaching a STRESS minimum, and was applicable to both unsupervised and supervised mappings, although it did not incorporate the implicit weight decay of previous methods. However, the algorithm does not suffer from the increased minimum problems also associated with those techniques and is therefore able to reproduce the best observed minimum generated by the Sammon Mapping. While, for equivalent values of σ, generalisation was seen to be poorer than that illustrated for models trained using relative supervision, this new approach allows large values of σ to be chosen. Such values of basis function width produce smoother transformations, and it was relatively simple to find a value of σ which gave lower training and test STRESS than that of previous approaches. From the evidence of this thesis, the shadow-targets training algorithm should therefore be the best method for training a topographic feed-forward neural network, given appropriate choice of σ.

8.4 Directions for Future Research

Applications. As underlined by Friedman [1994], many computational learning methods can be shown to be optimal under the appropriate circumstances. Therefore, while the results on the RAE data in Chapter 4 showed considerable promise for the NEUROSCALE approach, there is always scope for application to further datasets, particularly those with alternative forms of subjective information.

Relevance of theoretical results to subjective mappings. Much of the insight derived into the issue of model complexity in Chapter 6 was on the basis of purely objective, $\alpha = 0$, mappings. The assumption that maximal smoothness leads to optimal generalisation no longer holds when an alternative metric is permitted to influence the mapping. In a classification scenario, for example, curvature will be necessarily high at class boundaries, and this will affect the choice of σ in addition to requiring greater consideration of the appropriate number of basis functions to use. Nevertheless, the implicit weight decay element may still be of key benefit, but further study is necessary in this direction.

NEUROSCALE for classification. The use of particular feature spaces derived by NEUROSCALE as the basis for a subsequent data classification in Chapter 4 proved very effective, although the experiments were necessarily concise. This is an area that offers much potential for further study, particularly in terms of the effect of parameter 'α', the number of features 'q' and the form of subjective dissimilarities used. Another important question is whether the self-regularising component within relatively supervised NEUROSCALE models is relevant to the classification context, and does this lead to better prediction performance?

Radial basis functions. One adjustable element of the NEUROSCALE architecture is the type of basis function that is utilised. Various types were used in experiments (Gaussian, thin-plate splines, zlogz and cubic), with some being better than others for various datasets. However, no formal comparison / evaluation of basis function type has been undertaken. Additionally, the experiments in section 6.6.3 on the optimal width of Gaussian basis functions were generally consistent with predictions but revealed an unexplained mid-range 'bump' in the test error profile, which warrants further investigation. Furthermore, while parts of Chapter 6 described how generalisation did not deteriorate with large numbers of basis functions, there was no consideration of the smallest number of such functions that might be used (which would lead to computational improvements).

The scaling problem. The major drawback of NEUROSCALE is that the training time scales in the square of the number of patterns in the dataset, and thus places a relatively low maximum limit thereon

($\approx$ 1000). Chapter 7 considered some of the previous effort directed at alleviating this problem, but it was argued that most of the approaches were unprincipled and amounted to some form of heuristic subset selection. A very recent development has been the formulation of a topographic map, related to Kohonen's paradigm, based on the concept of a *generative* model (one that maps from the feature space to the data space) [Bishop, Svensén, and Williams 1996]. An adaptation of this model offers a potential for application to a Sammon-like mapping, and one that would permit a reduction in computation through the use of a constrained mixture-modelling approach. Indeed, an additional advantage would be the defining of a density approximation in the data space, a measure that is absent in Sammon's algorithm. Because of the importance of the scaling problem, this would perhaps be the most relevant, interesting and profitable direction for future research.

Bibliography

Aarts, E. H. L. and J. Korst (1989). *Simulated annealing and Boltzmann machines : a stochastic approach to combinatorial optimization and neural computing*. Chichester: Wiley.

Anonymous (1996). On-line collection of references concerning theory and application of Kohonen's self-organising map and related methods. Available on the world-wide-web at `http://liinwww.ira.uka.de/bibliography/Neural/SOM.LVQ.html`, or via `ftp` at `cochlear.hut.fi`.

Balakrishnan, P. V., M. C. Cooper, V. S. Jacob, and P. A. Lewis (1994). A study of the classification capabilities of neural networks using unsupervised learning: a comparison with *k*-means clustering. *Psychometrika* ***59***(4), 509–525.

