Chapter 4

Dimensionality reduction

This chapter introduces and defines the problem of dimensionality reduction, discusses the topics of the curse of the dimensionality and the intrinsic dimensionality and then surveys non-probabilistic methods for dimensionality reduction, that is, methods that do not define a probabilistic model for the data. These include linear methods (PCA, projection pursuit), nonlinear autoassociators, kernel methods, local dimensionality reduction, principal curves, vector quantisation methods (elastic net, self-organising map) and multidimensional scaling methods. One of these methods (the elastic net) does define a probabilistic model but not a continuous dimensionality reduction mapping. If one is interested in stochastically modelling the dimensionality reduction mapping then the natural choice are latent variable models, discussed in chapter 2. We close the chapter with a summary and with some thoughts on dimensionality reduction with discrete variables.

4.1 Introduction

Consider an application in which a system processes data in the form of a collection of real-valued vectors: speech signals, images, etc. Suppose that the system is only effective if the dimension of each individual vector—the number of components of the vector—is not too high, where *high* depends on the particular application. The problem of dimensionality reduction appears when the data are in fact of a higher dimension than tolerated. For example, take the following typical cases:

- A face recognition/classification system based on $m \times n$ greyscale images which, by row concatenation, can be transformed into mn-dimensional real vectors. In practice, one could have images of $m = n = 256$, or 65536-dimensional vectors; if, say, a multilayer perceptron was to be used as the classification system, the number of weights would be exceedingly large and would require an enormous training set to avoid overfitting. Therefore we need to reduce the dimension. While a crude solution in this case would be to simply scale down the images to a manageable size, more elaborate approaches exist.

- A statistical analysis of a multivariate population. Typically there will be a few variables and the analyst is interested in finding clusters or other structure of the population and/or interpreting the variables. To that aim, it is quite convenient to visualise the data, which requires reducing its dimensionality to 2 or 3.

Therefore, in a number of occasions it can be useful or even necessary to first reduce the dimensionality of the data to a manageable size, keeping as much of the original information as possible, and then feed the reduced-dimension data into the system. Figure 4.1 summarises this situation, showing the dimensionality reduction as a preprocessing stage in the whole system.

More generally, whenever the intrinsic dimensionality of a data set is smaller than the actual one, dimensionality reduction can bring an improved understanding of the data apart from a computational advantage. Dimensionality reduction can also be seen as a feature extraction or coding procedure, or in general as a representation in a different coordinate system. This is the basis for the definition given in section 4.2.

4.1.1 Classes of dimensionality reduction problems

We attempt here a rough classification of the dimensionality reduction problems:

This chapter is an extended version of reference Carreira-Perpiñán (1996).

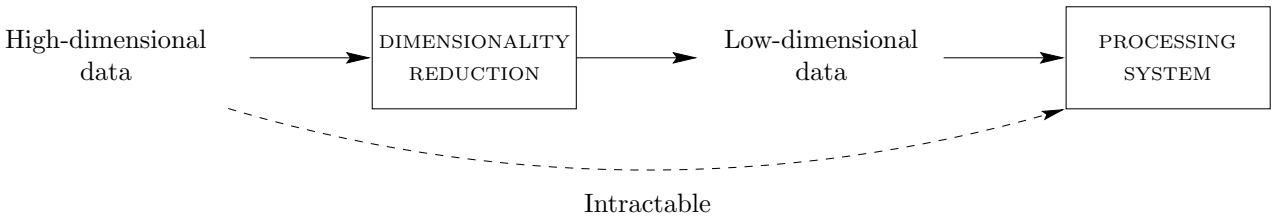


Figure 4.1: The dimensionality reduction problem. A given processing system is only effective with vector data of not more than a certain dimension, so data of higher dimension must be reduced before being fed into the system.

- *Hard* dimensionality reduction problems, in which the data have dimensionality ranging from hundreds to perhaps hundreds of thousands of components, and usually a drastic reduction (possibly of orders of magnitude) is sought. The components are often repeated measures of a certain magnitude in different points of space or in different instants of time. In this class we would find pattern recognition and classification problems involving images (e.g. face recognition, character recognition, etc.) or speech (e.g. auditory models).
- *Soft* dimensionality reduction problems, in which the data is not too high-dimensional (less than a few tens of components), and the reduction not very drastic. Typically, the components are observed or measured values of different variables, which have a straightforward interpretation. Most statistical analyses in fields like social sciences, psychology, etc. fall in this class.
- *Visualisation* problems, in which the data does not normally have a very high dimension in absolute terms, but we need to reduce it to 2 or 3 in order to plot it. Several representation techniques (reviewed by Scott, 1992) exist that allow to visualise up to about 5-dimensional data sets, using colours, rotation, stereography, glyphs or other devices, but they lack the appeal of a simple plot; a well-known one is the grand tour (Asimov, 1985). Chernoff faces (Chernoff, 1973) allow even a few more dimensions, but are difficult to interpret and do not produce a spatial view of the data.

If we allow the time variable in, we find two further categories: *static dimensionality reduction* and *time-dependent dimensionality reduction*. The latter could possibly be useful for vector time series, such as video sequences or continuous speech. We deal here only with static dimensionality reduction.

4.2 Definition of the problem of dimensionality reduction

Suppose we have a sample $\{\mathbf{t}_n\}_{n=1}^N$ of D-dimensional vectors lying in a data space $\mathcal{T}$ (usually $\mathbb{R}^D$ or a subset of it[1]). The fundamental assumption that justifies the dimensionality reduction is that the sample actually lies, at least approximately, on a manifold[2] (nonlinear in general) of smaller dimension than the data space. The goal of dimensionality reduction is to find a representation of that manifold (a coordinate system) that will allow to project the data vectors on it and obtain a low-dimensional, compact representation of the data.

Formally, dimensionality reduction consists of the following problem: given a sample $\{\mathbf{t}_n\}_{n=1}^N \subset \mathcal{T}$, find:

- A space $\mathcal{X}$ of dimension L (typically $\mathbb{R}^L$ or a subset of it).
- A **dimensionality reduction mapping $\mathbf{F}$**:

$$\begin{array}{rccc} \mathbf{F}: & \mathcal{T} & \longrightarrow & \mathcal{X} \\ & \mathbf{t} & \mapsto & \mathbf{x} = \mathbf{F}(\mathbf{t}). \end{array}$$

  As in section 2.9.1, we call $\mathbf{x}$ the *reduced-dimension representative* of $\mathbf{t}$.

- A smooth, nonsingular[3] **reconstruction mapping $\mathbf{f}$**:

$$\begin{array}{rccc} \mathbf{f}: & \mathcal{X} & \longrightarrow & \mathcal{M} \subset \mathcal{T} \\ & \mathbf{x} & \mapsto & \mathbf{t} = \mathbf{F}(\mathbf{x}). \end{array}$$

[1] As in chapter 2, we restrict ourselves to continuous variables.

[2] Section A.7 gives a formal definition of L-manifolds and coordinate systems.

[3] The reasons for requiring that the reconstruction mapping be smooth and nonsingular are the same as in chapter 2: to ensure that the domain and range have the same dimension (nonsingular) and to preserve the topographic structure of the domain (continuity). Piecewise smooth functions would be acceptable too.

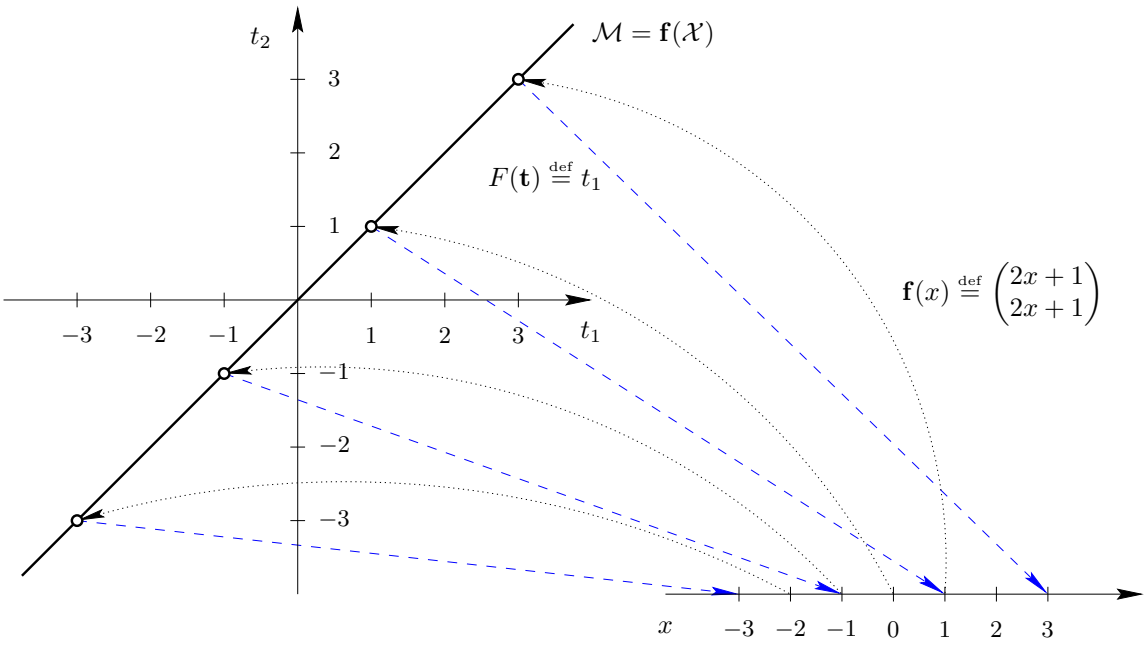


Figure 4.2: The dimensionality reduction mapping $\mathbf{F}$ (dashed lines) and the reconstruction mapping $\mathbf{f}$ (dotted lines) need not verify $\mathbf{F} \circ \mathbf{f} = \text{identity}$.

Such that:

- $L < D$ is as small as possible.
- The following condition is satisfied:

  The manifold $\mathcal{M} \stackrel{\text{def}}{=} \mathbf{f}(\mathcal{X})$ *approximately contains all the sample points:* $\{\mathbf{t}_n\}_{n=1}^N \subsetneq \mathcal{M}$. (DR)

  This condition can be restated in a different way:

  The reconstruction error of the sample is small. (DR′)

  The reconstruction error of the sample is defined as $E_d(\{\mathbf{t}_n\}_{n=1}^N) \stackrel{\text{def}}{=} \sum_{n=1}^N d(\mathbf{t}_n, \mathbf{t}_n^*)$ where $\mathbf{t}_n^* \stackrel{\text{def}}{=} \mathbf{f}(\mathbf{F}(\mathbf{t}_n))$ is the reconstructed vector for point $\mathbf{t}_n$ and d is a suitable distance in the space $\mathcal{T}$ (e.g. the Euclidean distance in $\mathbb{R}^D$).

Conditions (DR) and (DR′) are not equivalent: (DR′) implies (DR) but not vice versa. This is because, for a given manifold $\mathcal{M} = \mathbf{f}(\mathcal{X}) \subset \mathcal{T}$, $\mathbf{F} \circ \mathbf{f}$ need not be the identity mapping, as fig. 4.2 shows. Thus, it is possible that $\{\mathbf{t}_n\}_{n=1}^N \subset \mathcal{M}$, i.e., for each $\mathbf{t}_n$ there exists a point $\mathbf{x}_n \in \mathcal{X}$ with $\mathbf{f}(\mathbf{x}_n) = \mathbf{t}_n$, but $\mathbf{F}(\mathbf{t}_n) \neq \mathbf{x}_n$. Therefore $\mathbf{t}_n^* \stackrel{\text{def}}{=} \mathbf{f}(\mathbf{F}(\mathbf{t}_n)) \neq \mathbf{t}_n$ and $E_d(\{\mathbf{t}_n\}_{n=1}^N)$ may be large. This is not an arbitrary argument: the dimensionality reduction mappings derived from latent variable models are of this kind, due to the existence of a probability distribution on the space $\mathcal{X}$ and of a noise model, as described in section 2.9.

Due to the dimensionality mismatch between $\mathcal{X}$ and $\mathcal{T}$, there will be a whole submanifold of dimension $D - L$ in $\mathcal{T}$ that will be mapped onto the same point in $\mathcal{X}$, i.e., $\mathbf{F}^{-1}(\mathbf{x}) \stackrel{\text{def}}{=} \{\mathbf{t} \in \mathcal{T} : \mathbf{F}(\mathbf{t}) = \mathbf{x}\}$ will be a submanifold of dimension $D - L$, as discussed in section 2.9.1. For the example of fig. 4.2, $\mathbf{F}^{-1}(\mathbf{x})$ is the vertical line passing through the point $\mathbf{t} = (x\ 0)^T$.

The election of the coordinate system for a given manifold is not unique. For example, in figure 4.3 a one-dimensional nonlinear manifold in $\mathbb{R}^3$ (a curve), namely a spiral of radius R and step s, is parameterised in terms of the dimensionless parameter x: $\mathcal{M} \stackrel{\text{def}}{=} \{\mathbf{t} \in \mathbb{R}^3 : \mathbf{t} = \mathbf{f}(x), x \in [x_A, x_B]\}$ for $\mathbf{f}(x) \stackrel{\text{def}}{=} (R \sin 2\pi x, R \cos 2\pi x, sx)^T$. We could reparameterise the manifold in terms of the arc length λ between points $\mathbf{t}_A \stackrel{\text{def}}{=} \mathbf{f}(x_A)$ and $\mathbf{t}_B \stackrel{\text{def}}{=} \mathbf{f}(x_B)$:

$$\lambda \stackrel{\text{def}}{=} \int_{\mathbf{t}_A}^{\mathbf{t}_B} \sqrt{\left(\frac{dt_1}{dx}\right)^2 + \left(\frac{dt_2}{dx}\right)^2 + \left(\frac{dt_3}{dx}\right)^2}\, dx = \sqrt{(2\pi R)^2 + s^2}(x_B - x_A).$$

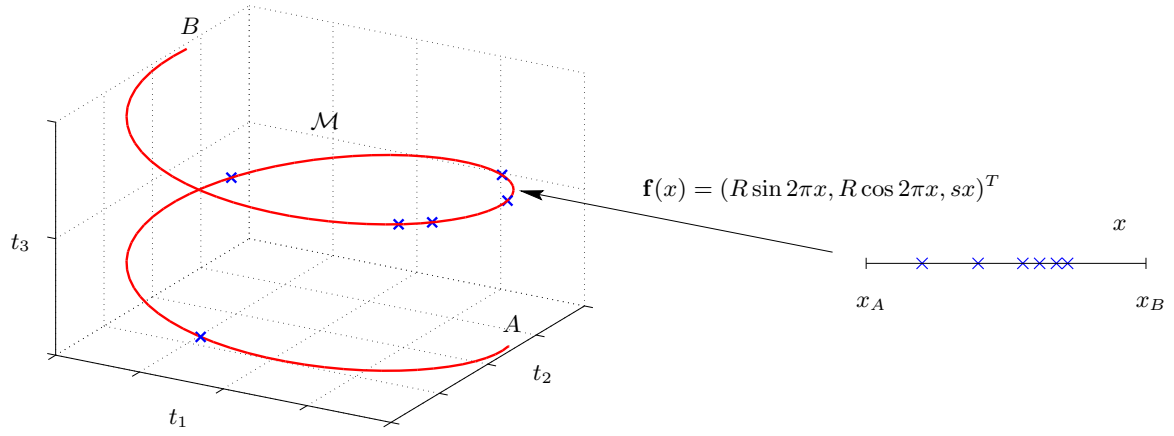


Figure 4.3: An example of coordinate representation of a one-dimensional manifold $\mathcal{M}$ in $\mathbb{R}^3$ (a curve): the segment of spiral $\mathcal{M} = \{\mathbf{t} \in \mathbb{R}^3 : \mathbf{t} = \mathbf{f}(x), x \in [x_A, x_B]\}$ for $\mathbf{f}(x) = (R\sin 2\pi x, R\cos 2\pi x, sx)^T$.

Yet another parameterisation would be in terms of the angle $\theta = 2\pi x$, etc. Another example are the Cartesian, spherical and cylindrical systems in $\mathbb{R}^3$. Any coordinate system is in principle acceptable, although some may be more appropriate than others for certain problems. Also, constraints on the problem may make the choice of coordinates unique. For example, in PCA the basis vectors are constrained to be orthonormal and ordered according to variance, which makes the choice of basis vectors unique (except when the data covariance matrix has multiple eigenvalues).

4.3 The curse of the dimensionality

The term *curse of the dimensionality*[4], coined by Bellman (1961), refers to the fact that, in the absence of simplifying assumptions, the sample size needed to estimate a function of several variables to a given degree of accuracy (i.e., to get a reasonably low-variance estimate) grows exponentially with the number of variables.

A related fact, responsible for the curse of the dimensionality, is the *empty space phenomenon* (Scott and Thompson, 1983): high-dimensional spaces are inherently sparse. For example, the probability that a point distributed uniformly in the unit 10-dimensional sphere falls at a distance of 0.9 or less from the centre is only 0.35. This is a difficult problem in multivariate density estimation, as regions of relatively very low density can contain a considerable part of the distribution, whereas regions of apparently high density can be completely devoid of observations in a sample of moderate size (Silverman, 1986). For example, for a one-dimensional standard normal $\mathcal{N}(0, 1)$, 70% of the mass is at points contained in a sphere of radius one standard deviation (i.e., the $[-1, 1]$ interval); for a 10-dimensional $\mathcal{N}(\mathbf{0}, \mathbf{I})$, that same (hyper)sphere contains only 0.02% of the mass and one has to take a radius of more than 3 standard deviations to contain 70%. Therefore, and contrarily to our intuition, in high-dimensional distributions the tails are much more important than in one-dimensional ones.

The curse of the dimensionality has the following consequence for density estimation: since most density estimation methods are based on some local average of the neighbouring observations (Silverman, 1986), in order to find enough neighbours in high-dimensional spaces, the neighbourhood has to reach out farther and the locality is lost. Another problem caused by the curse of the dimensionality is that, if there are linear correlations in the data (a very likely situation in high dimensions), the optimal mean integrated squared error when estimating the data density will be very large even if the sample size is arbitrarily large (Scott, 1992).

[4] As a philosophical aside, it is interesting that the curse of the dimensionality appears implicitly in Francis Bacon's inductive method. Bacon introduced the idea of induction in 1620 in his *Novum Organum* as a response to deduction, which Aristotle had formalised in his *Organum*. In Bacon's inductive method, the task of the scientist would be to derive laws of nature by carefully constructing tables containing the experimental data. Each line in a table would correspond to a combination of values of the variables having an influence on the phenomenon observed and the presence or absence of that phenomenon (a kind of binary classification problem in our days!), or the degree to which the phenomenon manifested itself (multivariate regression!). Given the combinatorial explosion of the table size, acknowledged by Bacon, it is not surprising that the method was never implemented—although its merit resides in the introduction of the concept of induction rather than in the method.

However, it is important to remark that the curse of the dimensionality is governed not by the dimensionality D of the data space $\mathcal{T}$, but by the intrinsic dimensionality L of the data. Or, more precisely, by the dimension L of the space $\mathcal{X}$ which the dimensionality reduction method is imposing. This is because the sample size required really depends on the volume of hyperspace occupied by the manifold being modelled, of dimension L, not on the higher-dimension space where it is embedded: thus the sample size grows as $\mathcal{O}(e^L)$. An example of this are latent variable models based on Monte Carlo sampling of the latent space (section 2.4), such as GTM.

Due to the fundamental character of the curse of the dimensionality, all dimensionality reduction methods (particularly the more general ones) are affected by it to some extent through the number of parameters that need to be estimated.

4.3.1 The geometry of high-dimensional spaces

The geometry of high-dimensional spaces provides a few surprises related to the curse of the dimensionality. Although, in fact, one should say that the surprises are in the usual, intuitive low-dimensional cases of 1 to 3 dimensions, when compared to the general (asymptotic) case of higher dimensions. We illustrate now some of these intriguing results for the case of the Euclidean space $\mathbb{R}^D$. First note that:

- The volume of the D-hypersphere of radius R is $V(\mathbb{S}_R^D) = V(\mathbb{S}_1^D)R^D$ with dimension-dependent constant

$$V(\mathbb{S}_1^D) = \frac{\pi^{D/2}}{\Gamma(\frac{D}{2}+1)}$$

  where $\Gamma(x)$ is the gamma function.

- The volume of the D-hypercube of side $2R$ is $V(\mathbb{C}_R^D) = V(\mathbb{C}_1^D)R^D$ with dimension-dependent constant $V(\mathbb{C}_1^D) = 2^D$.

Both volumes depend exponentially on the linear size of the object, but the constants are very different. This has as an interesting consequence a distortion of the space. Consider the following situations in the limit of high dimensions:

Sphere inscribed in a hypercube (Scott, 1992): the ratio of the volume of the hypersphere to the volume of the hypercube is

$$\frac{V(\mathbb{S}_1^D)}{V(\mathbb{C}_1^D)} = \frac{\pi^{D/2}}{2^D\Gamma(\frac{D}{2}+1)} \xrightarrow[D\to\infty]{} 0.$$

That is, with increasing dimension the volume of the hypercube concentrates on its corners and the centre becomes less important. Table 4.1 and figure 4.4 show the volumes $V(\mathbb{S}_1^D)$, $V(\mathbb{C}_1^D)$ and the ratio between them for several dimensions.