Baldi, P. and K. Hornik (1989). Neural networks and principal component analysis: Learning from examples without local minima. *Neural Networks* ***2***(1), 53–58.

Baldi, P. and K. Hornik (1995). Learning in linear neural networks: a survey. *IEEE Transactions on Neural Networks* ***6***(4), 837–858.

Barrett, P. J. (1995). *Data visualisation for marketing databases*. Ph. D. thesis, Aston University, Birmingham, UK.

Bauer, H.-U. and K. R. Pawelzik (1992). Quantifying the neighbourhood preservation of self-organizing feature maps. *IEEE Transactions on Neural Networks* ***3***(4), 570–579.

Bellman, R. E. (1961). *Adaptive Control Processes*. New Jersey: Princeton University Press.

Bezdek, J. C. and N. R. Pal (1995). An index of topological preservation for feature extraction. *Pattern Recognition* ***28***(3), 381–391.

Bishop, C. M. (1991). Improving the generalisation properties of radial basis function neural networks. *Neural Computation* ***3***, 579–588.

Bishop, C. M. (1993). Curvature driven smoothing: a learning algorithm for feedforward networks. *IEEE Transactions on Neural Networks* ***4***(5), 882–884.

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. Oxford: Clarendon Press.

Bishop, C. M., M. Svensén, and C. K. I. Williams (1996). GTM: A principled alternative to the self-organizing map. Technical Report NCRG/96/015, Neural Computing Research Group, Aston University, Aston Street, Aston Triangle, Birmingham B4 7ET, UK. Submitted to *Neural Computation*.

Biswas, G., A. K. Jain, and R. C. Dubes (1981). Evaluation of projection algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence* ***PAMI-3***(6), 701–708.

Blasdel, G. and G. Salama (1986). Voltage sensitive dyes reveal a modular organization in monkey striate cortex. *Nature* ***321***, 579–585.

Bottou, L. and Y. Bengio (1995). Convergence properties of the *k*-means algorithms. In G. Tesauro, D. S. Touretzky, and T. K. Leen (Eds.), *Advances in Neural Information Processing Systems 7*, pp. 585–592. Cambridge, Mass: MIT Press.

Broomhead, D. S. and D. Lowe (1988). Multivariable functional interpolation and adaptive networks. *Complex Systems* ***2***(3), 321–355.

Bryan, J. G. (1951). The generalized discriminant function: mathematical foundation and computational routine. *Harvard Educational Review* ***21***, 90–95.

Carroll, J. D. and P. Arabie (1980). Multidimensional scaling. *Annual Review of Psychology* ***31***, 607–649.

Chang, C. L. and R. C. T. Lee (1973). A heuristic relaxation method for nonlinear mapping in cluster analysis. *IEEE Transactions on Systems, Man and Cybernetics* ***3***, 197–200.

Chatfield, C. and A. J. Collins (1980). *Introduction to Multivariate Analysis*. London: Chapman & Hall.

Cox, T. F. and G. Ferry (1993). Discriminant analysis using non-metric multidimensional scaling. *Pattern Recognition* ***26***(1), 145–153.

Crace, J. (1995). Latest in the rating game. *The Guardian*, June 24.

Davison, M. L. (1983). *Multidimensional Scaling*. New York: Wiley.

de Leeuw, J. (1988). Covergence of the majorisation method for multidimensional scaling. *Journal of Classification* ***5***, 163–180.

de Soete, G., L. Hubert, and P. Arabie (1987). The comparative performance of simulated annealing on two problems of combinatorial data analysis. In *First Conf. IFCS, Technical University of Aachen*.

Domine, D., J. Devillers, M. Chastrette, and W. Karcher (1993). Non-linear mapping for structure-activity and structure-property modelling. *Journal of Chemometrics* ***7***, 227–242.

Dong, D. and T. J. McAvoy (1996). Nonlinear principal component analysis — based on principal curves and neural networks. *Computers and Chemical Engineering* ***20***(1), 65–78.

Duda, R. O. and P. E. Hart (1973). *Pattern Classification and Scene Analysis*. New York: Wiley.

Dzwinel, W. (1994). How to make Sammon's mapping useful for multidimensional data structures analysis. *Pattern Recognition* ***27***(7), 949–959.

Ekman, G. (1954). Dimensions of color vision. *Journal of Psychology* ***38***, 467–474.

Erwin, E., K. Obermayer, and K. Schulten (1992). Self-organizing maps: ordering, convergence properties and energy functions. *Biological Cybernetics* ***67***, 47–55.