Hypervolume of a thin shell (Wegman, 1990): consider the volume between two concentric spheric shells of respective radii R and $R(1-\epsilon)$, with ϵ small. Then the ratio

$$\frac{V(\mathbb{S}_R^D) - V(\mathbb{S}_{R(1-\epsilon)}^D)}{V(\mathbb{S}_R^D)} = 1 - (1-\epsilon)^D \xrightarrow[D\to\infty]{} 1.$$

Hence, virtually all the content of a hypersphere is concentrated close to its surface, which is only a $(D-1)$-dimensional manifold (see section A.7). Thus, for data distributed uniformly over both the hypersphere and the hypercube, most of the data fall near the boundary and edges of the volume. This example illustrates one important aspect of the curse of the dimensionality mentioned earlier. Figure 4.4 illustrates this point for $\epsilon = 0.1$.

Tail probability of the multivariate normal (Scott, 1992): the preceding examples make it clear that most (spherical) neighbourhoods of data distributed uniformly over a hypercube in high dimensions will be empty. In the case of the standard D-dimensional normal distribution, the equiprobable contours are hyperspheres. The probability that a point is within a contour of density ϵ times the value at the mode, or, equivalently, inside a hypersphere of radius $\sqrt{-2\ln\epsilon}$, is:

$$\Pr\left[\|\mathbf{x}\|^2 \leq -2\ln\epsilon\right] = \Pr\left[\chi_D^2 \leq -2\ln\epsilon\right] \tag{4.1}$$

D	1	2	3	4	$\cdots$	10
$V(\mathbb{S}_1^D)$	2	π	$\frac{4}{3}\pi$	$\frac{\pi^2}{2}$	$\cdots$	$\frac{\pi^5}{120} \approx 2.55$
$V(\mathbb{C}_1^D)$	2	4	8	16	$\cdots$	1024
$\frac{V(\mathbb{S}_1^D)}{V(\mathbb{C}_1^D)}$	1	$\frac{\pi}{4}$	$\frac{\pi}{6}$	$\frac{\pi^2}{32}$	$\cdots$	$\frac{\pi^5}{122880} \approx 0.0025$

Table 4.1: Volumes of unit D-hypersphere and D-hypercube.

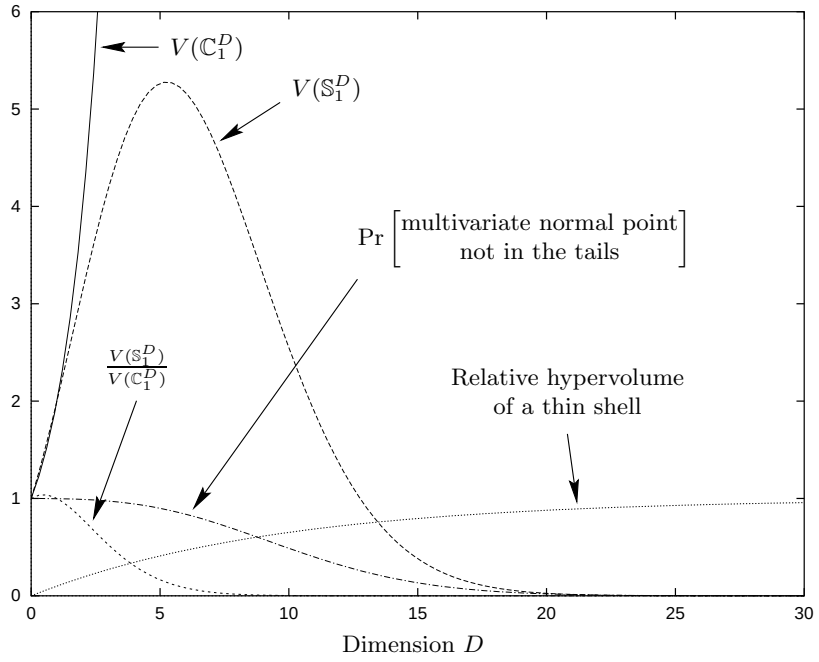


Figure 4.4: Dependence of several geometric quantities with the dimension (only natural numbers $D = 1, 2, \ldots$ are meaningful). See the main text for an explanation.

because if $\mathbf{t} = (t_1, \ldots, t_D)$ is distributed as a standard normal, then x_i, $i = 1, \ldots, D$ are univariate standard normal and $\|\mathbf{t}\|^2 = \sum_{d=1}^{D} t_d^2$ is distributed as a χ^2 distribution with D degrees of freedom. Equation 4.1 gives the probability that a random point will not fall in the tails, i.e., that it will fall in the medium- to high-density region. Figure 4.4 shows this probability for $\epsilon = 0.01$ (a radius of 3 standard deviations) and several dimensions: notice how around $D = 5$ the probability mass of a multivariate normal begins a rapid migration into the extreme tails. In very high dimensions the entire sample will be in the tails!

Diagonals in hyperspace (Scott, 1992): consider the hypercube $[-1, 1]^D$ and let any of the diagonal vectors from the centre to a corner be denoted by $\mathbf{v}$. Then $\mathbf{v}$ is one of the 2^D vectors of the form $(\pm 1, \pm 1, \ldots, \pm 1)^T$. The angle between a diagonal vector $\mathbf{v}$ and a coordinate axis $\mathbf{e}_d$ is given by

$$\cos\theta_D = \frac{\mathbf{v}\mathbf{e}_d}{\|\mathbf{v}\| \, \|\mathbf{e}_d\|} = \frac{\pm 1}{\sqrt{D}} \xrightarrow[D\to\infty]{} 0.$$

Thus, the diagonals are nearly orthogonal to all coordinate axes for large D.

Pairwise scatter diagrams essentially project the multivariate data onto all the 2-dimensional coordinate planes. Hence, any data cluster lying near a diagonal in hyperspace will be mapped into the origin in every paired scatterplot, while a cluster along a coordinate axis will be visible in some plot. Thus the choice of coordinate systems is critical in data analysis: 1- or 2-dimensional intuition is valuable but not infallible when continuing on to higher dimensions.

4.4 The intrinsic dimension of a sample

Consider a certain phenomenon governed by L independent variables. In practice, this phenomenon will actually appear as having (perhaps many) more degrees of freedom due to the influence of a variety of uncontrolled factors: noise, imperfection in the measurement system, addition of irrelevant variables, etc. However, provided this influence is not too strong as to completely mask the original structure, in principle we should be able to "filter" it out and recover the original variables or an equivalent set of them. We define the **intrinsic dimension**[5] of a phenomenon as the number of independent variables that explain satisfactorily that phenomenon. From a geometrical point of view, the intrinsic dimensionality would be the dimension L of the manifold that approximately embeds the sample data in a D-dimensional Euclidean space ($D > L$). The vagueness of the words *satisfactory* and *approximately* is intended, since the intrinsic dimensionality of a sample depends on the criteria that the user applies to the particular problem at hand, such as smoothness of the manifold, effect of noise, etc. A priori a discrete sample has dimension zero, but it is possible to make a manifold of any dimension pass through it. So the sample may have dimension one according to one criterion, dimension two according to a different one and so on. For example, in fig. 4.5 a one-dimensional manifold (the dotted curve) is forced to interpolate a set of points which naturally would seem to lie on a two-dimensional manifold (the shaded area). Thus, the problem of inferring the intrinsic dimensionality of a sample is ill-posed and requires of prior information to be solved.

The determination of the intrinsic dimensionality of a process given a sample of it is central to the problem of dimensionality reduction, because knowing it would eliminate the possibility of over- or underfitting. All the dimensionality reduction methods discussed in this thesis take the intrinsic dimensionality as a parameter to be given by the user; a trial-and-error process is necessary to obtain a satisfactory value for it (aided by model selection techniques such as cross-validation). In some practical applications, domain information may give insight into the intrinsic dimensionality. For probabilistic methods, such as latent variable models, a Bayesian approach can help to determine the intrinsic dimensionality. For example, one could consider the intrinsic dimensionality L as a trainable parameter over which a prior distribution is placed, perhaps uniform in $\{1, 2, \ldots, D\}$ (to reflect lack of information), or perhaps decreasing monotonically with increasing dimension (to favour the reduction of dimensionality). The posterior mode of the intrinsic dimensionality given the data sample could be chosen as the optimal dimensionality if a single value is desired (although, strictly, inferences should use the posterior parameter distribution, including L, rather than a single value). This has the problem that the rest of the parameters depend on L. A possible strategy is the use of a hierarchical prior model, as Richardson and Green (1997) have done to learn the number of components (and the rest of the parameters) of a finite mixture. Another possibility is to use the automatic relevance determination (ARD) framework (MacKay, 1995b), as Bishop (1999) has proposed for the probabilistic PCA and PCA mixture models described in sections 2.6.2 and 2.7. Although the computations involved in the Bayesian approach are very complicated and no exact treatment is possible even for the simplest models, this is a promising research direction.

Finally, we show two unusual cases of manifolds. First consider the sample in figure 4.6. Without further information one could say that it corresponds to a two-dimensional manifold on the left side and to a one-dimensional manifold on the right side (whatever that may mean). It actually corresponds to a one-dimensional distribution with variable noise (much higher on the left side), obtained from a speech enhancement application analysed by Xie and van Compernolle (1996). They consider that the observed noisy speech Y is due to independent noise E corrupting the clean speech S additively in the spectral (log) domain, with S and E distributed normally in a frame basis. Thus $10^{\frac{Y}{10}} = 10^{\frac{S}{10}} + 10^{\frac{E}{10}}$ or $y = s + e$, where y, s and e are the log-magnitudes of the observed speech, clean speech and noise, respectively, and s and e follow a log-normal distribution. Fig. 4.6 shows a synthetically generated sample where $S \sim \mathcal{N}(\mu = 10, \sigma = 17)$ and $E \sim \mathcal{N}(\mu = 0, \sigma = 3)$. The solid line corresponds to the minimum mean squared error estimate, the posterior mean $\mathrm{E}\{s|y\}$.

The second case may be unlikely to arise in a practical problem, but nonetheless it has a theoretical interest. Figure 4.7 shows the first 5 approximations to a *space-filling curve*, the Hilbert curve. While each curve is one-dimensional, it can be proven that the limit to which this sequence of curves converges exists and is the square: a two-dimensional manifold. That is, a space-filling curve is a continuous map of an interval of the real line on a rectangle of $\mathbb{R}^2$ or some other higher-dimensional manifold. Other fractal curves, such as the Koch snowflake curve, have even a non-integral (Hausdorff) dimension between 1 and 2 (Barnsley, 1988; Peitgen et al., 1992).

[5] We will not attempt to define formally the concept of dimension, for which, in fact, many different mathematical definitions exist, each one trying to capture some desirable properties of the notion of dimension: topological dimension, covering dimension, Hausdorff dimension, fractal dimension... For our purposes, an intuitive idea of the concept of dimension will be enough. Falconer (1990) has more details about the definition of dimension.

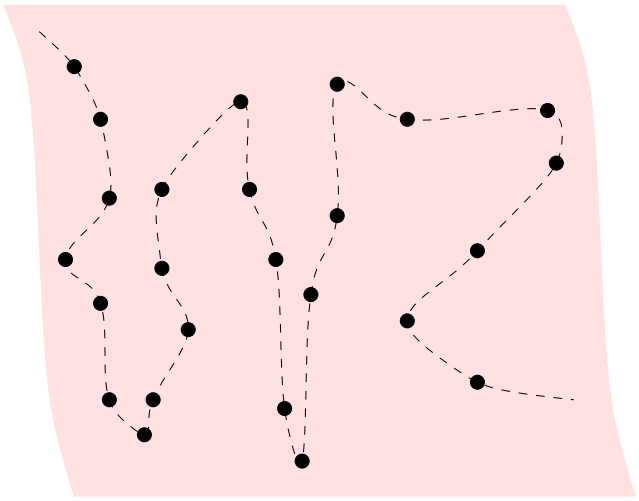


Figure 4.5: Curve or surface?

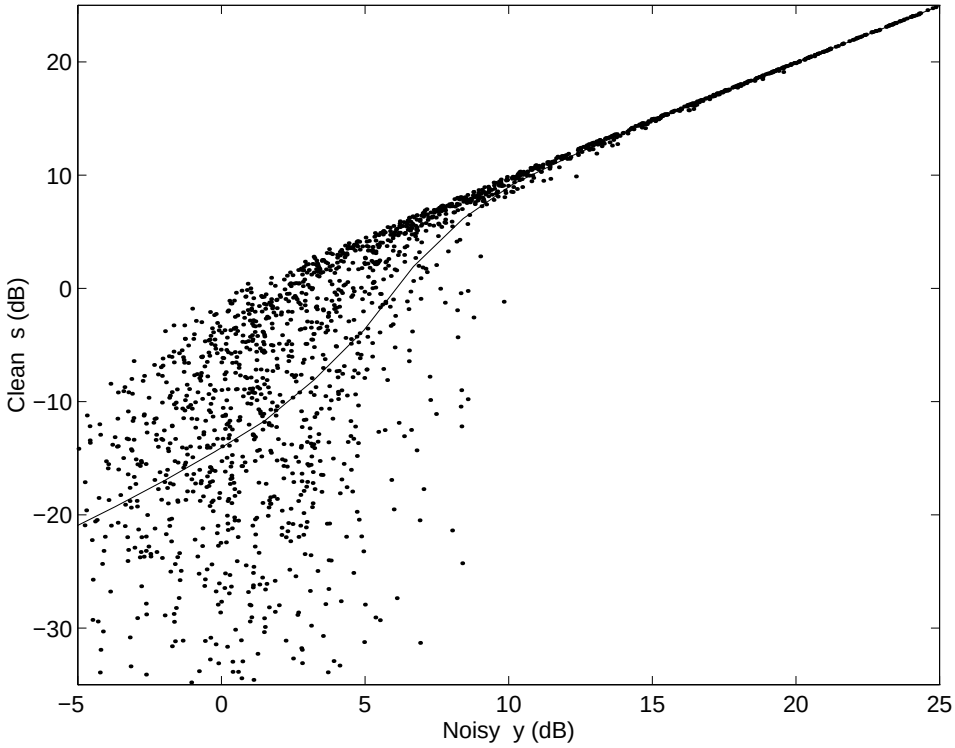


Figure 4.6: What is the intrinsic dimensionality of this sample? (adapted from Xie and van Compernolle, 1996 with permission from Elsevier Science).

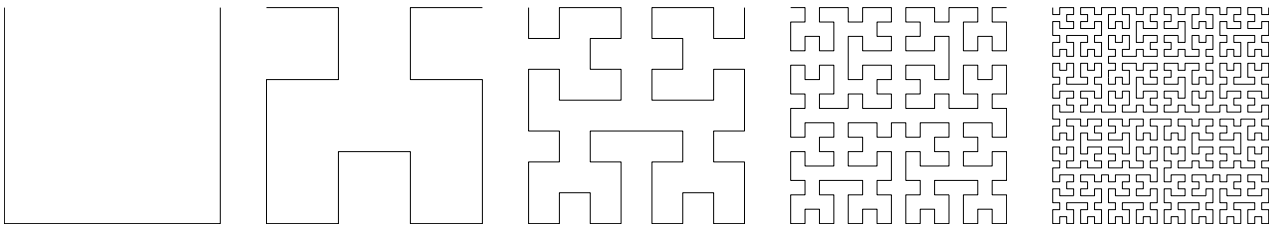


Figure 4.7: First 5 curves of the Hilbert space-filling curve sequence.

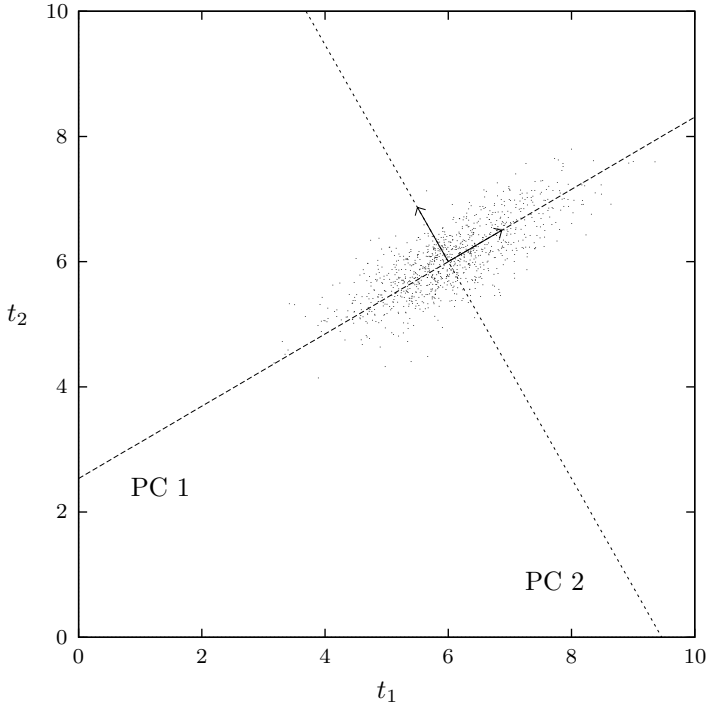


Figure 4.8: Bidimensional, normal point cloud with its principal components.

Generally, we know from set theory that we can map invertibly and continuously $\mathbb{R}^D$ into $\mathbb{R}$ or any interval of $\mathbb{R}$, i.e., $\text{card}\left(\mathbb{R}^D\right) = \text{card}\left(\mathbb{R}\right) = \text{card}\left((0,1)\right) = \mathfrak{c}$ for any $D \in \mathbb{N}$ (Rotman and Kneebone, 1966); for example, using the diagonal Cantor construction: given $\mathbf{t} \in \mathbb{R}^D$, write each of its components $t_1, \ldots, t_D$ as a binary expansion (which can be done in a unique way) and interleave the expansions to obtain the binary expansion of a number in $\mathbb{R}$. In principle, this would allow to find a nonlinear continuous mapping from $\mathbb{R}^D$ into $\mathbb{R}$ preserving all information: exact reduction of any dimensionality to $L = 1$! Of course, due to the finite precision of computers this is of no practical application, and even if it was possible, such a procedure would not help to understand the intrinsic dimensionality in the sense mentioned earlier. $\mathbb{R}^D$ and $\mathbb{R}$ may have the same cardinal, but they do not have the same dimension.

4.5 Principal component analysis

Principal component analysis (PCA) (Jackson, 1991; Jolliffe, 1986) is possibly the dimensionality reduction technique most widely used in practice, perhaps due to its conceptual simplicity, its analytical properties and the fact that relatively efficient algorithms (of polynomial complexity) exist for its computation. In signal processing it is known as the Karhunen-Loève transform.

Traditionally, PCA has been considered as a distribution-free linear dimensionality reduction technique, and that is the point of view that we follow in this section. However, it can be also seen as the maximum likelihood estimate of a specific latent variable model. We describe the probabilistic view of PCA in section 2.6.2.

Let us consider a sample $\{\mathbf{t}_n\}_{n=1}^N$ in $\mathbb{R}^D$ with mean $\overline{\mathbf{t}} \stackrel{\text{def}}{=} \frac{1}{N}\sum_{n=1}^N \mathbf{t}_n$ and covariance matrix $\mathbf{\Sigma} \stackrel{\text{def}}{=} \frac{1}{N}\sum_{n=1}^N (\mathbf{t}_n - \overline{\mathbf{t}})(\mathbf{t}_n - \overline{\mathbf{t}})^T$. The matrix $\mathbf{\Sigma}$ is symmetric semidefinite positive and admits a spectral decomposition

$$\mathbf{\Sigma} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$$

with $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_D)$ orthogonal, $\mathbf{\Lambda} = \text{diag}\,(\lambda_1, \ldots, \lambda_D)$ and $\lambda_d \geq 0$, $d = 1, \ldots, D$. $\mathbf{u}_d$ is the normalised eigenvector of $\mathbf{\Sigma}$ associated with eigenvalue λ_d. The principal component transformation $\mathbf{x} = \mathbf{U}^T(\mathbf{t} - \overline{\mathbf{t}})$ yields a reference system in which the sample has mean $\mathbf{0}$ and diagonal covariance matrix $\mathbf{\Lambda}$ containing the eigenvalues of $\mathbf{\Sigma}$: the variables are now uncorrelated. Figure 4.8 shows an example. In this new reference system one can discard the variables with small variance, i.e., project on the subspace spanned by the first L principal components, and obtain a good approximation (the best linear one in the least squares sense) to the original sample: $\mathbf{x} = \mathbf{U}_L^T(\mathbf{t} - \overline{\mathbf{t}})$ with $\mathbf{U}_L = (\mathbf{u}_1, \ldots, \mathbf{u}_L)$.