Fisher, R. A. (1936). The use of multiple measurements on taxonomic problems. *Annals of Eugenics* ***7***, 179–188.

Friedman, J. H. (1994). An overview of predictive learning and function approximation. In V. Cherkassky, J. H. Friedman, and H. Wechsler (Eds.), *From Statistics to Neural Networks — Theory and Pattern Recognition Applications*, Volume 136 of *NATO ASI Series F: Computer and Systems Sciences*, pp. 1–61. Berlin: Springer-Verlag.

Friedman, J. H. (1995). Introduction to computational learning and statistical prediction. Tutorial 6, ICANN95, Paris.

Friedman, J. H. and J. W. Tukey (1974). A projection pursuit algorithm for exploratory data analysis. *IEEE Transactions on Computers* ***C-23***, 881–889.

Fritzke, B. (1994). Growing cell structures — a self-organizing network for unsupervised and supervised learning. *Neural Networks* ***7***(9), 1441–1460.

Fukunaga, K. (1990). *Introduction to Statistical Pattern Recognition* (Second ed.). London: Academic Press.

Gallinari, P., S. Thiria, F. Badran, and F. Fogelman-Soulié (1991). On the relations between discriminant analysis and multilayer perceptrons. *Neural Networks* ***4***, 349–360.

Gamelin, T. W. and R. E. Greene (1983). *Introduction to Topology*. Philadelphia: Saunders College Publishing.

Geman, S., E. Bienenstock, and R. Doursat (1992). Neural networks and the bias / variance dilemma. *Neural Computation* ***4***(1), 1–58.

Golub, G. and W. Kahan (1965). Calculating the singular values and pseudo-inverse of a matrix. *SIAM Numerical Analysis, B* **2**(2), 205–224.

Goodhill, G. J., S. Finch, and T. J. Sejnowski (1995). Quantifying neighbourhood preservation in topographic mappings. Technical Report INC-9505, Institute for Neural Computation, The Salk Institute for Biological Studies, 10010 North Torrey Pines Road, La Jolla, CA 92037, USA.

Goodhill, G. J., M. W. Simmen, and D. J. Willshaw (1995). An evaluation of the use of multidimensional scaling for understanding brain connectivity. *Philosophical Transactions of the Royal Society of London Series B* ***348***, 265–280.

Gower, J. C. (1966). Some distance properties of latent root and vector methods used in multivariate analysis. *Biometrika* ***53***, 325–338.

Gower, J. C. and P. Legendre (1986). Metric and Euclidean properties of dissimilarity coefficients. *Journal of Classification* ***3***, 5–48.

Gower, J. C. and G. J. S. Ross (1969). Minimal spanning trees and single linkage cluster analysis. *Applied Statistics* ***18***(1), 54–64.

Hastie, T. and W. Stuetzle (1989). Principal curves. *Journal of the American Statistical Association* ***84***, 502–516.

Haykin, S. (1994). *Neural Networks: a comprehensive foundation*. New York: Macmillan.

Heiser, W. J. (1991). A generalized majorisation method for least squares multidimensional scaling of pseudodistances that may be negative. *Psychometrika* ***56***(1), 7–27.

Higher Education Funding Council for England (1994). The 1992 RAE database. Available through JANET on NISSWAIS.

Hinton, G. E. (1989). Connectionist learning procedures. *Artificial Intelligence* ***40***, 185–234.

Hofmann, T. (1995). Personal communication.

Hofmann, T. and J. Buhmann (1995). Multidimensional scaling and data clustering. In G. Tesauro, D. S. Touretzky, and T. K. Leen (Eds.), *Advances in Neural Information Processing Systems 7*, pp. 459–466. Cambridge, Mass: MIT Press.

Huber, P. J. (1985). Projection pursuit. *The Annals of Statistics* ***13***(2), 435–475.

Jain, A. K. and J. Mao (1992). Artificial neural network for nonlinear projection of multivariate data. In *IJCNN International Joint Conference on Neural Networks*, Volume **3**, pp. 335–340. New York: IEEE.

Johnes, J., J. Taylor, and B. Francis (1993). The research performance of UK universities: a statistical analysis of the results of the 1989 Research Selectivity Exercise. *Journal of the Royal Statistical Society, A* ***156***(2), 271–286.