The key property of principal component analysis is that it attains the best dimensionality reduction linear map $\mathbf{t} \in \mathbb{R}^D \longrightarrow \mathbf{x} \in \mathbb{R}^L$ in the senses of:

- maximal variance in the projected space subject to orthonormality (Hotelling, 1933):

$$\max_{\mathbf{A}^T\mathbf{A}=\mathbf{I}} \left\{ \operatorname{tr}\left(\frac{1}{N} \sum_{n=1}^{N} \mathbf{x}\mathbf{x}^T \right) \right\} = \sum_{l=1}^{L} \lambda_l \text{ with } \mathbf{x} \stackrel{\text{def}}{=} \mathbf{A}^T(\mathbf{t} - \overline{\mathbf{t}}), \text{ attained at } \mathbf{A} = \mathbf{U}_L$$

- L_2-norm, or least squared sum of errors of the reconstructed data (Pearson, 1901):

$$\min_{\mathbf{A}} \left\{ \sum_{n=1}^{N} \|\mathbf{t} - \mathbf{t}^*\|^2 \right\} = N \sum_{l=1}^{L} \lambda_l \text{ with } \mathbf{t}^* \stackrel{\text{def}}{=} \mathbf{A}\mathbf{A}^T(\mathbf{t} - \overline{\mathbf{t}}), \text{ attained at } \mathbf{A} = \mathbf{U}_L$$

- maximal mutual information (assuming the data vectors $\mathbf{t}$ are distributed normally) between the original vectors $\mathbf{t}$ and their projections $\mathbf{x}$ (Kapur, 1989, pp. 502–504; Cover and Thomas, 1991):

$$\max_{\mathbf{A}} \{I(\mathbf{t}; \mathbf{x})\} = \frac{1}{2} \ln \left(\prod_{l=1}^{L} 2\pi e \lambda_l \right) \text{ with } \mathbf{x} \stackrel{\text{def}}{=} \mathbf{A}^T(\mathbf{t} - \overline{\mathbf{t}}), \text{ attained at } \mathbf{A} = \mathbf{U}_L$$

where $\lambda_1 > \cdots > \lambda_L$ are the first L eigenvalues of the covariance matrix.

Geometrically, the hyperplane spanned by the first L principal components is the regression hyperplane that minimises the orthogonal distances to the data. In this sense, PCA is a symmetric regression approach, as opposed to standard linear regression, which points one component as response variable and the rest as predictors. In fact, the principal component subspace is a principal manifold in the sense of section 4.8.

The first principal components are often used as starting points for other algorithms, such as projection pursuit regression, principal curves, Kohonen's maps or the generalised topographic mapping. PCA is also useful as a first, coarse dimensionality reduction stage where a lot of unnecessary directions of negligible variance are discarded, particularly in very high-dimensional data (for example, when each data vector represents a bitmapped image or a sampled time-varying curve).

A number of numerical techniques exist for finding all or the first few eigenvalues and eigenvectors of a square, symmetric, semidefinite positive matrix (the covariance matrix) in time $\mathcal{O}(D^3)$: singular value decomposition, Cholesky decomposition, etc. (Wilkinson, 1965; Golub and van Loan, 1996; Press et al., 1992). When the covariance matrix, of order $D \times D$, is too large to be explicitly computed one could use neural network techniques (section 4.5.1), some of which do not require more memory space other than the one needed for the data vectors and the principal components themselves. Unfortunately, these techniques (usually based on a gradient descent method) are much slower than traditional methods.

PCA can also be computed from the $N \times N$ scalar-product matrix $\frac{1}{N}\mathbf{T}\mathbf{T}^T$, where $\mathbf{T} \stackrel{\text{def}}{=} (\mathbf{t}_1, \ldots, \mathbf{t}_N)^T$ (assuming centred vectors to simplify the notation). This is because $\frac{1}{N}\mathbf{T}\mathbf{T}^T$ has the same nonzero eigenvalues as $\mathbf{\Sigma} = \frac{1}{N}\mathbf{T}^T\mathbf{T}$, as can be seen from the singular value decomposition of $\mathbf{T}$:

$$\mathbf{T} = \mathbf{V}\mathbf{S}\mathbf{U}^T \Rightarrow \begin{cases} \mathbf{T}\mathbf{T}^T = \mathbf{V}\mathbf{S}\mathbf{S}^T\mathbf{V}^T \\ \mathbf{T}^T\mathbf{T} = \mathbf{U}\mathbf{S}^T\mathbf{S}\mathbf{U}^T \end{cases}$$

where $\mathbf{U}_{D\times D}$ and $\mathbf{V}_{N\times N}$ are orthogonal and $\mathbf{S}_{N\times D}$ has the singular values along its diagonal and zeroes elsewhere. The nonzero eigenvalues of $\mathbf{T}\mathbf{T}^T$ and $\mathbf{T}^T\mathbf{T}$ are the squared singular values. Using the scalar-product matrix instead of the covariance matrix is faster when the number of data points is smaller than the dimensionality, $N < D$, as is often the case in image processing applications, such as face recognition (Sirovich and Kirby, 1987). This is one of the bases of the kernel PCA method described in section 4.5.4.

The disadvantage of PCA is that it is only able to find a linear subspace and thus cannot deal properly with data lying on nonlinear manifolds. When the data is clustered, it can be more convenient to apply PCA locally (section 4.7).

The number of principal components to keep is a tricky question. Some rules of thumb are applied in practice, usually based on the *scree plot* (plot of the cumulative eigenvalue sum versus the number of components), such as to eliminate components whose eigenvalues are smaller than a fraction of the mean eigenvalue, or to keep as many as necessary to explain a certain fraction of the total variance, or to find where the curve

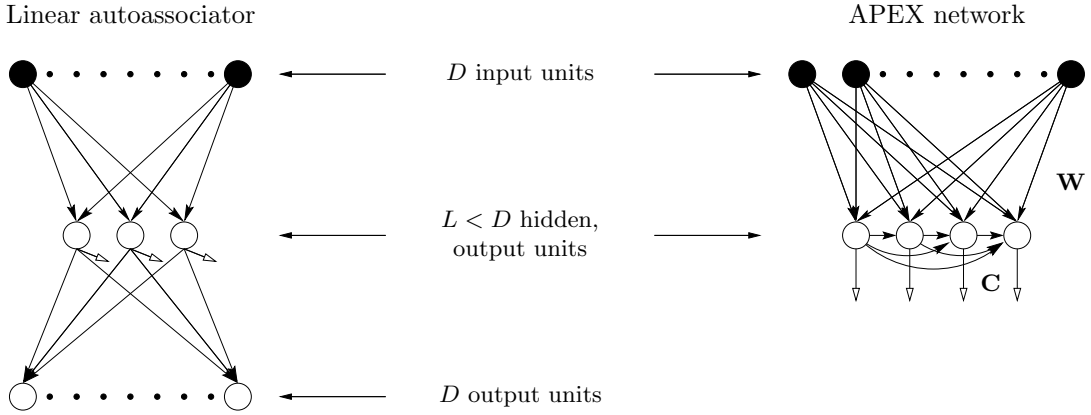


Figure 4.9: Two examples of neural networks able to perform a principal component analysis of its training set: left, a linear autoassociator, trained by backpropagation; right, the APEX network, with Hebbian weights $\mathbf{W}$ and anti-Hebbian, decorrelating weights $\mathbf{C}$. In both cases, the number of hidden units determines how many principal components are kept.

changes its slope abruptly (Jolliffe, 1986). From a statistical goodness of fit point of view, one can also test[6] the hypothesis that the data covariance can be explained by the principal components (H_0) against the alternative that the covariance is unconstrained (H_1), although this procedure does not really tell when to stop adding components (Bartholomew, 1987, pp. 46–47 or Everitt, 1984, pp. 21–22).

The fact is that if the scree plot does not have a sharp decrease at some cutoff number of eigenvalues[7], it is very likely that the data cannot be modelled with a few linear variables, in which case it seems pointless to try hard to obtain the "right" number of eigenvalues and let alone to interpret them (see also section 2.8.1): the model is just wrong. For linear reconstruction purposes PCA is a legitimate technique and the number of eigenvalues L to use can be determined objectively from $\sum_{l=1}^{L} \lambda_l / \sum_{l=1}^{D} \lambda_l \leq 0.9$ (for 10% error, say). As such its use is recommended for (preliminary) feature extraction.

Estimating the intrinsic dimensionality of a data sample using PCA will result in a larger dimensionality than the real one if the data lies on a nonlinear manifold (e.g. a circle is a one-dimensional manifold but it cannot be embedded in a one-dimensional linear manifold). It is also clear that the number of nonzero eigenvalues of the data covariance matrix (with respect to some threshold) gives an upper bound for the intrinsic dimensionality.

4.5.1 Principal component analysis networks

There exist several neural network architectures capable to extract principal components, some of which are represented in fig. 4.9. They can be classified in:

- Autoassociators (also called autoencoders or bottleneck networks), which are linear two-layer perceptrons with D inputs, L hidden units and D outputs, trained to replicate the input in the output layer minimising the squared sum of errors, typically with backpropagation. Bourlard and Kamp (1988) and Baldi and Hornik (1989) showed that this network finds a basis of the subspace spanned by the first L principal components, not necessarily coincident with them.

- Networks based on Oja's rule (Oja, 1992) with some kind of decorrelating device, e.g. the network of Földiák (1989), the APEX network of Kung et al. (1994) or the generalised Hebbian algorithm of Sanger (1989).

Diamantaras and Kung (1996) review PCA networks in detail.

[6] This test was originally developed for factor analysis, but it can be readily extended to PCA via its interpretation as a latent variable model (section 2.6.2).

[7] Also, consider K eigenvalues of slowly decreasing value: $\lambda_i \geq \lambda_{i+1} \geq \cdots \geq \lambda_{i+K-1}$. They determine a slightly elongated K-dimensional ellipsoid with semiaxes of lengths $\sqrt{\lambda_i}, \ldots, \sqrt{\lambda_{i+K-1}}$ in the K-dimensional subspace defined by their associated eigenvectors. Any direction in that subspace will have a variance in $[\lambda_{i+K-1}, \lambda_i]$ and thus all directions in that subspace (not just along the eigenvectors) are approximately equivalent (in the extreme case where $\lambda_i = \cdots = \lambda_{i+K-1}$ they define a hypersphere and all directions are strictly equivalent).

4.5.2 Nonlinear autoassociators

An obvious extension to linear autoassociators is the inclusion of nonlinear activation functions and several layers (see fig. 4.10). The representation obtained in the unit activations of one of the hidden layers (with $L < D$ units) can be taken as the reduced-dimension representative (the middle layer in the figure). This defines a dimensionality reduction mapping $\mathbf{F}$ and a reconstruction mapping $\mathbf{f}$ (sometimes called recognition and generative mappings, respectively). As in the linear case, the net is trained to replicate its input at the output layer in the least-squares sense by backpropagation or other method. Given their conceptual simplicity and the appeal of the idea of "squashing the input through a bottleneck," nonlinear autoassociators were used for dimensionality reduction quite early, e.g. by Saund (1989), Fleming and Cottrell (1990), Kramer (1991) or DeMers and Cottrell (1993), among others.

Bourlard and Kamp (1988) show that nonlinear autoassociators with only one hidden layer are no better than linear ones, i.e., than PCA. But clearly, nonlinear autoassociators with three hidden layers (as in fig. 4.10) must have, at least potentially, superior ability than linear ones for dimensionality reduction, since both $\mathbf{F}$ and $\mathbf{f}$ become universal approximators (Scarselli and Tsoi, 1998). Indeed, they have outperformed PCA in some applications. Malthouse (1998) cites several of them, particularly in chemometrics, where nonlinear autoassociators were popularised by Kramer (1991). Surprisingly, the approach has not been widely accepted as a dimensionality reduction method. Several reasons have been proposed for this from an empirical perspective:

- Nonlinear autoassociators are very slow to train. Rögnvaldsson (1994) has offered the following explanation for this: the risk that the Hessian of the error function of a multilayer perceptron is ill-conditioned grows with the number of layers. An ill-conditioned Hessian makes the error surface very flat and learning becomes very slow both with backpropagation and with second-order methods.

- Training with various local optimisers (e.g. gradient descent, conjugate gradient descent, stochastic gradient descent or a quasi-Newton method) very often results in local minima with a higher error than PCA (Kambhatla and Leen, 1997). Using neuronal spike trains data, Fotheringhame and Baddeley (1997) observed backpropagation to be slow and unreliable, while conjugate gradient descent worked better than PCA with synthetic data but not with real data.

- Several researchers, e.g. Fleming and Cottrell (1990), have observed that, at the end of the training (with backpropagation), the unit activations often concentrate on the linear region of the nonlinearity and therefore there is little difference with PCA networks.

These reasons seem to point to deficiencies in the optimisation algorithm rather than in the class of representations attainable. That is, a nonlinear autoassociator is potentially able to represent complex manifolds, but, for most initial values of the parameters, a local optimiser will end in a bad local minimum rather than in one of the good ones.

Some theoretical results are known for nonlinear autoassociators. Both the dimensionality reduction mapping $\mathbf{F}$ and the reconstruction mapping $\mathbf{f}$ are continuous if the unit nonlinearities are continuous (as is often the case, e.g. with the sigmoid). Malthouse (1998) uses this fact to show that nonlinear autoassociators can approximate neither self-intersecting manifolds nor discontinuous manifolds. He also notices that they cannot implement a dimensionality reduction mapping based on orthogonal projection on the closest point of the manifold (as does happen with principal curves, section 4.8): this would lead to a discontinuity in the dimensionality reduction mapping for those data points which are equidistant from different points of the hidden-layer manifold.

Nonlinear autoassociators have had some empirical success in other kinds of pattern recognition problems. Wiles et al. (1996) report finding good solutions to the travelling salesman problem[8], while Japkowicz et al. (2000) claim better performance than with linear autoassociators for classification in nonlinear multimodal domains.

4.5.3 Other linear transformations

Here we mention other usual linear transformations of the data, their relation to PCA and their effect on the covariance matrix. Consider again the sample $\{\mathbf{t}_n\}_{n=1}^N$ in $\mathbb{R}^D$ with mean $\overline{\mathbf{t}} \stackrel{\text{def}}{=} \frac{1}{N}\sum_{n=1}^N \mathbf{t}_n$ and covariance matrix $\mathbf{\Sigma} \stackrel{\text{def}}{=} \frac{1}{N}\sum_{n=1}^N (\mathbf{t}_n - \overline{\mathbf{t}})(\mathbf{t}_n - \overline{\mathbf{t}})^T$. Let $\mathbf{A}_{D\times L} = (\mathbf{a}_1, \dots, \mathbf{a}_L)$ with $\mathbf{a}_l \in \mathbb{R}^D$ a set of L projection directions.

[8] Dimensionality reduction here is conceptualised as going from the D-dimensional map of D cities (coded as 1-of-D) to a one-dimensional ordered list in which neighboring cities are listed close together.

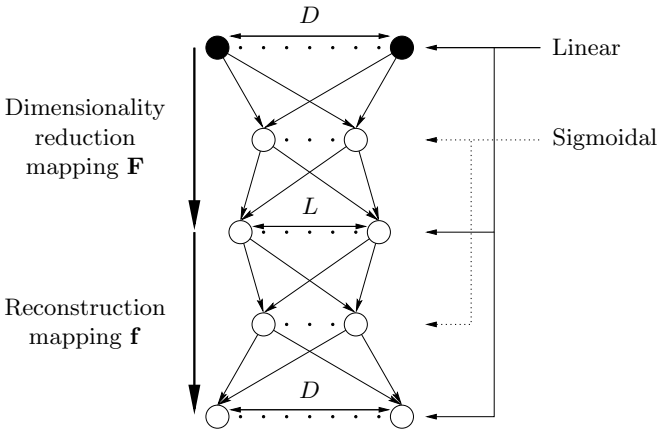


Figure 4.10: Nonlinear autoassociator, implemented as a four-layer nonlinear perceptron where $L < D$ and $\mathbf{t}^* \stackrel{\text{def}}{=} \mathbf{f}(\mathbf{F}(\mathbf{t}))$.

Transformation f	$\mathbf{t}' = f(\mathbf{t})$	$\overline{\mathbf{t}}'$	$\mathbf{\Sigma}'$	$\mathbf{U}'$	$\mathbf{u}'$	$\mathbf{\Lambda}'$	λ'
Translation	$\mathbf{t}+\mathbf{d}$	$\overline{\mathbf{t}}+\mathbf{d}$	$\mathbf{\Sigma}$	$\mathbf{U}$	same	$\mathbf{\Lambda}$	same
Rotation	$\mathbf{R}\mathbf{t}$	$\mathbf{R}\overline{\mathbf{t}}$	$\mathbf{R}\mathbf{\Sigma}\mathbf{R}^T$	$\mathbf{R}\mathbf{U}$	rotated	$\mathbf{\Lambda}$	same
Axis scaling	$\mathbf{D}\mathbf{t}$	$\mathbf{D}\overline{\mathbf{t}}$	$\mathbf{D}\mathbf{\Sigma}\mathbf{D}$	$\neq$	$\neq$	$\neq$	$\neq$
Uniform axis scaling	$a\mathbf{t}$	$a\overline{\mathbf{t}}$	$a^2\mathbf{\Sigma}$	$\mathbf{U}$	same	$a^2\mathbf{\Lambda}$	scaled
Affine	$a\mathbf{t}+\mathbf{d}$	$a\overline{\mathbf{t}}+\mathbf{d}$	$a^2\mathbf{\Sigma}$	$\mathbf{U}$	same	$a^2\mathbf{\Lambda}$	scaled
Centring	$\mathbf{t}-\overline{\mathbf{t}}$	$\mathbf{0}$	$\mathbf{\Sigma}$	$\mathbf{U}$	same	$\mathbf{\Lambda}$	same
PCA	$\mathbf{U}^T(\mathbf{t}-\overline{\mathbf{t}})$	$\mathbf{0}$	$\mathbf{\Lambda}$	$\mathbf{I}$	$\{\mathbf{e}_d\}_{d=1}^D$	$\mathbf{\Lambda}$	same
Sphering	$\mathbf{\Sigma}^{-1/2}(\mathbf{t}-\overline{\mathbf{t}})$	$\mathbf{0}$	$\mathbf{I}$	$\mathbf{I}$	$\{\mathbf{e}_d\}_{d=1}^D$	$\mathbf{I}$	1

Table 4.2: Transformation of the covariance matrix under transformations on the sample. $\overline{\mathbf{t}}$ is the sample mean and $\mathbf{\Sigma}$ the sample covariance, with spectral decomposition $\mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$. $a \in \mathbb{R}-\{0\}$, $\mathbf{d} \in \mathbb{R}^D$, $\mathbf{D} = \text{diag}\,(d_1, \ldots, d_D)$ is diagonal, $\mathbf{R}$ is orthogonal, $\mathbf{e}_d$ is the unit vector in the direction of t_d, the primes denote the new entity after the transformation f and the symbol $\neq$ is used to indicate that the subsequent transformation is complex (not obviously related to f).

Centring is a procedure by which we translate the sample so that its mean is at the origin: $\mathbf{t}' = \mathbf{t}-\overline{\mathbf{t}} \Rightarrow \overline{\mathbf{t}}' = \mathbf{0}$. Centring is inherited by any set of projections: $\text{E}\{\mathbf{t}\} = \mathbf{0} \Rightarrow \text{E}\{\mathbf{A}^T\mathbf{t}\} = \mathbf{A}^T\,\text{E}\{\mathbf{t}\} = \mathbf{0}$.

Scaling achieves unit variance in each axis by dividing componentwise by its standard deviation $\sigma_d \stackrel{\text{def}}{=} \sqrt{(\mathbf{\Sigma})_{dd}}$: $\mathbf{t}' = \text{diag}\left(\sigma_1^{-1}, \ldots, \sigma_D^{-1}\right)\mathbf{t} \Rightarrow (\mathbf{\Sigma}')_{dd} = 1\ \forall d = 1, \ldots, D$.

Sphering is an affine transformation that converts the covariance matrix (of the centred sample) into a unit variance matrix, thus destroying all the first- and second-order information of the sample: $\mathbf{t}' = \mathbf{\Sigma}^{-1/2}(\mathbf{t}-\overline{\mathbf{t}}) \Rightarrow \overline{\mathbf{t}}' = \mathbf{0}$ and $\mathbf{\Sigma}' = \mathbf{I}$, where[9] $\mathbf{\Sigma}^{-1/2} = \mathbf{\Lambda}^{-1/2}\mathbf{U}^T$. Sphering is inherited by any orthogonal set of projections: $\text{E}\{\mathbf{t}\} = \mathbf{0}$ and $\text{cov}\{\mathbf{t}\} = \mathbf{I} \Rightarrow \text{cov}\left\{\mathbf{A}^T\mathbf{t}\right\} = \text{E}\left\{(\mathbf{A}^T\mathbf{t})(\mathbf{A}^T\mathbf{t})^T\right\} = \mathbf{A}^T\,\text{E}\left\{\mathbf{t}\mathbf{t}^T\right\}\mathbf{A} = \mathbf{A}^T\mathbf{A} = \mathbf{I}$.

PCA is another affine transformation that converts the covariance matrix (of the centred sample) into a diagonal matrix, thus decorrelating the variables but preserving the variance information: $\mathbf{t}' = \mathbf{U}^T(\mathbf{t}-\overline{\mathbf{t}}) \Rightarrow \overline{\mathbf{t}}' = \mathbf{0}$, $\mathbf{\Sigma}' = \mathbf{\Lambda}$.

PCA and sphering are both translation and rotation invariant, i.e., applying a translation and a rotation to the data and then performing PCA or sphering produces the same results as performing them on the original data. Table 4.2 summarises the effect of linear transformations on the covariance matrix.

[9] $\mathbf{\Sigma}^{-1/2}$ is defined as a matrix $\mathbf{B}$ such that $\mathbf{B}^T\mathbf{B} = \mathbf{\Sigma}$ and so any matrix $\mathbf{B}' = \mathbf{R}\mathbf{B}$ with $\mathbf{R}$ orthogonal is also valid. Canonically, we can take $\mathbf{\Sigma}^{-1/2} = \mathbf{\Lambda}^{-1/2}\mathbf{U}^T$.

4.5.4 Kernel PCA

Kernel PCA (Schölkopf et al., 1998) is an unsupervised feature extraction method closely related to PCA[10] that, given a data set $\{\mathbf{t}_n\}_{n=1}^N$ contained in an *input space* $\mathcal{T} \subset \mathbb{R}^D$, extracts up to $\max(N, D)$ features from a vector $\mathbf{t} \in \mathcal{T}$. It is based on the following ideas:

- Carrying out PCA on the dot-product matrix of the data points, as mentioned in section 4.5, which is an $N \times N$ symmetric matrix of rank smaller or equal than $\min(N, D)$.
- Nonlinearly mapping input vectors to a high-dimensional *feature space* $\mathcal{F}$, $\mathbf{\Phi} : \mathcal{T} \longrightarrow \mathcal{F}$, where standard PCA is performed—this requires dot products in $\mathcal{F}$. Each component of $\mathbf{\Phi}$ will be a particular real function of the variables $t_1, \ldots, t_D$ and will give information on a particular relationship between those variables. To be able to account for many different relationships, the dimensionality of $\mathcal{F}$ will be extremely high, possibly a power of the dimensionality of $\mathcal{T}$. Standard PCA is recovered by taking $\mathbf{\Phi}$ as the identity function.
- Computing dot products in $\mathcal{F}$ via a kernel function in $\mathcal{T}$, $k : \mathcal{T} \times \mathcal{T} \longrightarrow \mathbb{R}$. This allows to perform PCA efficiently on the data set $\{\mathbf{\Phi}(\mathbf{t}_n)\}_{n=1}^N$ using the dot-product Gram matrix[11] $\mathbf{K} \stackrel{\text{def}}{=} \left(\mathbf{\Phi}(\mathbf{t}_m)^T \mathbf{\Phi}(\mathbf{t}_n)\right)_{mn}$ but never explicitly calculating $\mathbf{\Phi}$ for any point. The nonlinear nature of map $\mathbf{\Phi}$ means that the associated component analysis back in input space $\mathcal{T}$ is nonlinear.

Let us analyse more in detail the procedure:

PCA in feature space is easily seen to require the solution of the eigenvalue problem $N\lambda\mathbf{a} = \mathbf{K}\mathbf{a}$ for eigenvectors $\mathbf{a}$ and nonzero eigenvalues λ, which results in up to N nonzero eigenvalues and associated eigenvectors, normalised such that $\|\mathbf{a}_n\| = \lambda_n^{-1/2}$. This ensures that $\mathbf{v}_n \stackrel{\text{def}}{=} \sum_{m=1}^N a_{nm} \mathbf{\Phi}(\mathbf{t}_m)$ is a unit-norm eigenvector of the sample covariance matrix in feature space, $\mathbf{\Sigma}_{\mathcal{F}} \stackrel{\text{def}}{=} \frac{1}{N} \sum_{n=1}^N \mathbf{\Phi}(\mathbf{t}_n)\mathbf{\Phi}(\mathbf{t}_n)^T$. Orthogonal projection in $\mathcal{F}$ of a vector $\mathbf{\Phi}(\mathbf{t})$ on the principal component directions $\{\mathbf{v}_n\}_{n=1}^N$ is accomplished as usual via the scalar product and results in an associated set of nonlinear components in $\mathcal{T}$: $\mathbf{v}_n^T \mathbf{\Phi}(\mathbf{t})$ is the orthogonal projection of vector $\mathbf{\Phi}(\mathbf{t})$ on the nth principal component in $\mathcal{F}$, or the nonlinear projection of input vector $\mathbf{t}$ on the nth nonlinear component in $\mathcal{T}$.

Dot products in feature space Defining a kernel function $k(\mathbf{t}_1, \mathbf{t}_2) \stackrel{\text{def}}{=} \mathbf{\Phi}(\mathbf{t}_1)^T \mathbf{\Phi}(\mathbf{t}_2)$ creates a correspondence between kernel functions $k : \mathcal{T} \times \mathcal{T} \longrightarrow \mathbb{R}$ and nonlinear maps $\mathbf{\Phi} : \mathcal{T} \longrightarrow \mathcal{F}$. Rather than choosing $\mathbf{\Phi}$ and then defining k, which is straightforward but does not give an efficient form for k, one chooses k, which then implicitly defines $\mathbf{\Phi}$. But not every function k can be expressed as a dot product in some space $\mathcal{F}$. A way to select kernels is given by Mercer's theorem of functional analysis (Courant and Hilbert, 1953), which states a necessary condition for a kernel to be expressible as a dot product:

Theorem 4.5.1 (Mercer). *If k is the continuous kernel of an integral operator $K : \mathcal{L}_2 \longrightarrow \mathcal{L}_2$, $Kf(\mathbf{t}_2) \stackrel{\text{def}}{=} \int_{\mathcal{T}} k(\mathbf{t}_1, \mathbf{t}_2) f(\mathbf{t}_1)\, d\mathbf{t}_1$, which is positive, $\int_{\mathcal{T}\times\mathcal{T}} f(\mathbf{t}_1) k(\mathbf{t}_1, \mathbf{t}_2) f(\mathbf{t}_2)\, d\mathbf{t}_1\, d\mathbf{t}_2 \geq 0 \ \forall f \in \mathcal{L}_2$, then $k(\mathbf{t}_1, \mathbf{t}_2) = \sum_{n=1}^\infty \lambda_n \psi_n(\mathbf{t}_1)\psi_n(\mathbf{t}_2)$ with $\lambda_n \geq 0 \ \forall n$.*

Such kernels are called Mercer or reproducing kernels. We can then define $\mathbf{\Phi}(\mathbf{t}) \stackrel{\text{def}}{=} \left(\sqrt{\lambda_1}\psi_1(\mathbf{t}), \sqrt{\lambda_2}\psi_2(\mathbf{t}), \ldots\right)^T$ and so $k(\mathbf{t}_1, \mathbf{t}_2) = \mathbf{\Phi}(\mathbf{t}_1)^T \mathbf{\Phi}(\mathbf{t}_2)$. Table 4.3 gives examples of such kernels, all of which have also been used in support vector machines (Schölkopf et al., 1999a; Cristianini and Shawe-Taylor, 2000). The $\mathbf{\Phi}(\mathbf{t})$ mapping associated with the polynomial kernel is the collection of all possible pth degree ordered products of components of $\mathbf{t}$; e.g. for $p = 2$ we have $\mathbf{t} = (t_1\ t_2)^T$ and $k(\mathbf{t}_1, \mathbf{t}_2) = (\mathbf{t}_1^T \mathbf{t}_2)^2 = \mathbf{\Phi}(\mathbf{t}_1)^T \mathbf{\Phi}(\mathbf{t}_2)$ with $\mathbf{\Phi}(\mathbf{t}) = (t_1^2\ t_2^2\ t_1 t_2\ t_2 t_1)^T$. As an application, each t_d could be a pixel in a bitmapped image.

Evidently, the usual properties of PCA (section 4.5) remain in feature space: orthogonal directions of maximal variance, minimal L_2-reconstruction error and maximal mutual information with respect to the inputs (under Gaussian assumptions). But not anymore in input space, where the procedure is nonlinear. Kernel PCA will be unitary invariant (i.e., independent of the orthogonal coordinates used) if the kernel depends only on dot products and/or distances (as those of table 4.3 do), since the only operations on the input vectors that the whole procedure involves are computing the kernel values.

[10] In this section, we will use interchangeably the terms PCA, standard PCA or usual PCA to mean the linear PCA of section 4.5.

[11] The formula for $\mathbf{K}$ assumes centred vectors in $\mathcal{F}$. A corrected expression for uncentred vectors that does not explicitly compute $\mathbf{\Phi}$ at any point is easily derived from the one given.

Kernel	$k(\mathbf{t}_1, \mathbf{t}_2)$
Polynomial	$(\mathbf{t}_1^T \mathbf{t}_2)^p$
Gaussian (radial basis function)	$e^{-\frac{\|\mathbf{t}_1 - \mathbf{t}_2\|^2}{2\sigma^2}}$
Sigmoid	$\tanh(a\mathbf{t}_1^T \mathbf{t}_2 + b)$

Table 4.3: Examples of Mercer kernel functions for kernel PCA and support vector machines. In the polynomial kernel, $p = 1$ gives standard PCA.

Kernel PCA requires N computations of the kernel function where standard PCA required a dot product, which is not a significant extra cost. The real problem appears with large data sets ($N \gg D$: the normal situation in most applications) since the dot product matrix $\mathbf{K}_{N \times N}$ becomes huge. In this case, standard acceleration tricks can be applied, such as extracting only the first few components or estimating $\mathbf{K}$ from a small subset of the data.

For dimensionality reduction purposes, the projections on the principal components can be taken as features. While standard PCA can extract up to $\min(N, D)$ or fewer nonnull features (the rank of the sample covariance matrix in $\mathcal{T}$-space, $\mathbf{\Sigma}$), kernel PCA can obtain up to N, since in $\mathcal{F}$-space the rank of the dot product matrix $\mathbf{K}$ can be up to N if $\mathbf{\Phi}$ is nonlinear. A natural definition of reconstruction would be based on the usual orthogonal projection on the first principal components in feature space, but in general not every point in the subspace spanned by those principal components will have a preimage in input space! That is, even though $\{\mathbf{\Phi}(\mathbf{t}_n)\}_{n=1}^N$ have by definition preimages $\{\mathbf{t}_n\}_{n=1}^N$, a point $\mathbf{v} \in \operatorname{span}\left\{\{\mathbf{\Phi}(\mathbf{t}_n)\}_{n=1}^N\right\}$ may not have one $\mathbf{t} \in \mathcal{T}$ with $\mathbf{\Phi}(\mathbf{t}) = \mathbf{v}$. This requires to define an approximate (in some sense) reconstruction mapping in input space. Schölkopf et al. (1999b) give an algorithm to find a close preimage in the L_2 sense. They approximate $\mathbf{v} = \sum_{n=1}^N a_n \mathbf{\Phi}(\mathbf{t}_n)$ by a multiple (for computational considerations) of a vector $\mathbf{\Phi}(\mathbf{t})$ on the image of the input space:

$$\min_{\mathbf{t} \in \mathcal{T}} \left\| \frac{\mathbf{v}^T \mathbf{\Phi}(\mathbf{t})}{\mathbf{\Phi}(\mathbf{t})^T \mathbf{\Phi}(\mathbf{t})} \mathbf{\Phi}(\mathbf{t}) - \mathbf{v} \right\| \Longleftrightarrow \max_{\mathbf{t} \in \mathcal{T}} \frac{(\mathbf{v}^T \mathbf{\Phi}(\mathbf{t}))^2}{\mathbf{\Phi}(\mathbf{t})^T \mathbf{\Phi}(\mathbf{t})}.$$

The maximisation can be carried out with standard methods or, for kernels that satisfy $k(\mathbf{t}_1, \mathbf{t}_2) = \kappa(\|\mathbf{t}_1 - \mathbf{t}_2\|^2)$ (e.g. the Gaussian kernel), with a fixed-point iteration method: taking the gradient of $(\mathbf{v}^T \mathbf{\Phi}(\mathbf{t}))^2$ with respect to $\mathbf{t}$ and equating to 0 results in

$$\mathbf{t}^{(\tau+1)} = \frac{\sum_{n=1}^N a_n \kappa'(\|\mathbf{t}_n^{(\tau)} - \mathbf{v}\|) \mathbf{t}_n^{(\tau)}}{\sum_{n=1}^N a_n \kappa'(\|\mathbf{t}_n^{(\tau)} - \mathbf{v}\|)}$$

which for the Gaussian kernel is formally identical to the fixed-point iteration scheme we give in eq. (8.4) for finding the modes of a Gaussian mixture.

Experimentally, Schölkopf et al. (1998) show that kernel PCA outperforms standard PCA as feature extraction preprocessor for some classification tasks over a wide range of polynomial kernels and that the nonlinear components can be interpreted as separating or splitting clusters in a toy problem. Using approximate preimages, Schölkopf et al. (1999b) show improvements over PCA when denoising patterns if a large number of nonlinear components are used. The intuitive explanation they propose is that, since kernel PCA can extract more components (up to N) than PCA (up to D), it can provide a larger number of components that carry information about the structure in the data before they start to carry noise information as well (which always happens for high-order components). This would seem counterproductive for dimensionality reduction purposes, though.

In summary, kernel PCA has the following advantages:

- We obtain nonlinear components without any nonlinear optimisation, just computing standard PCA.
- We can extract up to N components, which is usually much more than what PCA allows (D if $N > D$), although this may not be conducing to dimensionality reduction. As in PCA, we do not need to restart the procedure if we want more or less components.
- For feature extraction, it can be readily used wherever PCA is.

And the following disadvantages:

- The procedure is sensitive to the kernel used (with different kernels resulting in different performances) but we do not know a priori what kernel to use. Thus, while kernel PCA is free from falling in local minima (a ubiquitous problem with methods based on nonlinear optimisation), we have a whole space of kernel functions to explore.
- While in feature space the geometrical interpretation of the principal components as orthogonal directions of maximal variance remains, in general and a priori we do not really know what the first components geometrically are in input space.
- For large data sets ($N \gg D$) the algorithm must be approximated to limit its computational requirements.
- There is an uncomfortable lack of natural dimensionality reduction and reconstruction mappings due to the fact that the principal component subspace in feature space may not have preimages in input space.

4.6 Projection pursuit

Principal component analysis selects linear projections of the data according to maximal variance subject to orthogonality. In general, one can search for projections that satisfy other properties. This is the basis of projection pursuit (Friedman and Tukey, 1974; Huber, 1985; Jones and Sibson, 1987; Ripley, 1996). Projection pursuit is an unsupervised technique that picks *interesting* low-dimensional linear orthogonal projections of a high-dimensional point cloud by optimising a certain objective function called **projection index**. It is typically used in exploratory data analysis to take profit of the human ability to discover patterns in low-dimensional (1- to 3D) projections: clustering, skewness, kurtosis, concentration along nonlinear manifolds, etc. That is, it is mainly used for visualisation; but it is equally useful for dimensionality reduction and regression, as shown below.

The (scaled) variable loadings (components of the projection vectors) that define the corresponding solution indicate the relative strength that each variable contributes to the observed effect. As in factor analysis, applying a rotation or similar transformations to the projections will produce the same picture but with an easier interpretation of the variable loadings.

Projections are smoothing operations in that structure can be obscured but never enhanced: any structure seen in a projection is a shadow of an actual structure in the original space. It is of interest to pursue the sharpest projections, that will reveal most of the information contained in the high-dimensional data distribution. We consider that a projection is interesting if it contains structure. Structure is considered as departure from normality, since:

- For fixed variance, the normal distribution has the least information, in both the senses of Fisher information and negative entropy (Cover and Thomas, 1991).
- For most high-dimensional clouds, most low-dimensional projections are approximately normal (Diaconis and Freedman, 1984).

Therefore, the normal distribution is the least structured (or least interesting) density.

For example, figure 4.11 shows two 2D projections of a 3D data set consisting of two clusters. The projection on the plane spanned by $\mathbf{e}_2$ and $\mathbf{e}_3$ is not very informative, as both clusters confuse in one; this projection nearly coincides with the one in the direction of the first principal component, which proves that the projection index of PCA (maximal variance; see section 4.6.1) is not a good indicator of structure. However, the projection on the plane spanned by $\mathbf{e}_1$ and $\mathbf{e}_2$ clearly shows both clusters.

The projection pursuit procedure will tend to identify outliers because the presence of the latter in a sample gives it the appearance of nonnormality. This sometimes can obscure the clusters or other interesting structure being sought. Also, the sample covariance matrix is strongly influenced by extreme outliers, so that methods relying on it (e.g. through data sphering) will not be robust against outliers. The effect of outliers can be partially tackled by robust sphering, e.g. using a trimming method (where all observations that lie farther than a threshold from the mean are deleted).

4.6.1 The projection index

A **projection index** I is a real functional on the space of distributions on $\mathbb{R}^L$:

$$I: \; p \in L_2(\mathbb{R}^L) \longrightarrow I(p) \in \mathbb{R}.$$

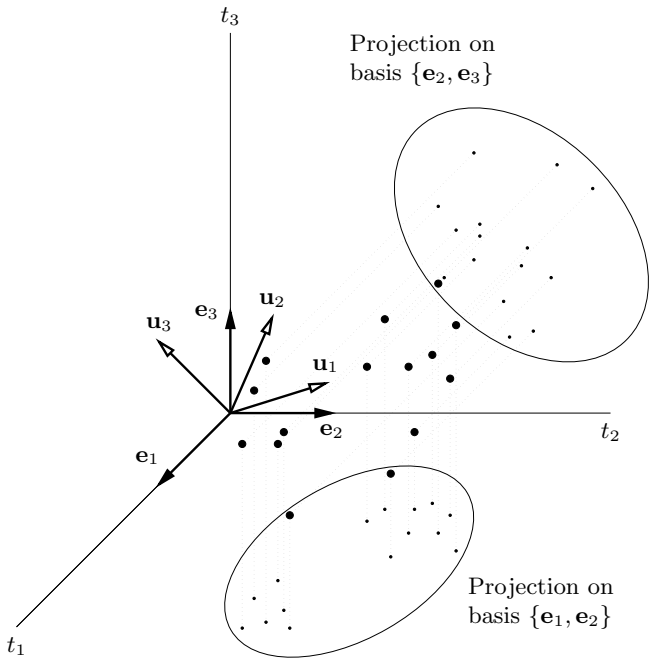


Figure 4.11: Two-dimensional projections of a three-dimensional data set. $\{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3\}$ are the principal component directions.

p is the pdf of an L-dimensional random variable $\mathbf{x} \stackrel{\text{def}}{=} \mathbf{A}^T\mathbf{t}$, itself the projection (of matrix $\mathbf{A}_{D\times L}$) of a D-dimensional random variable $\mathbf{t}$. Abusing of notation, we will write $I(\mathbf{x})$ instead of $I(p)$. We will consider only one-dimensional projections for simplicity, but the treatment is easily extendable to multidimensional projections.

Projection pursuit attempts to find projection directions for a given distribution which produce local maxima of I. To make the maximisation problem independent of the length of the projection vectors and to obtain uncorrelated directions, the directions are constrained to be unit length and mutually orthogonal (i.e., the column vectors of $\mathbf{A}$ must be orthonormal). The optimisation problem is then

$$\max I(\mathbf{A}) \text{ subject to } \mathbf{A}^T\mathbf{A} = \mathbf{I}.$$

In general, there will be several interesting projections, each showing different insight, which correspond to local optima of the projection index. In fact, a difficulty with many projection pursuit indices that has often been observed (Friedman, 1987; Ripley, 1996) is the existence of many local maxima, in particular with small data sets, which makes finding good, high maxima difficult with local optimisers.

Huber (1985) lists the following properties that a good projection index should satisfy:

- Have continuous first (at least) derivative, to allow the use of gradient methods.
- Be rapidly computable—as well as its derivative(s)—since the optimisation procedure requires evaluating it many times.
- Be invariant to all nonsingular affine transformations in the data space, to discover structure not captured by the correlation.
- Satisfy: $I(\mathbf{x}_1 + \mathbf{x}_2) \leq \max(I(\mathbf{x}_1), I(\mathbf{x}_2))$ because, by the central limit theorem, $\mathbf{x}_1 + \mathbf{x}_2$ must be more normal (less interesting) than the less normal of $\mathbf{x}_1$ and $\mathbf{x}_2$. It follows that $I(\mathbf{x}_1 + \cdots + \mathbf{x}_N) \leq I(\mathbf{x})$ if $\mathbf{x}_1, \ldots, \mathbf{x}_N$ are copies of $\mathbf{x}$, and therefore $I(\mathcal{N}) \leq I(\mathbf{x})$ if $\mathcal{N}$ is normal.

4.6.1.1 Examples of projection indices

Consider a random variable $\mathbf{t}$ of expectation $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$.

- Average:

$$I(\mathbf{x}) \stackrel{\text{def}}{=} \mathrm{E}\{\mathbf{x}\}.$$

In this case $\max_{\|\mathbf{a}\|=1} I(\mathbf{a}^T\mathbf{t}) = \|\boldsymbol{\mu}\|$ for $\mathbf{a} = \boldsymbol{\mu}/\|\boldsymbol{\mu}\|$.

- Variance:

$$I(\mathbf{x}) \stackrel{\text{def}}{=} \text{var}\{\mathbf{x}\}.$$

In this case $\max_{\|\mathbf{a}\|=1} I(\mathbf{a}^T\mathbf{t}) = \lambda_1$ for $\mathbf{a} = \mathbf{u}_1$, the largest eigenvalue of $\boldsymbol{\Sigma}$ and its associated normalised eigenvector, respectively. In other words, this index finds the first principal component. PCA is then a particular case of projection pursuit.

- Standardised absolute cumulants $k_m(\mathbf{x})$ (defined in section A.2):

$$I(\mathbf{x}) \stackrel{\text{def}}{=} \frac{|k_m(\mathbf{x})|}{k_2(\mathbf{x})^{m/2}} \qquad m > 2.$$

- Fisher information:

$$I(\mathbf{x}) \stackrel{\text{def}}{=} J(\boldsymbol{\theta}) \stackrel{\text{def}}{=} \text{E}\left\{\left(\frac{\partial p(\mathbf{x};\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right)\left(\frac{\partial p(\mathbf{x};\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right)^T\right\}$$

where $p(\mathbf{x};\boldsymbol{\theta})$ depends on some parameters $\boldsymbol{\theta}$. $J(\boldsymbol{\theta} = \text{cov}\{\mathbf{x}\})$ is minimised by the normal density (Huber, 1985).