Joint Performance Indicators Working Group (1994). Explanatory and statistical material to accompany recomendations for research indicators. Technical report, Higher Eduction Funding Council for England, Northavon House, Coldharbour Lane, Bristol, BS16 1QD.

Jolliffe, I. T. (1986). *Principal Component Analysis*. New York: Springer-Verlag.

Kaas, J. H., R. J. Nelson, M. Sur, C.-S. Lin, and M. M. Merzenich (1979). Multiple representations of the body within the primary somatosensory cortex of primates. *Science* ***204***, 521–523.

Kakusho, O. and R. Mizoguchi (1983). A new algorithm for non-linear mapping with applications to dimension and cluster analyses. *Pattern Recognition* ***16***(1), 109–117.

Karhunen, J. (1994). Stablilty of Oja's PCA subspace rule. *Neural Computation* ***6***(4), 739–747.

Karhunen, J. and J. Joutsensalo (1994). Representation and separation of signals using nonlinear PCA type learning. *Neural Networks* ***7***(1), 113–127.

Klein, R. W. and R. C. Dubes (1989). Experiments in projection and clustering by simulated annealing. *Pattern Recognition* ***22***(2), 213–220.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics* ***43***, 59–69.

Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE* ***78***(9), 1464–1480.

Kohonen, T. (1995). *Self-Organizing Maps*. Berlin: Springer-Verlag.

Koontz, W. L. G. and K. Fukunaga (1972). A nonlinear feature extraction algorithm using distance transformation. *IEEE Transactions On Computers* ***C-21***(1), 56–63.

Kramer, M. A. (1991). Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal* ***37***(2), 233–243.

Kruskal, J. B. (1964a). Multidimensional scaling by optimising goodness of fit to a nonmetric hypothesis. *Psychometrika* ***29***(1), 1–27.

Kruskal, J. B. (1964b). Nonmetric multidimensional scaling: a numerical method. *Psychometrika* ***29***(2), 115–129.

Kruskal, J. B. (1971). Comments on 'A nonlinear mapping for data structure analysis'. *IEEE Transactions on Computers* ***C-20***, 1614.

Krzanowski, W. J. and F. H. C. Marriott (1994). *Multivariate Analysis Part I: Distributions, Ordination and Inference*. London: Edward Arnold.

LeBlanc, M. and R. Tibshirani (1994). Adaptive principal surfaces. *Journal of the American Statistical Association* ***89***(425), 53–64.

Lee, R. C. T., J. R. Slagle, and H. Blum (1977). A triangulation method for the sequential mapping of points from N-space to two-space. *IEEE Transactions on Computers* ***C-26***, 288–292.

Levine, S. H. and J. G. Kreifeldt (1994). Uniquely representing point patterns with minimal information. *IEEE Transactions on Systems, Man and Cybernetics* ***SMC-24***(6), 895–900.

Li, X., J. Gasteiger, and J. Zupan (1993). On the topology distortion in self-organising feature maps. *Biological Cybernetics* ***70***, 189–198.

Linde, Y., A. Buzo, and R. M. Gray (1980). An algorithm for vector quantizer design. *IEEE Transactions on Communications* ***COM-28***, 84–95.

Lippmann, R. P. (1994). Neural networks, Bayesian *a posteriori* probabilities, and pattern classification. In V. Cherkassky, J. H. Friedman, and H. Wechsler (Eds.), *From Statistics to Neural Networks — Theory and Pattern Recognition Applications*, Volume 136 of *NATO ASI Series F: Computer and Systems Sciences*, pp. 83–104. Berlin: Springer-Verlag.

Lowe, D. (1993). Novel 'topographic' nonlinear feature extraction using radial basis functions for concentration coding in the 'artificial nose'. In *3rd IEE International Conference on Artificial Neural Networks*. London: IEE.

Lowe, D. (1995). Radial basis function networks. In M. A. Arbib (Ed.), *The Handbook of Brain Theory and Neural Networks*, pp. 779–782. Cambridge, Mass: MIT Press.

Lowe, D. and M. E. Tipping (1995). A novel neural network technique for exploratory data analysis. In *Proceedings of ICANN '95 (Scientific Conference)*, Volume 1, pp. 339–344. Paris: EC2 & Cie.

Lowe, D. and M. E. Tipping (1996). Feed-forward neural networks and topographic mappings for exploratory data analysis. *Neural Computing and Applications* ***4***, 83–95.

Lowe, D. and A. R. Webb (1990). Exploiting prior knowledge in network optimization: an illustration from medical prognosis. *Network: Computation in Neural Systems* ***1***(3), 299–323.

MacLeod, D. (1995). Head hunters. *The Guardian*, September 26.

MacQueen, J. (1967). Some methods of classification and analysis of multivariate observations. In L. M. LeCam and J. Neyman (Eds.), *Proceedings of the 5th Berkeley Symposium on Mathematics, Statistics and Probability*, pp. 281–297. Berkeley, CA: University of California Press.

Mao, J. and A. K. Jain (1995). Artificial neural networks for feature extraction and multivariate data projection. *IEEE Transactions on Neural Networks* ***6***(2), 296–317.

Mardia, K. V. (1978). Some properties of classical multidimensional scaling. *Communications in Statistics — Theory and Methods* ***A7***(13), 1233–1241.

Mardia, K. V., J. T. Kent, and J. M. Bibby (1979). *Multivariate Analysis*. Probability and Mathematical Statistics. London: Academic Press.

Martinetz, T. and K. Schulten (1991). A "neural gas" network learns topologies. In T. Kohonen, K. Mäkisara, K. Simula, and J. Kangas (Eds.), *Artificial Neural Networks*, pp. 397–402. Amsterdam: Springer.

Mathar, R. and A. Zilinskas (1993). On global optimization in two-dimensional scaling. *Acta Applicandae Mathematicae* ***33***, 109–118.

Moody, J. E. and C. J. Darken (1989). Fast learning in networks of locally-tuned processing units. *Neural Computation* ***1***(2), 281–294.

Niemann, H. and J. Weiss (1979). A fast-converging algorithm for nonlinear mapping of high-dimensional data to a plane. *IEEE Transactions on Computers* ***C-28***, 142–147.

Oja, E. (1982). A simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology* ***15***, 267–273.

Oja, E. (1989). Neural networks, principal components, and subspaces. *International Journal of Neural Systems* ***1***, 61–68.

Oja, E. (1992). Principal components, minor components, and linear neural networks. *Neural Networks* ***5***, 927–935.

Park, J. and I. W. Sandberg (1991). Universal approximation using radial basis function networks. *Neural Computation* ***3***(2), 246–257.

Plumbley, M. D. (1995). Lyapunov functions for convergence of principal component algorithms. *Neural Networks* ***8***(1), 11–23.

Plybon, B. F. (1992). *An Introduction to Applied Numerical Analysis*. Boston: PWS-KENT.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery (1992). *Numerical Recipes in C* (Second ed.). Cambridge: Cambridge University Press.

Pykett, C. E. (1978). Improving the efficiency of Sammon's nonlinear mapping by using clustering archetypes. *Electronics Letters* ***14***(25), 799–800.

Rao, C. R. (1948). The utilization of multiple measurements in problems of biological classification. *Journal of the Royal Statistical Society, Series B* ***10***, 159–203.

Richardson, M. W. (1938). Multidimensional psychophysics. *Psychological Bulletin* ***35***, 659–660.

Ritter, H. (1991). Asymptotic level density for a class of vector quantization processes. *IEEE Transactions on Neural Networks* ***2***, 173–175.

Ritter, H. and K. Schulten (1986). On the stationary state of Kohonen's self-organizing sensory mapping. *Biological Cybernetics* ***54***, 99–106.

Rothkopf, E. Z. (1957). A measure of stimulus similarity and errors in some paired-associate learning tasks. *Journal of Experimental Psychology* ***53***, 94–104.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams (1986). Learning internal representations by error propagation. In D. E. Rumelhart, J. L. McClelland, and the PDP Research Group (Eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Volume 1, Chapter 8, pp. 318–362. Cambridge, Mass: MIT Press.

Sammon, J. W. (1969). A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers* ***C-18***(5), 401–409.

Sanger, T. D. (1989). Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural Networks* ***2***(6), 459–473.

Saund, E. (1989). Dimensionality-reduction using connectionist networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* ***PAMI-11***(3), 304–315.

Schiffman, S. S., M. L. Reynolds, and F. W. Young (1981). *Introduction to Multidimensional Scaling. Theory, Methods and Applications.* New York: Academic Press.

Schwartz, E. L., A. Shaw, and E. Wolfson (1989). A numerical solution to the generalised mapmaker's problem: flattening nonconvex polyhedral surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence* ***PAMI-11***(9), 1005–1008.