- Negative Shannon entropy:

$$I(\mathbf{x}) \stackrel{\text{def}}{=} -h(\mathbf{x}) = \text{E}\{\ln p(\mathbf{x})\} = \int p(\mathbf{x}) \ln p(\mathbf{x})\, d\mathbf{x}.$$

For fixed variance, this index is minimised by the normal density (Cover and Thomas, 1991). Its drawback is that it is difficult to compute, so that several approximations have been proposed. Jones and Sibson (1987) propose two ways to evaluate it:

  - By numerical integration or implementing it as a sample entropy: $\frac{1}{N}\sum_{n=1}^{N} \ln p(\mathbf{a}^T\mathbf{t}_n)$, where p is a nonparametric density estimate of the projected points $\mathbf{a}^T\mathbf{t}_n$. Both are very slow to compute.
  - Approximating p by an expansion in terms of the cumulants (skew and kurtosis); for 1D projections:

$$\int p \ln p \approx \frac{1}{12}\left(k_3^2 + \frac{1}{4}k_4^2\right).$$

Cumulant-based approximations to the differential entropy arise from a Gram-Charlier or Edgeworth polynomial expansion of the density (Kendall and Stuart, 1977) and have been also used in the context of independent component analysis (Comon, 1994). However, they give a poor approximation to the entropy, since the cumulants are much more sensitive to structure in the tails of the distribution than in its centre (a fact that is accentuated if the cumulants, being very sensitive to outliers, are estimated from finite samples). Also, the skew and kurtosis are not meaningful for multimodal distributions. For 1D projections, Hyvärinen (1998) gives a further approximation of the differential entropy based on a first-order approximation of the density of maximum entropy given some simple constraints. In section 8.7.6 we give upper and lower bounds for the entropy of a Gaussian mixture of known parameters.

Jones (1983) found little difference in practice between $\int p \ln p$ and $\int p^2$. However, for fixed variance the integral $\int p \ln p$ is minimised by the normal distribution, while $\int p^2$ is minimised by the Epanechnikov kernel (Silverman, 1986).

- The original univariate index of Friedman and Tukey (1974), of difficult optimisation and dependent on a parameter h, is large whenever many points are clustered in a neighbourhood of size h. Huber (1985) noted that this index is essentially based on $\int p^2 = \|p\|_2^2 = \text{E}_p\{p(x)\}$.

- The univariate exploratory projection pursuit index of Friedman (1987):

$$I(x) \stackrel{\text{def}}{=} \int_{-1}^{1}\left(p(u) - \frac{1}{2}\right)^2 du = \int_{-1}^{1} p^2(u)\, du - \frac{1}{2}$$

where $u \stackrel{\text{def}}{=} 2\Phi(x) - 1 \in [-1, 1]$ and Φ is the standard normal cdf (the data are assumed to have been sphered). u is uniform in $[-1, 1]$ if x is standard normal in $(-\infty, \infty)$ and hence nonuniformity of u

will mean nonnormality of x. The index is minimised by the normal distribution and in practice it is implemented approximately by a Legendre polynomial expansion.

This index can be reexpressed by transforming back from the u variable to x as

$$I(x) = \int_{-\infty}^{\infty} \frac{(p(x) - \phi(x))^2}{2\phi(x)}\, dx$$

which suggests some variations, such as $\int (p-\phi)^2$ (Hall, 1989) or $\int (p-\phi)^2\phi$ (Cook et al., 1993).

- Intrator and Cooper (1992) present an objective function formulation of the BCM neuron (a synaptic modification rule proposed to explain visual cortical plasticity by Bienenstock et al., 1982) which is equivalent to a projection index that seeks multimodality.

Many other indices exist, often designed for one- or two-dimensional projections, that measure various kinds of departure from normality (e.g. Eslava and Marriott, 1994; Posse, 1990, 1995). Extension to high dimensions is often difficult.

Indices based on polynomial moments operate by approximating the density by a truncated series of orthogonal polynomials (Legendre, Hermite, etc.), such as the cumulant-based indices mentioned. They do not have to be recomputed at each step of the numerical procedure, as they can be derived for each projection direction from sufficient statistics of the original data set. However, they perform poorly unless high-order polynomials are used, since deviation from normality cannot be captured in second-order correlations, as mentioned in the discussion of independent component analysis (section 2.6.3).

4.6.2 Projection pursuit regression and density approximation

Low-dimensional projections of a distribution can be used as building blocks for function approximation. If the number of projections used L is smaller than the number of predictor variables D, then this combines dimensionality reduction (of the predictor variables) and regression. This idea is the basis of the following procedures:

Projection pursuit regression (Friedman and Stuetzle, 1981) is a nonparametric regression approach that works by additive composition, constructing an approximation to the desired response function by means of a sum of low-dimensional smooth functions, called *ridge functions*, that depend on low-dimensional projections through the data (we consider a one-dimensional function for simplicity):

$$f(\mathbf{t}) = \sum_{l=1}^{L} g_l(\mathbf{a}_l^T \mathbf{t}).$$

Algorithms exist to estimate the projection directions $\{\mathbf{a}_l\}_{l=1}^L$ and ridge functions $\{g_l\}_{l=1}^L$ nonparametrically (Friedman and Stuetzle, 1981) or parametrically, using a neural network (Hwang et al., 1994; Zhao and Atkeson, 1996).

Generalised additive models (Hastie and Tibshirani, 1990) are a particular case of projection pursuit regression where the projection directions are fixed to the coordinate axes:

$$f(\mathbf{t}) = \alpha + \sum_{d=1}^{D} g_d(t_d).$$

It is more easily interpretable and the individual components $g_d(t_d)$ can be plotted, but it is more restricted. Hastie and Tibshirani (1990) give a backfitting[12] algorithm to estimate $\{g_d\}_{d=1}^D$ nonparametrically.

Multivariate adaptive regression splines (MARS) (Friedman, 1991) are an extension to generalised additive models to allow interactions between variables: each basis function g_d is a product of one-dimensional spline functions, each one depending on one data variable.

Several methods developed in the chemometrics literature, such as *principal component regression* and *partial least squares* (reviewed by Frank and Friedman, 1993) are also closely related to projection pursuit

[12]The backfitting algorithm is an iterative method to fit additive models that fits each term to the residuals given the rest. It is a version of the Gauss-Seidel method of numerical linear algebra.

regression[13]. Such methods include mechanisms to determine the number of projections that gives the smallest regression error on a test set. They usually perform much better than ordinary least squares regression when the sample size N is smaller than the dimensionality D and the predictor variables have a high degree of collinearity (as is the case in many chemical applications).

Projection pursuit density approximation (Friedman et al., 1984) uses a multiplicative composition so that the estimate can be nonnegative and integrate to 1:

$$f(\mathbf{t}) = \prod_{l=1}^{L} h_l(\mathbf{a}_l^T \mathbf{t}).$$

Friedman et al. (1984) and Huber (1985) give algorithms to fit the model. This approach is related to the recently proposed product-of-experts architecture (Hinton, 1999).

4.7 Local dimensionality reduction

In local dimensionality reduction methods, a global model for the data manifold is built as a combination of several simple local models (usually linear). This is justified by several reasons:

- Taylor's theorem: any differentiable function becomes approximately linear in a sufficiently small region around a point.
- The data manifold may actually consist of separate manifolds which may or may not be connected together in one piece; i.e., it may be clustered.
- The intrinsic dimensionality of the data may vary along the manifold. Consider, for example, a manifold with the aspect of fig. 4.6; or the Lorenz attractor (Peitgen et al., 1992), which globally spans three dimensions but locally can be described with only two in most of the space (fig. 4.12).
- The intrinsic dimensionality may not vary, but the orientation may vary as one moves along the manifold. For example, the Lorenz attractor could be embedded in two non-parallel planes.

The key point is that, while the global data manifold may be highly nonlinear and require the whole data space to embed it, individual parts of it may require simple, linear models. In fact, using a complex global model able to represent a large number of manifolds (via a large number of parameters), such as nonlinear autoassociators (section 4.5.2), has several disadvantages:

- The power of the model is wasted in those areas of the space where the manifold is approximately linear.
- A large data set is required to fit a large number of parameters.
- Training becomes difficult because, due to the high flexibility of the model, the error function is likely to have a lot of local minima.

In contrast, in local dimensionality reduction we split the global manifold into simple parts that can be learned easily: fast, with little data and no (or few) local minima. This, then, does not result anymore in a single, global dimensionality reduction mapping (from the data space to a single low-dimensional space) because each local mapping has its own low-dimensional space with its own dimension; and likewise there is no single, global reconstruction mapping. This also happened with mixtures of latent variable models (section 2.9.3). Local dimensionality reduction thus requires:

- Simple dimensionality reduction models as building blocks (typically PCA), usually distributed around the space and each one having a limited reach (hence the locality).
- A way to determine the dimensionality of each component.
- A responsibility assignation rule that, given a point in data space, assigns a weight or responsibility for it to each component. This can be seen as clustering.
- A way to learn both the local models (manifold fitting) and the responsibility assignment (clustering).

[13] Surprisingly, Frank and Friedman (1993) make no mention of projection pursuit regression in their review of chemometrics regression tools.

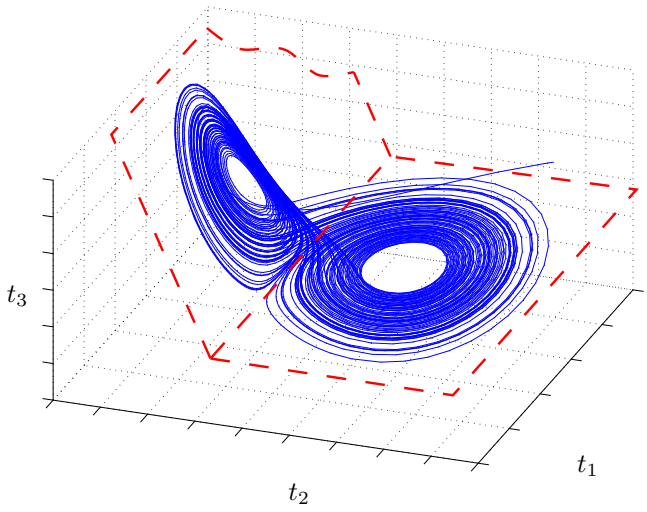


Figure 4.12: The Lorenz attractor can locally be described with two dimensions in most of the three-dimensional space.

The responsibility assignation can be *hard* or *soft*:

- Hard: a single component receives all the responsibility and the rest receive no responsibility at all. It is a winner-take-all approach, usually a form of vector quantisation.
- Soft: the responsibility is distributed among all components as a partition of unity, so that when training, a given data point will result in an update of all components; and when reducing dimensionality, the reduced-dimension representative will be the average of the local reduced-dimension representatives weighted by the respective responsibilities.

It has been observed empirically that soft assignation of responsibilities outperforms hard assignation, which in a probabilistic framework is a consequence of the optimality (in the L_2 sense) of the mean of a distribution to represent the whole distribution—the mode (a form of hard assignation) not being optimal. But the average of two points in different local models need not (and probably will not) belong to the global manifold. We found this situation in section 2.9.1 when having to pick a point of the posterior distribution in latent space as reduced-dimension representative, and we will find it in more detail in chapter 7 when reconstructing arbitrarily missing values. The moral is that data points for which more than one local model are significatively responsible are problematic.

Also, a soft assignment will yield a continuous dimensionality reduction mapping (if the local models have continuous dimensionality reduction mappings as well), which violates the self-consistency condition (4.2) of principal curves, as discussed in section 4.8.

From a probabilistic point of view, the concept of local models and responsibility assignation is naturally expressed as a mixture (of latent variable models) and was covered in section 2.9.3. The training criterion is then log-likelihood rather than reconstruction error, since the probability model attempts to model the noise as well as (and separately from) the underlying manifold. Formulating the local dimensionality reduction problem as a mixture of distributions results in a unified view of the whole model and its probabilistic nature brings a number of well-known advantages, in particular the fact that typically we can derive an EM algorithm that will train all parameters (those of the local models and those of the responsibility assignment) at the same time, with guaranteed convergence and often in a simple way: the E step assigns the responsibilities while the M step fits each local model. Other advantages and also disadvantages are mentioned in the conclusions of the thesis (chapter 11).

In the rest of this section we briefly mention some local dimensionality reduction approaches that do not define a density model in the data space. As one would expect, all of them use PCA-based components, given the attractive properties of PCA mentioned in section 4.5. Therefore, the global manifold is piecewise linear; we can visualise it as a collection of patches glued together. In the region of data space where a given individual model is much more responsible than all the others the dimensionality reduction mapping is then the orthogonal projection on the local principal component subspace. Therefore, the self-consistency

initialise
    $\{\boldsymbol{\mu}_m\}_{m=1}^M \leftarrow$ random subset of the data points — Centroids
    $\boldsymbol{\Sigma}_m \leftarrow \mathbf{I}$ for each $m = 1, \ldots, M$ — Covariances
repeat
    Assign each data point to the cluster that best reconstructs it — Hard assignation
    for $m = 1, \ldots, M$ — For each cluster
        $\boldsymbol{\mu}_m \leftarrow$ mean of data points assigned to cluster m — Update centroids
        $\boldsymbol{\Sigma}_m \leftarrow$ covariance matrix of data points assigned to cluster m — Update covariances
        $\mathbf{U}_m \leftarrow$ first principal components of $\boldsymbol{\Sigma}_m$ — Update local PCA directions
    end
until the fractional change in average reconstruction error is small
return $\{\boldsymbol{\mu}_m, \mathbf{U}_m\}_{m=1}^M$

Figure 4.13: Pseudocode of the VQPCA algorithm of Kambhatla and Leen (1997).

condition (4.2) of principal curves is locally satisfied. Globally this need not be true—and it certainly is not in the case of soft assignment.

Bregler and Omohundro (1994, 1995) perform the clustering with a Gaussian mixture starting from the k-means solution (Duda and Hart, 1973) and associate to every centroid thus obtained a local PCA. Although the algorithmic details are not clear in these papers, it seems that a fixed number of nearest neighbours of each centroid is used to compute a local covariance matrix, from which one infers how many principal components to keep as well as the local principal component subspace. Therefore, some data points may participate in the covariance matrices of different, neighbouring centroids. The dimensionality reduction is soft, where the weight of each local model is its posterior probability (with respect to the Gaussian mixture density) given the data point. They use this model for learning a space of lip shapes in a lip-tracking application.

The *optimally adaptive transform coding* of Dony and Haykin (1995) consists of a collection of principal component subspaces with the peculiarity that they are centred at the same point. Therefore, no centroids need to be estimated, although the approach is clearly more limited. Responsibility is computed in a hard fashion both for dimensionality reduction and for training (i.e., for a given data point only one model has its orientation updated). The winner model is the most parallel to the data vector, i.e., the one over which the projection of the data vector is longest. The update of the model is done via an online PCA algorithm, such as those mentioned in section 4.5.1. They use the model for compression and feature extraction of images.

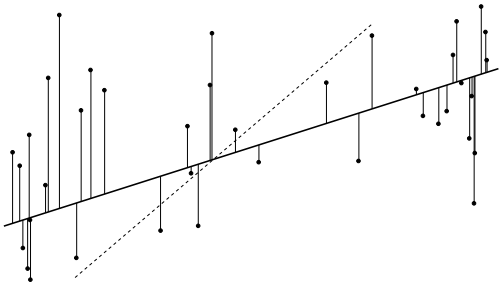
Hinton et al. (1997) applied a mixture of PCAs to model the manifold of handwritten characters. Each PCA module was implemented with a linear autoassociator and both hard and soft versions of training were used, based on k-means and EM, respectively. For the soft version, the training criterion was a log-likelihood function derived from an image model (not from a probabilistic view of PCA). They found a similar performance between the mixture of PCAs and a mixture of factor analysers (which generates a density model for the data).

The *vector quantisation PCA algorithm (VQPCA)* of Kambhatla and Leen (1997) uses a vector quantisation algorithm to produce a hard partition of the data space (a Voronoi tessellation, defined in section 4.10.2). In each of the Voronoi cells a separate PCA is fitted, whose and mean covariance matrix are calculated from the data points in that cell. Rather than assigning data points to centroids using the Euclidean distance, they use a distortion function that takes into account not only the distance to the centroid but also the projection on its local subspace. The iterative algorithm is shown in fig. 4.13. They apply their model to dimensionality reduction of speech and image data and show it to outperform nonlinear autoassociators in both speed and reconstruction error.

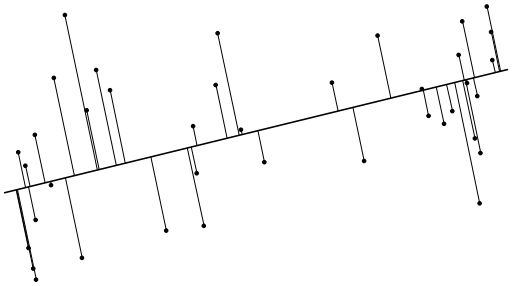
It is interesting that the partitions formed when clustering according to reconstruction error alone can be nonlocal, as simulations by Kambhatla and Leen (1997) and Tipping and Bishop (1999a) showed.

The literature contains many similar local dimensionality reduction models, so the ones we have presented should be considered only as representative. References to such other models can be found in several of the papers mentioned (Hinton et al., 1997; Kambhatla and Leen, 1997; Tipping and Bishop, 1999a). The same kind of ideas has been used not only for dimensionality reduction but also for regression and classification, as in the hierarchical mixtures of experts of Jordan and Jacobs (1994).

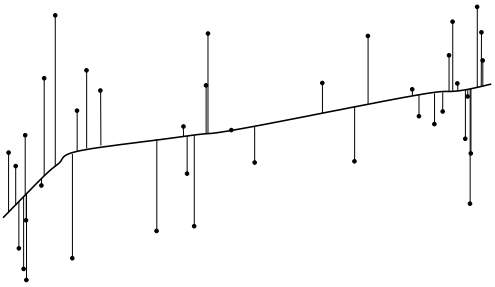
a. Linear, nonsymmetric: regression line.



b. Linear, symmetric: principal-component line.



c. Nonlinear, nonsymmetric: regression curve.



d. Nonlinear, symmetric: principal curve.

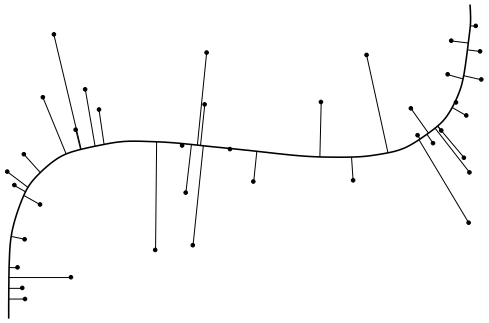


Figure 4.14: Principal curves as generalised (nonlinear, symmetric) regression. (a) The linear regression line minimises the sum of squared deviations in the response variable (or in the independent one, for the dashed line). (b) The principal-component line minimises the sum of squared deviations in all the variables. (c) The smooth regression curve minimises the sum of squared deviations in the response variable, subject to smoothness constraints. (d) The principal curve minimises the sum of squared deviations in all the variables, subject to smoothness constraints. From Hastie and Stuetzle (1989).

4.8 Principal curves

Principal curves are intuitively defined as smooth curves that pass through the middle of a D-dimensional data set, providing a nonlinear summary of it (Hastie and Stuetzle, 1989). They are estimated in a nonparametric way, i.e., their shape is suggested by the data. Their motivation is a generalisation of regression (see fig 4.14):

- Linear regression minimises the sum of squared deviations in the response variable $y = ax + b$ (i.e., in the vertical direction in an X-Y graph). Thus changing the roles produces a different regression line (dotted line).
- The first principal component is a regression line symmetrical with respect to all the variables, minimising orthogonal deviation to that line.
- Nonlinear regression produces a curve that minimises the sum of squared deviations in the response variable (the vertical deviations) subject to some form of smoothness constraint. Again, changing the roles produces a different regression curve.
- Principal curves are a natural generalisation for nonlinear, symmetric regression: they attempt to minimise the sum of squared deviations in all the variables (i.e., the orthogonal or shortest distance from the curve to the points) subject to smoothness constraints.

Therefore, if we consider that $\mathbf{f} : x \in \mathcal{X} \subset \mathbb{R} \longrightarrow \mathbf{t} = \mathbf{f}(x) \in \mathbb{R}^D$ is a smooth curve in $\mathbb{R}^D$ parameterised by its arc length x, principal curves define, in the sense of section 4.2:

- A dimensionality reduction mapping F: given $\mathbf{t} \in \mathbb{R}^D$, $F(\mathbf{t})$ is the arc length of the nearest point[14] in the curve to $\mathbf{t}$ in the Euclidean distance, i.e., the arc length of its orthogonal projection on the curve.
- A reconstruction mapping $\mathbf{f}$ given by the principal curve parametric equation, $\mathbf{f}(x)$.

The advantage of using the arc length parametrisation is that the distance *along the curve* between points $\mathbf{f}(x_1)$ and $\mathbf{f}(x_2)$ is simply $|x_1 - x_2|$ (i.e., the geodetic distance, which in general is larger than the Euclidean, or straight-line, distance between $\mathbf{f}(x_1)$ and $\mathbf{f}(x_2)$).

We say that a curve is **self-consistent** with respect to a distribution if the average of all data points that (orthogonally) project onto a given point on the curve coincides with the point on the curve:

$$\forall x \in \mathcal{X} : \ \mathrm{E}\left\{\mathbf{t} | F(\mathbf{t}) = x\right\} = \mathbf{f}(x). \tag{4.2}$$

We can then say that principal curves pass through the middle of the data in a smooth way and are self-consistent for that distribution. Using this property, Hastie and Stuetzle (1989) prove that, for a given distribution, principal curves are the stationary points of the average of the Euclidean distance of a data point to its projection on the curve for perturbations of bounded norm and bounded derivative. This result, verified by principal components if only straight lines are considered, confirms the aforementioned role of principal curves as nonlinear, symmetric regression.

However, this definition of principal curves poses several questions: for what kinds of distributions do principal curves exist, how many different principal curves exist for a given distribution and what are their properties? These questions have only been answered for some particular cases:

- For ellipsoidal distributions (e.g. the normal distribution) the principal components are principal curves. In higher dimensions, the subspaces spanned by any subset of principal components are principal manifolds.
- For spherically symmetric distributions any straight line through the mean is a principal curve.
- For 2D spherically symmetric distributions a circle with centre at the mean and radius $\mathrm{E}\left\{\|\mathbf{t}\|\right\}$ is a principal curve too and has smaller expected squared distance from the distribution than the straight lines.
- If a straight line is self-consistent, then it is a principal component. In other words, linear principal curves are principal components.

For a model of the form $\mathbf{t} \stackrel{\text{def}}{=} \mathbf{f}(x) + \mathbf{e}$, with $\mathbf{f}$ smooth and $\mathrm{E}\left\{\mathbf{e}\right\} = 0$ (such as all the latent variable models described in chapter 2), $\mathbf{f}$ is not a principal curve in general, as fig. 4.15 shows. This means that the principal curve is biased for the functional model, although the bias seems to be small and to decrease to 0 as the variance of the errors gets small relative to the radius of curvature of $\mathbf{f}$. This bias is a consequence of the self-consistency condition (4.2); in fact, relaxing it to the condition (2.5) and considering $\mathbf{t}$ and x as random variables results in a continuous latent variable model, which is unbiased by definition (eq. (2.5) in section 2.3.1). Banfield and Raftery (1992) give a robust estimation method for closed principal curves that reduces both bias and variance.

The definition of principal curves can be naturally extended to several dimensions, in which case we could call them *principal manifolds* (although the arc length, or unit-speed, parameterisation is not naturally defined for more than one dimension). However, once again the curse of the dimensionality makes smoothing in several dimensions hard unless data are abundant. Note also that principal curves depend critically on the scaling of the features, as all projection techniques do.

The definition of the projection on the principal manifold $\mathcal{M}$ as orthogonal projection leads necessarily to discontinuities in the dimensionality reduction mapping if the projection sets of two points in $\mathcal{M}$ intersect:

$$\exists \mathbf{t} \in \mathbf{F}^{-1}(\mathbf{x}_1) \cap \mathbf{F}^{-1}(\mathbf{x}_2) \Rightarrow \text{ is } \mathbf{F}(\mathbf{t}) = \mathbf{x}_1 \text{ or } \mathbf{F}(\mathbf{t}) = \mathbf{x}_2?$$

An example is the centre of the circle in figure 4.15, since for any two points in the circle, $\mathbf{F}^{-1}(x_1) \cap \mathbf{F}^{-1}(x_2) = \{C\}$. Choosing one of the possibilities (as Hastie and Stuetzle (1989) do: the one with smallest arc length) leads to a discontinuity at such point $\mathbf{t}$. This will happen at some data points for any manifold except when no two projection sets intersect, i.e., they are all parallel, which implies that the principal manifold is a hyperplane—spanned by principal components.

Hastie and Stuetzle (1989) give a construction algorithm for principal curves, shown in fig. 4.16. Although by definition principal curves are fixed points of this algorithm, it has not been proven to converge in general. Observe that:

[14] For definiteness, in the exceptional case where there are several nearest points, we take the one with largest arc length.

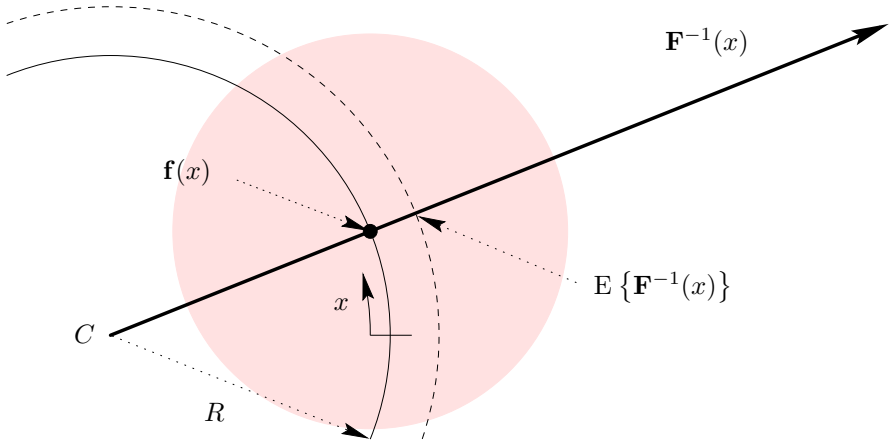


Figure 4.15: Bias in principal curves. In this example, the data distribution is $\mathbf{t} \stackrel{\text{def}}{=} \mathbf{f}(x) + \mathbf{e}$ where $\mathbf{f}$ is the circle of radius R (solid line), parameterised by arc length x, and $\mathbf{e} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$. The radius of the shaded circle is equal to σ. For any point $\mathbf{f}(x)$ on the curve, $\mathbf{F}^{-1}(x)$ (the set of points that are closest to $\mathbf{f}(x)$, i.e., whose reduced-dimension representative is x) is, by symmetry, the half-line starting in the circle centre, passing through $\mathbf{f}(x)$ and going towards infinity (marked thick). However, $\mathrm{E}\left\{\mathbf{F}^{-1}(x)\right\} \neq \mathbf{f}(x)$ and therefore the generating curve $\mathbf{f}$ is not self-consistent in the sense of condition (4.2). The radius of the self-consistent circle (dotted line) is larger than R and can be found to be equal to $R + \left(1 + \operatorname{erf}\left(\frac{R}{\sigma\sqrt{2}}\right)\right)^{-1} \sqrt{\frac{\pi}{2}}\sigma e^{-\frac{1}{2}\left(\frac{R}{\sigma}\right)^2}$.

- The algorithm converges to the first principal component if the conditional expectation operation is replaced by fitting a least-squares straight line. Principal curves are then local minima of the distance function (sum of squared distances).
- For probability distributions, both operations—projection and average or conditional expectation—reduce the expected distance from the points to the curve; for discrete data sets this is unknown.
- The algorithm does not necessarily produce smooth curves, in particular at the curve endpoints.
- The smoothness of the resulting principal curve is sensitive to the size of the neighbourhood over which averaging takes place.

By relaxing the self-consistency condition (4.2), Tibshirani (1992) redefines principal curves based on a continuous mixture model and estimates it via an EM algorithm. His view is equivalent to nonparametric estimation of a continuous latent variable model, as discussed in sections 2.5.2 and 2.3.1.1.

initialise
  $\tau \leftarrow 0$ — Iteration counter
  $\mathbf{f}^{(0)} \leftarrow$ first principal components — Principal curve
repeat
  Project the distribution onto $\mathbf{f}^{(\tau)}$, i.e., compute ${F^{(\tau)}}^{-1}(x)$ for each $x \in \mathcal{X}$. — Projection step
  $\mathbf{f}^{(\tau+1)}(x) \leftarrow \mathrm{E}\left\{\mathbf{t} | F^{(\tau)}(\mathbf{t}) = x\right\} = \mathrm{E}\left\{{F^{(\tau)}}^{-1}(x)\right\}$. — Averaging step
  Reparameterise $\mathbf{f}^{(\tau+1)}$ in terms of arc length.
until $\mathbf{f}^{(\tau+1)} \equiv \mathbf{f}^{(\tau)}$ — Self-consistence condition
return $\mathbf{f}^{(\tau)}$

Figure 4.16: Pseudocode of the construction algorithm for principal curves from Hastie and Stuetzle (1989). Since principal curves are estimated nonparametrically, the averaging step requires a local average of the data set.

Hastie and Stuetzle (1989) give an alternative definition of principal curves, related to spline smoothing, as curves that minimise the distance (conveniently defined) of the data points to the curve subject to a smoothness constraint. Thus, it becomes a regression problem of the data $\{\mathbf{t}_n\}_{n=1}^N$ as a function of the unknown variables $x_1, \dots, x_L$ (in L dimensions). This is then very similar to latent variable models but without a joint probability model for $\{\mathbf{t}, \mathbf{x}\}$. This second definition of principal curves has proven more amenable to developments. LeBlanc and Tibshirani (1994) implement principal curves parametrically in this sense using a MARS model (mentioned in section 4.6.2) extensible to higher dimensions. Kégl et al. (2000) prove that, by restricting the definition to curves of a given length at most, such principal curves always exist if the distribution has finite second moments and give a learning algorithm for them based on polygonal lines (restricted to the one-dimensional case).

The natural interpretation of principal curves as nonlinear, symmetric regression seems very attractive in terms of least-squares dimensionality reduction. However, before they become a practical framework for dimensionality reduction, a number of theoretical and practical questions must be answered and estimation algorithms that work in arbitrary dimensions must be developed.

4.9 Methods based on vector quantisation

Vector quantisation (Gray, 1984; Gray and Neuhoff, 1998) consists of summarising a data distribution in data space $\mathbb{R}^D$ via a discrete collection of **reference** or **codebook vectors** $\{\boldsymbol{\mu}_m\}_{m=1}^M \subset \mathbb{R}^D$. Once the codebook has been constructed (the reference vectors trained), for which various algorithms are available, a given data point $\mathbf{t} \in \mathbb{R}^D$ becomes *quantised* to the closest reference vector to it according to some convenient distance, usually the Euclidean. As such, vector quantisation is useful for classification but not for dimensionality reduction. A limited form of dimensionality reduction becomes possible if one imposes a topographic structure on the reference vectors. This can be done via a learning rule or objective function that, at the same time that it tries to fit the reference vectors to the data, discourages configurations that violate the topographic arrangement. Several such methods exist, of which we mention two: Kohonen's self-organising maps and the elastic net.

None of these methods properly defines a manifold that embeds the reference vectors[15]: the L-dimensional manifold in data space is defined indirectly by the location of the reference vectors. Therefore, no continuous dimensionality reduction and reconstruction mappings in the sense of section 4.2 are derived: given a data point, all one can do is to assign it to the closest reference vector and use its associated knot in the topographic arrangement as reduced-dimension representative, or interpolate in some way. This latter option consists of defining dimensionality reduction and reconstruction mappings by assuming the low-dimensional space to be embedded in an Euclidean space (e.g. by placing the lattice nodes of a self-organising map in the $\mathcal{X} = [0,1]^D$ hypercube in an equispaced way) and then fitting a regularised universal mapping approximator to the functions $\mathcal{X} \xrightarrow{\mathbf{f}} \mathcal{T}$ (reconstruction) and/or $\mathcal{T} \xrightarrow{\mathbf{F}} \mathcal{X}$ (dimensionality reduction), learned in a supervised way from the reference vectors and lattice points.

4.9.1 Kohonen's self-organising maps

Let $\{\mathbf{t}_n\}_{n=1}^N$ be a sample in the data space $\mathcal{T} = \mathbb{R}^D$. Kohonen's self-organising maps (SOMs) (Kohonen, 1995) learn, in an unsupervised way, a mapping between a 2D lattice[16] $\mathcal{X}$ and the data space that preserves the two-dimensional topology of the lattice and adapts to the manifold spanned by the sample. One can visualise the learning process as a flat sheet that twists around itself in D dimensions to resemble as much as possible the distribution of the data vectors.

Each of the reference vectors $\{\boldsymbol{\mu}_m\}_{m=1}^M$ in data space is associated with a node v_m in the 2D lattice $\mathcal{X}$. Assume we have defined two distances (typically Euclidean) $d_{\mathcal{T}}$ and $d_{\mathcal{X}}$ in the data space and in the lattice, respectively. The topology of the lattice is determined by the **neighbourhood function** $h(v_m, v_{m'})$. This is a symmetric function with values in $[0,1]$ that behaves as an inverse distance in the lattice: $h(v_m, v_m) = 1$ for any node v_m and, given a node v_m, for any other node $v_{m'}$ $h(v_m, v_{m'})$ is smaller the farther apart node $v_{m'}$ is from node v_m in the lattice. The neighbourhood of node v_m is composed of those nodes for which $h(v_m, v_{m'})$ is not negligibly small. In practice, usually $h(v_m, v_{m'}) = \exp(-d_{\mathcal{X}}(v_m, v_{m'})/2\sigma^2)$, where σ quantifies the range of the neighbourhood.

[15] One can always fit an interpolating manifold of some kind to the reference vectors once these have been trained, but this is then the original problem applied to the reference vectors rather than to the original sample.

[16] In the typical case, but the idea is valid for L-dimensional topological arrangements.

The reference vectors are initially distributed at random (or perhaps are a random set of the data vectors, or are scattered along the first principal components). A competitive learning rule, **Kohonen learning**, is applied iteratively over all data vectors until convergence is achieved. Given a data vector $\mathbf{t}_n$, let $\boldsymbol{\mu}_{m^*}$ be the reference vector closest to $\mathbf{t}_n$ in data space:

$$m^* = \arg \min_{m=1,\ldots,M} d_{\mathcal{T}}(\boldsymbol{\mu}_m, \mathbf{t}_n).$$

Learning occurs as follows (where τ is the iteration index and $\alpha^{(\tau)} \in [0,1]$ is the learning rate):

$$\boldsymbol{\mu}_m^{(\tau+1)} = \boldsymbol{\mu}_m^{(\tau)} + \alpha^{(\tau)} h^{(\tau)}(v_{m^*}, v_m)(\mathbf{t}_n - \boldsymbol{\mu}_m^{(\tau)}) = (1-\rho)\boldsymbol{\mu}_m^{(\tau)} + \rho \mathbf{t}_n$$

i.e., reference vector $\boldsymbol{\mu}_m$ is drawn a distance $\rho = \alpha^{(\tau)} h^{(\tau)}(v_{m^*}, v_m)$ toward data vector $\mathbf{t}_n$. The update affects only vectors whose associated nodes lie in the neighbourhood of the winner v_{m^*} and its intensity decreases with the iteration index τ because both $\alpha^{(\tau)}$ and the range of $h^{(\tau)}$ must decrease with τ for convergence reasons.

Intuitively it seems that the reference vectors $\boldsymbol{\mu}_m$ will become abundant in regions of $\mathbb{R}^D$ where the $\mathbf{t}_n$ are common and sparse where the $\mathbf{t}_n$ are uncommon, thus following the distribution of the data vectors. However, they do not approximate the data density even with infinite data; in fact, they underestimate the density where it is high and overestimate it where it is low (Kohonen, 1995, p. 110–111).

A batch training algorithm for SOMs exists that can be written as (see Mulier and Cherkassky, 1995 and references therein):

$$\boldsymbol{\mu}_m^{(\tau+1)} = \frac{\sum_{m'=1}^M h^{(\tau)}(v_{m'} - v_m) N_{m'}^{(\tau)} \boldsymbol{\mu}_{m'}^{(\tau)}}{\sum_{m'=1}^M h^{(\tau)}(v_{m'} - v_m) N_{m'}^{(\tau)}}$$

where $N_{m'}^{(\tau)}$ is the number of data points $\{\mathbf{t}_n\}_{n=1}^N$ that lie in the Voronoi cell of reference vector $\boldsymbol{\mu}_{m'}^{(\tau)}$ (i.e., the number of data points whose closest reference vector is $\boldsymbol{\mu}_{m'}^{(\tau)}$) and each iteration contains the updates due to all the data points. This equation has the form of a kernel regression (Nadaraya-Watson estimator; Silverman, 1986) where the response variables are the data variables $\mathbf{t}$, the predictor variables are the nodes v (which can be assumed to lie on an Euclidean space), the (unnormalised) kernels are $h^{(\tau)}(v_{m'} - v) N_{m'}^{(\tau)}$ and the "training set" is $\{(v_m, \boldsymbol{\mu}_m^{(\tau)})\}_{m=1}^M$, with both kernels and training set depending on the iteration index τ. From this point of view, at each iteration the SOM defines a continuous mapping $\mathbf{t} = \mathbf{f}^{(\tau)}(v)$ from latent space onto data space:

$$\mathbf{f}^{(\tau)}(v) = \frac{\sum_{m'=1}^M h^{(\tau)}(v_{m'} - v) N_{m'}^{(\tau)} \boldsymbol{\mu}_{m'}^{(\tau)}}{\sum_{m'=1}^M h^{(\tau)}(v_{m'} - v) N_{m'}^{(\tau)}}$$

as Mulier and Cherkassky (1995) argue. However, for $\tau \to \infty$, $h^{(\tau)}(v_{m'} - v) \to \delta(v_{m'} - v)$ and so the mapping $\mathbf{f}^{(\infty)}$ is only defined at the nodes $\{v_m\}_{m=1}^M$. Thus, a trained SOM does not define a continuous mapping from latent to data space.

In summary, Kohonen learning creates an L-dimensional arrangement such that:

- The number density of reference vectors in data space is approximately proportional to a power (smaller than one) of the data probability density.
- The mapping from the L-dimensional arrangement into data space (ideally) preserves the topographic ordering: nearby points in the data space are mapped onto nearby points in the lattice.

Kohonen's SOMs have proven successful in many practical applications, particularly in visualisation. However, their heuristic nature results in several shortcomings:

- Many parameters must be tuned by trial and error with little guarantee of success: the shape of the lattice (rectangular, etc.), the number of codebook vectors or the schedules for the evolution of the neighbourhood function and learning rate. In general, no schedules that guarantee convergence and no proofs of convergence exist. Thus, training is unreliable. However, the shape of the neighbourhood function is largely irrelevant (as happens in kernel estimation).
- No cost function to optimise can be defined, although SOMs have been shown to be approximately related to probabilistic cost functions (Luttrell, 1994; Utsugi, 1997a,b).
- Neither a probability distribution function nor a manifold function is obtained for the data.

Table 4.4 compares Kohonen's SOMs with GTM (section 2.6.5), which—being defined as a latent variable model—enjoys much more attractive theoretical properties. The table applies to most of the variations of SOMs that have been proposed (see Kohonen, 1995 for a review).

	SOM	GTM
Internal representation of manifold	Nodes $\{v_m\}_{m=1}^M$ in L-dimensional array, held together by neighbourhood function h	Point grid $\{\mathbf{x}_k\}_{k=1}^K$ in L-dimensional latent space that keeps its topology through smooth mapping $\mathbf{f}$
Definition of manifold in data space	Indirectly by locations of reference vectors	Continuously by mapping $\mathbf{f}$
Objective function	No	Yes: log-likelihood
Self-organisation	Difficult to quantify	Smooth mapping $\mathbf{f}$ preserves topology
Convergence	Not guaranteed	Yes, by the EM algorithm
Smoothness of manifold	Depends on $\alpha^{(\tau)}$ and $h^{(\tau)}$	Depends on basis functions parameters and prior distribution $p(\mathbf{x})$
Generative model	No; hence no density function	Yes
Additional parameters to select	$\alpha^{(\tau)}$, $h^{(\tau)}$ arbitrarily	None
Speed of training	Comparable according to Bishop et al. (1998b)	
Magnification factors	Approximated by the difference between reference vectors	Exactly computable anywhere

Table 4.4: Comparison between GTM and Kohonen's SOM.

4.9.2 The elastic net

The travelling salesman problem (Lawler et al., 1985) is a combinatorial optimisation task that requires to find the shortest circular tour through a set of N cities that passes through each city exactly once. It is an NP-complete problem: solving it is $\mathcal{O}(e^N)$. The elastic net algorithm (Durbin and Willshaw, 1987) generates good solutions in much less time by stretching a circular tour formed by M knots elastically linked in succession to fit the cities (the topographic arrangement of the elastic net can be extended to any dimension); that is, it is an algorithm for continuous, rather than discrete, optimisation. Let $\{\mathbf{t}_n\}_{n=1}^N$ represent the positions of the N cities in $\mathbb{R}^D$ and $\{\boldsymbol{\mu}_m\}_{m=1}^M$ the elastic net knots. The algorithm consists of the updating rule

$$\Delta\boldsymbol{\mu}_m \stackrel{\text{def}}{=} \alpha \sum_{n=1}^{N} w_{nm}(\mathbf{t}_n - \boldsymbol{\mu}_m) + \frac{1}{2}\beta K(\boldsymbol{\mu}_{m+1} - 2\boldsymbol{\mu}_m + \boldsymbol{\mu}_{m-1}) \qquad w_{nm} \stackrel{\text{def}}{=} \frac{e^{-\frac{1}{2}\left(\frac{\|\mathbf{t}_n - \boldsymbol{\mu}_m\|}{K}\right)^2}}{\sum_{m'=1}^{M} e^{-\frac{1}{2}\left(\frac{\|\mathbf{t}_n - \boldsymbol{\mu}_{m'}\|}{K}\right)^2}}$$

where α and β are constants and K is the scale parameter (which is typically annealed, i.e., slowly decreased to zero). The α term pulls the path toward the cities while the β term pulls neighbouring path points toward each other. The rule can be seen as an energy minimisation (or wiring length minimisation in the context of cortical maps; Durbin and Mitchison, 1990):

$$E(\{\boldsymbol{\mu}_m\}_{m=1}^M, K) \stackrel{\text{def}}{=} -\alpha K \sum_{n=1}^{N} \ln\left(\sum_{m=1}^{M} e^{-\frac{1}{2}\left(\frac{\|\mathbf{t}_n - \boldsymbol{\mu}_m\|}{K}\right)^2}\right) + \frac{\beta}{2}\sum_{m=1}^{M}\left\|\boldsymbol{\mu}_m - \boldsymbol{\mu}_{m+1}\right\|^2 \qquad \Delta\boldsymbol{\mu}_m = -K\frac{\partial E}{\partial \boldsymbol{\mu}_m} \quad (4.3)$$

or (by exponentiation, analogously to the use of the Boltzmann-Gibbs distribution in statistical mechanics) as a maximum a posteriori estimation of the following probability model (Durbin et al., 1989):

- Prior probability of a tour that favours short tours: $p(\{\boldsymbol{\mu}_m\}_{m=1}^M) \stackrel{\text{def}}{\propto} \prod_{m=1}^{M} e^{-\frac{\beta}{2\alpha K}\|\boldsymbol{\mu}_m - \boldsymbol{\mu}_{m+1}\|^2}$. That is, a correlated Gaussian. Topologies other than the nearest-neighbour one can be used (Dayan, 1993; Utsugi, 1997a).
- Probability of the city collection given a tour: $p(\{\mathbf{t}_n\}_{n=1}^N|\{\boldsymbol{\mu}_m\}_{m=1}^M) \stackrel{\text{def}}{=} \prod_{n=1}^{N} p(\mathbf{t}_n|\{\boldsymbol{\mu}_m\}_{m=1}^M)$ with $p(\mathbf{t}_n|\{\boldsymbol{\mu}_m\}_{m=1}^M) \stackrel{\text{def}}{=} \frac{1}{M}\sum_{m=1}^{M}(2\pi K^2)^{-D/2} e^{-\frac{1}{2}\left(\frac{\|\mathbf{t}_n - \boldsymbol{\mu}_m\|}{K}\right)^2}$. That is, a product of Gaussian mixtures with isoprobable components centred at the tour knots.