Shepard, R. N. (1958). Stimulus and response generalisation: Deduction of the generalisation gradient from a trace model. *Psychological Review* ***65***, 242–256.

Shepard, R. N. (1962a). The analysis of proximities: Multidimensional scaling with an unknown distance function, I. *Psychometrika* ***27***(2), 125–140.

Shepard, R. N. (1962b). The analysis of proximities: Multidimensional scaling with an unknown distance function, II. *Psychometrika* ***27***(3), 219–246.

Shepard, R. N. and J. D. Carroll (1966). Parametric representation of nonlinear data structures. In P. Krishnaiah (Ed.), *Multivariate Analysis*, pp. 561–592. New York: Academic Press.

Siedlecki, W., K. Siedlecka, and J. Sklansky (1988). An overview of mapping techniques for exploratory pattern analysis. *Pattern Recognition* ***21***(5), 411–429.

SOM Programming Team (1995). SOM_PAK: the Self-Organizing Map Program Package, Version 3.1. Available by anonymous `ftp` from Helsinki University of Technology at `cochlea.hut.fi`.

Strang, G. (1988). *Linear Algebra and its Applications.* Orlando: Harcourt Brace Jovanovich.

Suga, N. and W. E. O'Neill (1979). Neural axis representing target range in the auditory cortex of the mustache bat. *Science* ***206***, 351–353.

Tarazaga, P. and M. W. Trosset (1993). An optimization problem on subsets of the symmetric positive-definite matrices. *Journal of Optimization Theory and Applications* ***79***(3), 513–524.

Tattersall, G. D. and P. R. Limb (1994). Visualisation techniques for data mining. *BT Technology Journal, British Telecom Labs, Ipswich IP5 7RE, UK* ***12***(4), 23–31.

Taylor, J. (1994). Measuring research performance in business and management studies in the United Kingdom: The 1992 Research Assessment Exercise. *British Journal of Management* ***5***, 275–288.

Taylor, J. (1995). A statistical analysis of the 1992 Research Assessment Exercise. *Journal of the Royal Statistical Society A* ***158***(2), 241–261.

Tikhonov, A. N. and V. Y. Arsenin (1977). *Solutions of Ill-Posed Problems.* Washington, D.C.: V.H. Winston.

Tipping, M. E. (1996). Topographic mappings, classical multidimensional scaling and the principal subspace network. Submitted for publication in *IEEE Transactions on Neural Networks.*

Torgerson, W. S. (1952). Multidimensional scaling: I. theory and method. *Psychometrika* ***17***, 401–419.

Torgerson, W. S. (1958). *Theory and methods of scaling.* New York: Wiley.

Webb, A. R. (1992). Non-metric multidimensional scaling using feed-forward networks. CSE1 Research Note 192, Defence Research Agency, Malvern, UK.

Webb, A. R. (1995). Multidimensional scaling by iterative majorisation using radial basis functions. *Pattern Recognition* ***28***(5), 753–759.

Webb, A. R. and D. Lowe (1990). The optimised internal representation of multilayer classifier networks performs nonlinear discriminant analysis. *Neural Networks* ***3***(4), 367–375.

Weiss, S. M. and C. A. Kulikowski (1991). *Computer Systems That Learn.* San Mateo, CA: Morgan Kaufmann.

Williams, R. (1985). Feature discovery through error-correcting learning. Technical Report 8501, Institute of Cognitive Science, University of California, San Diego.

Willshaw, D. J. and C. von der Malsburg (1976). How patterned neural connections can be set up by self-organisation. In *Proceedings of the Royal Society of London B*, Volume **194**, pp. 431–445.

Wyatt, J. L. and I. M. Elfadel (1995). Time-domain solutions of Oja's equations. *Neural Computation* ***7***(5), 915–922.

Xu, L. (1993). Least mean square error reconstruction principle for self-organising neural-nets. *Neural Networks* ***6***(5), 627–648.

Yan, W.-Y., U. Helmke, and J. B. Moore (1994). Analysis of Oja's flow for neural networks. *IEEE Transactions on Neural Networks* ***5***(5), 674–683.

Young, F. W. and D. F. Harris (1990). Multidimensional scaling: procedure ALSCAL. In M. Norusis (Ed.), *SPSS base system: user's guide*, pp. 397–461. Chicago: SPSS.

Young, M. P. (1992). Objective analysis of the topological organization of the primate cortical visual system. *Nature* ***358***, 152–155.