The algorithm minimises E at large K, where there is a single minimum, and then tracks the minimum down to small K, where every city location $\mathbf{t}_n$ is matched by some tour knot $\boldsymbol{\mu}_m$, as Durbin et al. (1989) prove.

4.10 Distance-preserving methods

In this section we discuss an approach to dimensionality reduction based on representing the data manifold by a low-dimensional Euclidean space where the distances between observed points are preserved. This is a purely geometric point of view, as opposed to the generative one of latent variable models or the minimum squared reconstruction error (orthogonal projection) of several of the methods of this chapter. That is, given a noisy sample $\{\mathbf{t}_n\}_{n=1}^N$ from a data manifold $\mathcal{M} \subset \mathcal{T} \subset \mathbb{R}^D$ of dimensionality $L \leq D$, the aim of distance-preserving methods is to find a collection of points $\{\mathbf{x}_n\}_{n=1}^N$ in a metric space $\mathcal{X} \subset \mathbb{R}^L$ associated with the high-dimensional sample such that the distance (to be defined) between two points $\mathbf{t}_n$ and $\mathbf{t}_m$ is approximately the same as the distance (usually Euclidean) between their associated low-dimensional points $\mathbf{x}_n$ and $\mathbf{x}_m$.

The low-dimensional Euclidean space $\mathcal{X}$ is not unique, since any rigid motion (translation, orthogonal rotation or reflection) of the $\{\mathbf{x}_n\}_{n=1}^N$ points preserves the distances between them; and the topology of $\mathcal{X}$ should match that of the data manifold $\mathcal{M}$. For example, if the latter is a closed surface, then using a rectangular low-dimensional space will lead to discontinuities.

Regarding the construction of a dimensionality reduction mapping $\mathcal{T} \xrightarrow{\mathbf{F}} \mathcal{X}$ and a reconstruction mapping $\mathcal{X} \xrightarrow{\mathbf{f}} \mathcal{T}$, there are two possibilities:

- Implicit definition: analogously to the case of methods based on vector quantisation, the data manifold is implicitly defined through the collection of points $\{\mathbf{t}_n\}_{n=1}^N$ in data space, and together with their associated points $\{\mathbf{x}_n\}_{n=1}^N$ in the low-dimensional map, a dimensionality reduction mapping and a reconstruction mapping are also implicitly defined. Such implicit definitions can be made explicit by fitting to them a regularised parametric model with supervised learning, e.g. an MLP with weight decay. Regularising the mapping approximator is very important since the whole process implicitly assumes that there is no noise—clearly a dangerous assumption since the interpoint distances (either through a straight line or through a geodesic) are heavily influenced by noise. Traditional multidimensional scaling, the Sammon mapping and methods based on geodetic distances generally belong to the implicit-definition type.

- Explicit definition: rather than directly learning the map points $\{\mathbf{x}_n\}_{n=1}^N$, define a parametric dimensionality reduction mapping $\mathbf{x} \stackrel{\text{def}}{=} \mathbf{F}(\mathbf{t}; \boldsymbol{\Theta})$ and learn instead the parameters $\boldsymbol{\Theta}$ that minimise the stress function (4.4) below. This allows to map new data vectors not in $\{\mathbf{t}_n\}_{n=1}^N$ and has the additional advantage of having fewer parameters compared to directly learning $\{\mathbf{x}_n\}_{n=1}^N$ (which requires LN parameters). This approach has been proposed by Webb (1995) (see also Webb, 1999), who used a radial basis function network for $\mathbf{F}$ with multidimensional scaling and iterative majorisation[17] as stress minimisation algorithm; and by Mao and Jain (1995), who used a multilayer perceptron for $\mathbf{F}$ with Sammon's mapping and online gradient descent (resulting in a variation of backpropagation) with an optional momentum term as stress minimisation algorithm.

4.10.1 Multidimensional scaling (MDS)

Multidimensional scaling (MDS) (Cox and Cox, 1994; Kruskal and Wish, 1978; Mardia et al., 1979; Webb, 1999) is the traditional statistical method for uncovering structure in a data set by plotting a low-dimensional map that preserves the proximities in the (high-dimensional) data set. That is, MDS plots similar objects close together. It has been applied to psychology, sociology, anthropology, economy, educational research, etc. MDS is actually a set of mathematical techniques differing in various theoretical and algorithmic aspects.

Suppose we have a set of N objects and that a measure of the similarity of these objects with each other is known. This measure, called *proximity*, is a number that indicates how similar two objects are or are perceived to be. It can be obtained in different ways, e.g. by asking people to judge the psychological closeness of the stimulus objects. What MDS does is to draw a spatial representation or map in which each object is represented by a point and the distances between points resemble as faithfully as possible the original

[17] Iterative majorisation is an minimisation algorithm where at each iteration one defines a majorisation function (i.e., an upper bound of the objective function) that has a single, easily computable minimum—typically a quadratic function. Under certain conditions, the sequence of minima converges to a minimum of the objective function. It does not require computing gradients of the latter but is very slow.

	A	B	C	···	0
A	82	04	08		03
B	06	84	37		04
C	04	38	87		12
···				···	
0	09	03	11		94

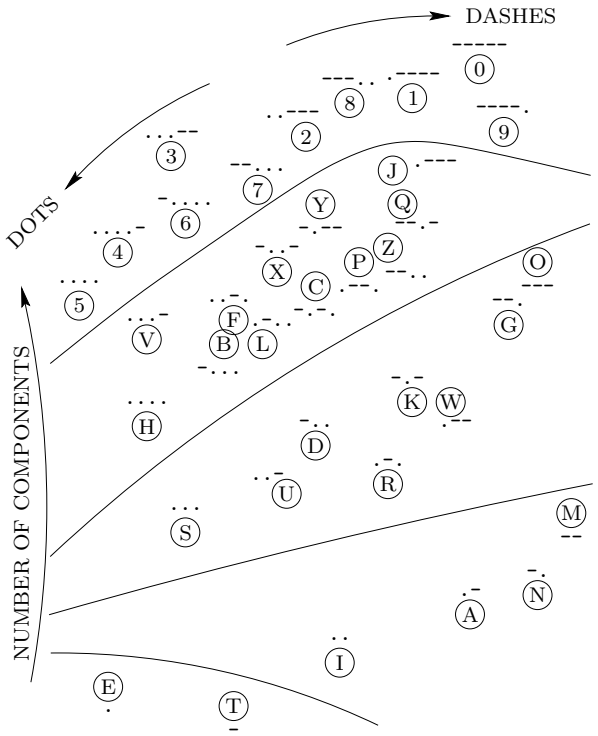


Figure 4.17: *Left*: data from Rothkopf (1957) on similarities among Morse code symbols. *Right*: 2D map obtained for the Morse code similarities by Shepard (1963) with multidimensional scaling.

similarity information; i.e., the larger the dissimilarity between two objects, the farther apart they should be in the spatial representation. This geometrical configuration of points reflects the hidden structure of the data and often makes it much easier to understand.

Let us consider a classical example. Confusions among 36 auditory Morse code signals were collected by Rothkopf (1957). Our "high-dimensional" objects are here the signals, each of which consists of a sequence of dots and dashes, such as `-.-` for `K` and `..---` for `2`. Subjects who did not know Morse code listened to a pair of signals (produced at a fixed rapid rate by machine and separated by a quiet period of 1.4 seconds), and were required to state whether the two signals they heard were the same or different. Each number in the table of fig. 4.17 is the percentage of roughly 150 observers who responded "same" to the row signal followed by the column signal. This matrix is roughly symmetric, with large diagonal entries and small off-diagonal entries, as expected, and contains the proximities data. Figure 4.17 (right) shows the result of applying MDS to those proximities (from Shepard, 1963) using a two-dimensional map. The 36 circles represent the points found and are labelled with the corresponding Morse code. In this case, MDS clearly shows that the data are governed by a sort of length parameter, the number of components of the signal, as well as by the individual numbers of dots and dashes.

All is needed for MDS is the proximities matrix, not the actual locations of the objects in a hypothetic high-dimensional space, which may be nonsensical, as in the Morse code example. The proximities need not be distances in the mathematical sense; in particular, they need not satisfy the symmetry property or the triangle inequality. However, for the dimensionality reduction problem these proximities will be determined from a set of points $\{\mathbf{t}_n\}_{n=1}^N$ in a high-dimensional space $\mathcal{T} \subset \mathbb{R}^D$. Typical definitions of proximity for this case are the Euclidean distance (L_2-norm), the Mahalanobis distance with respect to some (semi)positive definite matrix (e.g. the inverse data covariance matrix, in which case it is equivalent to applying MDS to the presphered data), the Manhattan distance (L_1-norm), etc.

Formally, assume the input data are the pairwise proximities[18] $\{\delta_{nm}\}_{n,m=1}^N$. If they come from a data set $\{\mathbf{t}_n\}_{n=1}^N$, then $\delta_{nm} \stackrel{\text{def}}{=} d_{\mathcal{T}}(\mathbf{t}_n, \mathbf{t}_m)$. For an L-dimensional map, the output data will be a set of points $\{\mathbf{x}_n\}_{n=1}^N \subset \mathbb{R}^L$, referred to some unimportant coordinate system, such that the distances $d_{nm} \stackrel{\text{def}}{=} d_{\mathcal{X}}(\mathbf{x}_n, \mathbf{x}_m)$ $\forall n, m = 1, \ldots, N$ (typically Euclidean) are as close as possible to a function f of the corresponding proximities,

[18]This is called two-way MDS. In three-way MDS we have several sets of proximities (e.g. in different times or by different subjects).

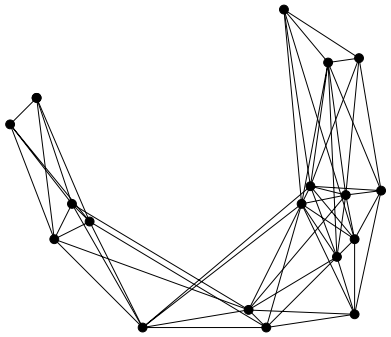


Figure 4.18: The horseshoe phenomenon. The graph shows a two-dimensional Euclidean map obtained by MDS where points whose associated objects have a similarity above a certain threshold are linked by a line.

$f(\delta_{nm})$. The computational procedure is as follows. Define an objective function, traditionally called *stress*, such as:

$$\text{stress}(\{\mathbf{x}_n\}_{n=1}^N, f) \stackrel{\text{def}}{=} \sqrt{\frac{\sum_{n,m=1}^N (f(\delta_{nm}) - d_{nm})^2}{\text{scale factor}}} \tag{4.4}$$

where the scale factor will typically be $\sum_{n,m=1}^N d_{nm}^2$. If the map is defined via a parametric function $\mathbf{F}$, $\mathbf{x} \stackrel{\text{def}}{=} \mathbf{F}(\mathbf{t};\boldsymbol{\Theta})$, then $d_{nm} \stackrel{\text{def}}{=} d_{\mathcal{X}}(\mathbf{F}(\mathbf{t}_n;\boldsymbol{\Theta}), \mathbf{F}(\mathbf{t}_m;\boldsymbol{\Theta}))$ and so the stress is a function of $\boldsymbol{\Theta}$ and f. Then, find the function f and the map $\{\mathbf{x}_n\}_{n=1}^N$ or $\boldsymbol{\Theta}$ that produce the minimal stress (for which there are several algorithms available).

If f is constrained to be monotonic, the ordering of the distances will be preserved (even if they are nonuniformly stretched or shrunk). Such MDS is called *metric*. The particular case where f is the identity (or linear) and the stress function is $\sum_{n,m=1}^N (\delta_{nm} - d_{nm})^2$ is called *classical scaling* and has an analytic solution which, if the distances come from a data set in an Euclidean space, is basically equivalent to PCA. If only the ordering of the proximities is used (but not their values), the MDS is called *nonmetric* or *ordinal* and is particularly suitable when the distances are qualitative and only determine a rank

Once f and $\{\mathbf{x}_n\}_{n=1}^N$ or $\boldsymbol{\Theta}$ have been determined, the solution map can be freely translated and rotated (perhaps to appear in a more aesthetical way) without changing the value of the stress. f can be plotted together with the pairs (δ_{nm}, d_{nm}) in a scatter (or Shepard) diagram, that plots the distance in L-dimensional space versus the proximities. If the proximities are dissimilarities (i.e., dissimilar objects have a large proximity value), it will be a rising pattern; otherwise, it will be a falling one. If each of the objects has an associated value y_n, regression can be performed on the generated map to further help to interpret the data: $\{(\mathbf{x}_n, y_n)\}_{n=1}^N$.

Degeneracy can happen if the objects have a natural clustering and the dissimilarities between objects in different clusters are (almost) all larger than the dissimilarities within each cluster. In this case, (almost) all points of a single cluster will converge to a single location, the stress will converge to 0 and the scatter diagram will be a staircase.

4.10.1.1 Selection of the map dimension

Obviously, the larger the dimension of the map is, the smaller the stress will be, but one should keep L as small as possible, ideally to match the intrinsic dimensionality of the data (in a visualisation application one will even force L to be less than 3, but for generic dimensionality reduction this is not necessary). Too small a dimension of the map can give a misleading view of the data. For example, points apparently clustered in a 2D map can actually lie far apart in a 3D one. For two-dimensional maps, a simple way to embed information from the original data in the map is to draw a line between every pair of objects whose proximity exceeds some threshold value: the presence of long, haphazardly crossing lines will indicate a discrepancy between closeness in the data and closeness in the space. Clusters will only be valid if they are consonant with the lines, i.e., points within a cluster should be well connected with each other and poorly connected with those outside the cluster. Sometimes the lines can connect many points in some nonlinear shape, like in figure 4.18, which suggests that only one curvilinear dimension would be enough to give a reasonable description of the data. This is called the *horseshoe phenomenon*.

To assure an adequate degree of statistical stability, the map dimension cannot be arbitrarily large for a given sample size. A rule of thumb often used in statistics (Kruskal and Wish, 1978) is that the number of

(significant) pairs of objects should be at least twice the number of parameters to be estimated:

$$\frac{N(N-1)}{2} \geq 2NL \Rightarrow N \geq 4L+1$$

but this just gives a trivial upper bound on L.

Another rule of thumb, analogous to the scree plot for PCA (plot of the cumulative eigenvalue sum versus the number of components) is to plot the stress obtained for map dimensions $L = 1, 2, 3\ldots$ and try to determine a cutoff point; for example, where the slope of the stress curve changes abruptly, or where the stress falls below a certain threshold. As in PCA, there is no guarantee that such heuristic rules can find the intrinsic dimensionality. Besides, while running PCA for different numbers of components is immediate, for MDS it can be quite time consuming.

4.10.1.2 Problems of MDS

- There is no foolproof method to select the appropriate dimension of the map; one must try several.
- MDS does a much better job in representing large distances (the global structure) than small ones (the local structure).
- Contrarily to principal component analysis, in MDS one cannot obtain an $(L-1)$-dimensional map out of an L-dimensional one by dropping one coordinate (or, in general, by linearly projecting along some direction). That is, it does not verify the property of additivity mentioned in section 2.6.2.

4.10.1.3 The Sammon mapping

With the objective of preserving the interpoint Euclidean distances of a collection of real vectors $\{\mathbf{t}_n\}_{n=1}^N$, Sammon (1969) proposed a particular type of MDS where the low-dimensional Euclidean space points $\{\mathbf{x}_n\}_{n=1}^N$ are chosen to minimise the criterion

$$E(\{\mathbf{x}_n\}_{n=1}^N) \stackrel{\text{def}}{=} \frac{1}{\sum_{n<m}^N \delta_{nm}} \sum_{n<m}^N \frac{(\delta_{nm}-d_{nm})^2}{\delta_{nm}}$$

where δ_{nm} is the distance in $\mathbb{R}^D$ between $\mathbf{t}_n$ and $\mathbf{t}_m$ and d_{nm} is the distance in $\mathbb{R}^L$ between $\mathbf{x}_n$ and $\mathbf{x}_m$. Unlike in usual MDS, this criterion gives weight to small distances, which helps to detect clusters. The Sammon mapping is the mapping implicitly defined by the pairs $\{(\mathbf{t}_n, \mathbf{x}_n)\}_{n=1}^N$. Sammon (1969) proposed a diagonal Newton method to minimise the criterion function:

$$\mathbf{x}_{nl}^{(\tau+1)} = \mathbf{x}_{nl}^{(\tau)} - \eta \frac{(\partial E/\partial x_{nl})^{(\tau)}}{(\partial^2 E/\partial x_{nl}^2)^{(\tau)}}$$

with a "magical factor" $\eta \approx 0.3$ or 0.4 and, as starting point for the $\{\mathbf{x}_n\}_{n=1}^N$, random values or the projections on the first L principal components of the data. This algorithm is a poor minimiser according to Ripley (1996), who notes that it often achieves very bad local minima from random starting points and that it is difficult to set η to achieve convergence.

4.10.2 Methods for preserving the geodetic distances

In the problem we are concerned with, dimensionality reduction of continuous data, the distance between points is the Euclidean distance in $\mathbb{R}^D$ (the length of the straight line segment joining both points), or perhaps some other definition of distance, which depends only on the point coordinates. However, this distance is not useful because it ignores the data manifold: two points that are close in the Euclidean distance in $\mathbb{R}^D$ may be far from each other inside the (low-dimensional) manifold defined by the data, as fig. 4.19 illustrates. Inputting such distances to an MDS or Sammon method would produce a wrong representation. The relevant distances are really the distances along the manifold, or more precisely, the geodetic distances. The **geodetic distance** between two points of a manifold is defined as a minimum of the length of a path joining both points that is contained in the manifold, and such minimal-length paths are called **geodesics**[19] (do Carmo, 1976).

[19] Actually, a geodesic is usually defined as a curve on a manifold that has zero acceleration on the manifold, i.e., that the acceleration vector (second derivative of the curve with respect to some parametrisation) is perpendicular to the manifold and therefore its orthogonal projection on it is zero. From this definition follow some interesting minimising properties such as the one mentioned.

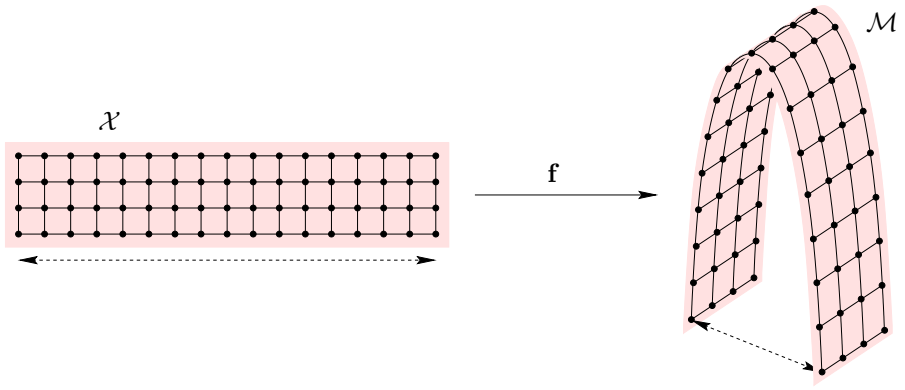


Figure 4.19: Distances along the straight line between two points (Euclidean distance) versus distances along a geodesic in the manifold (geodetic distance). Points close in the former sense may be far in the latter.

Depending on the manifold, there may be more than one geodesic between two given points (e.g. in a sphere, the two sections of a great circle passing through the two points are geodesics), but we will ignore that fact in this discussion.

Computing geodetic distances along arbitrary nonlinear manifolds is complicated, requiring the solution of the Euler-Lagrange equation for the arclength functional, which is a second order nonlinear differential equation. A computationally feasible approach consists of discretising the manifold as a Delaunay triangulated graph (Preparata and Shamos, 1985; Aurenhammer, 1991). Let us first define the Voronoi diagram. Given a collection of M vectors $\{\boldsymbol{\mu}_m\}_{m=1}^M$ in $\mathbb{R}^D$, the **Voronoi diagram** of $\mathbb{R}^D$ induced by those vectors is the collection of Voronoi cells $\{\mathcal{V}_m\}_{m=1}^M$ defined as

$$\mathcal{V}_m \stackrel{\text{def}}{=} \{\mathbf{t} \in \mathbb{R}^D : \ d(\mathbf{t}, \boldsymbol{\mu}_m) \leq d(\mathbf{t}, \boldsymbol{\mu}_n) \ \forall n \neq m\} \quad m = 1, \ldots, M$$

i.e., cell $\mathcal{V}_m$ is the set of points closest to $\boldsymbol{\mu}_m$ than to any other vector according to a certain distance d defined in $\mathbb{R}^D$. Each Voronoi cell is a convex D-dimensional polyhedron (a D-polytope) and the union of all cells is the space $\mathbb{R}^D$. The dual of the Voronoi diagram is the **Delaunay triangulation** of the collection $\{\boldsymbol{\mu}_m\}_{m=1}^M$, defined as the graph with vertices $\{v_m\}_{m=1}^M$ (vertex v_m being associated with vector $\boldsymbol{\mu}_m$) and adjacency matrix $\mathbf{A} = (a_{mn})$ verifying

$$a_{mn} = \begin{cases} 1, & \mathcal{V}_m \cap \mathcal{V}_n \neq \varnothing \\ 0, & \mathcal{V}_m \cap \mathcal{V}_n = \varnothing \end{cases}$$

i.e., two nodes are connected if and only if their associated Voronoi cells are adjacent. Analogous definitions are derived for a manifold $\mathcal{M}$ of $\mathbb{R}^D$ by taking the intersection of each Voronoi cell with $\mathcal{M}$; the collection of links of the resulting *restricted Delaunay triangulation* is a subset of that of the unrestricted Delaunay triangulation (for $\mathcal{M} \equiv \mathbb{R}^D$). The key point to note is that the Delaunay triangulation restricted to a manifold carries the topological structure of the manifold and that it enforces a discretisation of paths along the manifold that allows to compute (approximate) geodetic distances, as well as to plan paths from one point to another. Planning of paths through geodesics also offers interesting possibilities to applications where interpolation in data space is required, such as in morphing or animating an object from one three-dimensional configuration to another one (Bregler and Omohundro, 1995; Tenenbaum, 1998). Figure 4.20 illustrates these ideas.

In the rest of this section, we describe an algorithm to obtain the restricted Delaunay triangulation (topology-preserving networks) and a method that uses it to perform MDS with geodetic distances (ISOMAP).

4.10.2.1 Topology-preserving networks

Martinetz and Schulten (1994) give an algorithm to obtain the Delaunay triangulation induced on a manifold. They call the resultant graph a *topology-preserving network*. Their approach is opposite to that of self-organising maps: in SOMs, one fixes the links of the graph (which determine its topology: linear array, planar grid, etc.) and tries to fit it to the data manifold. This does not work if the dimensionality or the topology of the graph do not match those of the data manifold. The algorithm of Martinetz and Schulten (1994) works the other way, specifying the nodes and then constructing only the appropriate links, which leads to the graph. The algorithm, showed in fig. 4.21, requires as input a set of reference vectors $\{\boldsymbol{\mu}_m\}_{m=1}^M$ in data space which

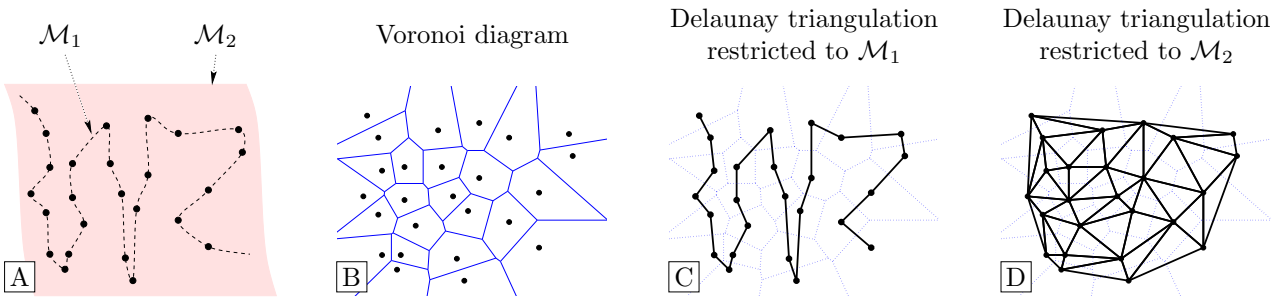


Figure 4.20: Voronoi diagram (graph B) and restricted Delaunay triangulation (graphs C and D). For a given collection of points (graph A), the restricted Delaunay triangulation depends on the manifold that embeds them: the dotted line $\mathcal{M}_1$ in graph C and the shaded area $\mathcal{M}_2$ in graph D. The points and the manifolds on graph A are the same as those of fig. 4.5.

inputs
  $\{\boldsymbol{\mu}_m\}_{m=1}^M \subset \mathcal{M} \subseteq \mathbb{R}^D$ uniformly distributed on $\mathcal{M}$ — Reference vectors
  $\mathcal{U}(\mathcal{M})$ — Uniform distribution on $\mathcal{M}$
initialise
  Graph with nodes $\{v_m\}_{m=1}^M$ and no links — Topology-preserving network
repeat forever
  Draw $\mathbf{t}$ from $\mathcal{U}(\mathcal{M})$
  $m_1 \leftarrow \arg\min_{m=1,\ldots,M} \|\mathbf{t} - \boldsymbol{\mu}_m\|$ — Closest node
  $m_2 \leftarrow \arg\min_{\substack{m=1,\ldots,M \\ m \neq m_1}} \|\mathbf{t} - \boldsymbol{\mu}_m\|$ — Second closest node
  Connect nodes v_{m_1} and v_{m_2}
end
return graph

Figure 4.21: Pseudocode of the construction algorithm for topology-preserving networks of Martinetz and Schulten (1994).

is *uniformly* distributed over the data manifold. The reason for this requirement is that the reference vectors must define the shape of the manifold. To each reference vector $\boldsymbol{\mu}_m$ we associate a node v_m in a graph in which initially there are no links. To derive the appropriate links, the algorithm uses an online rule that they term competitive Hebbian: given a point drawn uniformly from the data manifold $\mathcal{M}$, connect the nodes associated with the two reference vectors closest to it. The rule is competitive because only the link between the two winners (the two closest reference vectors) is updated; and it is Hebbian because, if we assume that each node has an activity proportional to the proximity of its associated reference vector to the data point presented, then the connection between two nodes is strengthened if both nodes have a high activity. It is intuitively clear that this must converge to the Delaunay triangulation restricted to $\mathcal{M}$ and indeed Martinetz and Schulten (1994) prove so under the assumption that the reference vectors are (using their nomenclature) *dense* in the data manifold $\mathcal{M}$: for any point $\mathbf{t}$ in $\mathcal{M}$, the triangle formed by $\mathbf{t}$ and the two closest reference vectors must be contained in $\mathcal{M}$. However, this definition is only appropriate for manifolds of the same dimensionality as the embedding space $\mathbb{R}^D$, since it trivially holds for convex manifolds of any dimensionality and never holds for nonconvex manifolds of lower dimensionality.

Clearly, if the data manifold coincides with the data space then an unrestricted Delaunay triangulation is obtained. While we are concerned here with the Delaunay triangulation restricted to a low-dimensional data manifold, the unrestricted Delaunay triangulation is interesting in its own right for many other situations, such as the finite-element method and other geometric and graph-theoretical problems mentioned by Martinetz and Schulten (1994).

The number of links from a node in a fully-connected graph of M nodes is $M-1$, which leads to $\mathcal{O}(M^2)$ for the total number of links in the graph. But if only local connections are allowed, as in the Delaunay triangulation, the average number of links from a node becomes proportional to L (the dimensionality of the

graph), and so the total number of links in the graph becomes $\mathcal{O}(LM)$. In reality, this is $\mathcal{O}(Le^L)$ since the number of reference vectors necessary to represent an L-dimensional manifold is exponential in L.

The algorithm of Martinetz and Schulten (1994) has several problems:

- There is no way to assess convergence. The fact that points are blindly drawn from a uniform distribution on $\mathcal{M}$ means that a lot of them will be redundant, since they will establish links already done and we may need to wait very long until a point is drawn that establishes a necessary link. We presume that the algorithm will build a proportion of the Delaunay links early during training, with the remaining ones appearing very slowly afterwards—the slower the higher the manifold dimensionality is. Thus, the graph obtained when the algorithm is stopped after a certain number of iterations will probably contain many missing links if the number of reference vectors is large. This problem would disappear if there was a way of setting to zero the probability of the manifold region affecting a link just created (so that no further samples are drawn there).

  In practice, a finite data sample $\{\mathbf{t}_n\}_{n=1}^N$ is used, which is all the information we have about the data manifold. This ensures that the algorithm stops after N point presentations, but the resulting graph is affected by the factors mentioned below.

- It is not easy to distribute uniformly the reference vectors over an arbitrary manifold, in particular when the manifold is defined by a data sample (our case). Directly using a data sample (that, ignoring the noise for the moment, belongs to the data manifold by definition) is unlikely to work in general, since the data distribution over the manifold need not be uniform (e.g. see the sample in fig. 2.13). This means that the triangulation found will be too coarse in low probability areas of the manifold and far too dense in high-probability areas.

- It is difficult to decide how many reference vectors to use and the final triangulation is very sensitive to this. The number of reference vectors is intuitively related to the Nyquist (spatial) frequency, since they are effectively a sample of $\mathcal{M}$ from which $\mathcal{M}$ could be reconstructed in principle. It also must depend exponentially on L, as the volume of $\mathcal{M}$ does. This is again a result of the curse of the dimensionality and is analogous to the situation found with the Monte Carlo sampling of the latent space in latent variable models (section 2.4).

  For storage efficiency and speed of training, the number of reference vectors should be kept as small as possible. This means selecting a small subset of the data sample—which may severely reduce the number of reference vectors in low-probability areas of the manifold. Thus, a critical point of the algorithm is to select a subset of M points from the data sample such that it spans the data manifold uniformly and M is neither too large (which would lead to many missing links in the graph) nor too small (which would yield a poor approximation of the manifold and of the geodetic distances).

- Sensitivity to noise. The reference vectors are assumed to lie in the data manifold, i.e., the noise is considered zero—a clearly untenable assumption in many practical problems. If the noise level is high enough many points will fall out of the data manifold (if the noise was such that the points were perturbed inside the manifold there would be no problem, of course, but this is never going to happen). Such points, in particular outliers, will result in the establishment of links between reference vectors which are far away in the manifold, distorting the topological structure of the graph. Thus we expect the graph representation to degrade ungracefully with the noise level.

4.10.2.2 The ISOMAP algorithm

Tenenbaum (1998) has proposed a straightforward combination of the topology-preserving network algorithm with multidimensional scaling for manifold modelling that he calls ISOMAP: (1) to use ordinal multidimensional scaling with the geodetic interpoint distances computed with the method of Martinetz and Schulten (1994) to derive a low-dimensional Euclidean map and then (2) fit a radial basis function network to the mapping from the low-dimensional points to the observed ones or vice versa, thus defining reconstruction and dimensionality reduction mappings, respectively. In particular, ISOMAP works as follows. Given a sample $\{\mathbf{t}_n\}_{n=1}^N \subset \mathbb{R}^D$, it randomly selects M reference vectors from it and constructs an approximation to the restricted Delaunay triangulation graph whose nodes are associated with those M reference vectors, using the method of Martinetz and Schulten (1994). The infinite sequence of points uniformly distributed over the data manifold that this algorithm requires is approximated by the data sample $\{\mathbf{t}_n\}_{n=1}^N$. Once such graph has been obtained, it is used to compute the geodetic distances between every two reference vectors using Floyd's

algorithm, whose complexity is $\mathcal{O}(M^3)$. Tenenbaum claims to be able to determine the intrinsic dimensionality of the data manifold by trying several dimensions and plotting the stress obtained, but, as mentioned in section 4.10.1.1, this heuristic guarantees very little. ISOMAP then uses an RBF network to implement the mapping between reference vectors and the low-dimensional points found by the ordinal MDS.

ISOMAP is a promising method, but it inherits the problems of all the methods it is based on, namely the lack of guarantee that the reference vectors uniformly span the data manifold[20] and the potential problems of local minima when approximating implicitly defined mappings via a universal mapping approximator. It also has a high computational complexity: $\mathcal{O}(M^3)$ to compute the $\mathcal{O}(M^2)$ geodetic distances, where the number of reference vectors M is $\mathcal{O}(e^L)$, plus several trial-and-error MDS runs on those distances (till a convenient dimension L is determined), plus the final RBF fit. Tenenbaum (1998) claims that the geodetic distances computed from the graph are very good approximations to the true ones for some toy examples, although the distances are slightly overestimated due to the discretisation of the manifold. Since the version of MDS used, coincident with eq. (4.4), represents large distances much better than small ones, as mentioned in section 4.10.1.2, the net effect may be a misrepresentation of distances at all scales.

Tenenbaum et al. (2000) present a variation of ISOMAP where (1) the topology-preserving network is constructed, instead of with the algorithm of Martinetz and Schulten (1994), simply by linking every data point with its K nearest neighbouring data points (or with all other data points within a distance ϵ of itself), with each link weighted by the corresponding distance; and (2) an easy-to-minimise stress function with a single minimum is used, akin to classical scaling (solvable via the principal components of a matrix derived from the geodetic distances). Tenenbaum et al. (2000) state that, in the limit of infinite training data, this new scheme recovers the true dimensionality and geometric structure of the data manifold if this belongs to a certain class of Euclidean manifolds (which excludes manifolds such as hemispheres and tori). They give a proof based on the fact that as training data increases, the graph becomes denser and better approximates the true geodetic distances; the actual convergence rate depends on unknowns such as the curvature of the data manifold, the separation between branches and the data density. Unfortunately, in practice this proof is of little use because, generally, training data in high dimensions will be scarce and not uniformly distributed on the manifold; and the computational complexity (at least quadratic in the number of data points) would preclude using many data points anyway. Also importantly, step (1) now crucially depends on the neighbourhood size (K or ϵ): if too large, it will include data points from other branches of the manifold, shortcutting them and resulting in wrong geodetic distances; if too small, it may not contain enough neighbours. Determining a good neighbourhood size may be difficult in practice. Finally, while (2) is faster than an arbitrary MDS (metric or ordinal), it is also more restrictive: its unique minimum may not be as good as some of the local minima of an arbitrary MDS. Other problems mentioned earlier remain: sensitivity to noise, bad local minima when explicitly learning a dimensionality reduction mapping and no foolproof way to determine the intrinsic dimensionality.

ISOMAP also needs a more thorough evaluation than the fact that it seemed to work in the examples given by Tenenbaum (1998) and Tenenbaum et al. (2000). Particularly dubious are the results regarding the problem of finding the manifold spanned by a set of bitmapped face images in various azimuth and elevation view angles, where ISOMAP magically picks the (to a human perceptually salient) azimuth and elevation degrees of freedom but not the also prominent changes in illumination and translation and the noise. The data set used (10 000 images) is also small given the dimensionality of the images (32×32). Similar criticism applies to the manifold spanned by the images of a hand undergoing non-rigid articulated motion, which cannot be modelled with only two degrees of freedom (Amir Assadi, personal communication).

4.10.2.3 Summary

Representing a manifold by a skeleton graph from a noisy sample from the manifold includes two steps:

- finding an approximate set of vectors that are uniformly spread on the manifold
- finding the restricted Delaunay triangulation of those points.

The algorithm of Martinetz and Schulten (1994) is a first step toward this objective. This skeleton graph can then be used to compute pairwise geodetic distances, feed them into an ordinal MDS method and solve the nonlinear regression between the low-dimensional map and the high-dimensional sample with a universal

[20] The toy examples offered by Tenenbaum (1998), as well as those by Martinetz and Schulten (1994) and Tenenbaum et al. (2000), have the data sample uniformly distributed on the manifold by definition. In this ideal case, it is easy to obtain a collection of reference vectors uniformly spread over the manifold by applying a vector quantisation algorithm (Martinetz and Schulten (1994) use the neural gas) or even by simply choosing a random subset of a data sample.

mapping approximator. This is the basis of the ISOMAP procedure of Tenenbaum (1998). Defining manifolds via the geodetic distance is an exciting idea and a promising avenue for further research.

Preservation of distances can also be imposed on other models, for example GTM. Tenenbaum (1998) and Marrs and Webb (1999) have criticised the GTM model because it often finds estimates that offer a distorted view of the high-dimensional manifold. That is, the distances along the data manifold in data space are stretched or shrunk in a complex way in different amounts in different regions of the latent space—even though the mapping $\mathbf{f}$ may be approximating well the data manifold. In section 2.8.3 we mentioned the disadvantages that this has for visualisation of structure in the data, even though such a distorted coordinate system is as valid as any other to which it can be invertibly mapped. Anyway, it would be good to be able to enforce the preservation of geodetic distances in GTM. Marrs and Webb (1999) try to achieve this via a regularisation term in the log-likelihood that constrains GTM's mapping from latent onto data space to be unit-speed on the average, so that unit steps in latent space correspond to unit steps in data space on the average. We discussed this approach in section 2.8.3.

4.10.3 Locally linear embedding

Roweis and Saul (2000) have recently proposed a dimensionality reduction method that they call *locally linear embedding* (LLE). Given a collection $\{\mathbf{t}_n\}_{n=1}^N$ of training points, it uses a linear mapping to capture local neighbourhood relations—representative of the local geometry of the data manifold—that are then preserved as much as possible in an associated low-dimensional collection of points $\{\mathbf{x}_n\}_{n=1}^N$, which is the end product, just as in MDS. Specifically, each data point is reconstructed by least squares as a linear combination of its K nearest neighbouring data points (or of all other data points within a distance ϵ of itself):

$$\hat{\mathbf{t}}_n = \sum_{m\in\mathcal{N}(\mathbf{t}_n)} w_{nm}\mathbf{t}_n \qquad \min E_1\left(\{w_{nm}\}\right) \stackrel{\text{def}}{=} \sum_{n=1}^N \left\|\mathbf{t}_n - \hat{\mathbf{t}}_n\right\|^2 \text{ subject to } \sum_{m\in\mathcal{N}(\mathbf{t}_n)} w_{nm} = 1$$

where $\mathcal{N}(\mathbf{t}_n)$ is the set of neighbours of data point $\mathbf{t}_n$. The least-squares problem is solved for the optimal weights w_{nm}^* as N separate linear systems of $K\times K$ equations, regularised if $K < D$ (in which case the systems are underdetermined). The error function E_1 is invariant to translations (since the weights sum one), rotations and uniform scalings of the neighbourhoods. Ideally, minimising E_1 results in the weights characterising local, intrinsic geometric properties of the data manifold. Such properties can then be preserved in a low-dimensional representation of the data by imposing on it the weights, again as an objective function to be minimised by least-squares, but this time over the low-dimensional points $\{\mathbf{x}_n\}_{n=1}^N$ for constant weights:

$$\min E_2\left(\{\mathbf{x}_n\}_{n=1}^N\right) \stackrel{\text{def}}{=} \sum_{n=1}^N \left\|\mathbf{x}_n - \hat{\mathbf{x}}_n\right\|^2 \qquad \hat{\mathbf{x}}_n = \sum_{m\in\mathcal{N}(\mathbf{t}_n)} w_{nm}^*\mathbf{x}_n.$$

The points $\{\mathbf{x}_n\}_{n=1}^N$ are constrained to be zero-mean and unit-covariance to eliminate arbitrary translations and rotations. The minimisation is then a constrained quadratic programming problem with a unique minimum that can be found by solving an eigenvalue problem. Note that it cannot be decomposed into an independent subprogram for each n as before, i.e., the local neighbourhood relations interact. However, it is additive in the same sense PCA is: the map with $L+1$ dimensions is obtained from the one with L by computing the next eigenvalue. The method can also be run using as input the pairwise distances between data points instead of the data points themselves, just as in MDS. As usual, a mapping approximator can be fit to the pairs $\{(\mathbf{t}_n, \mathbf{x}_n)\}_{n=1}^N$ to obtain a dimensionality reduction mapping. This will define a single global mapping, unlike the local dimensionality reduction methods of section 4.7.

LLE is attractively simple and has similarities with MDS methods, in particular with ISOMAP. In fact, it can be considered as a topography-preserving MDS, where the local topography is preserved by linear neighbourhood relations rather than by the pairwise distances. It also shares some of the disadvantages of MDS methods: (1) likely sensitivity to noise and to the training set used; (2) sensitivity to data density that is not uniform on the data manifold (which makes difficult to obtain local neighbourhoods in low-density regions); (3) no dimensionality reduction mapping is defined, the only way being to fit a mapping approximator to the pairs $\{(\mathbf{t}_n, \mathbf{x}_n)\}_{n=1}^N$, which is prone to bad local minima; (4) quadratic complexity on the training set size, both when computing the neighbourhood weights and when solving the eigenvalue problem for the low-dimensional points; (5) no foolproof way to determine the intrinsic dimensionality; and finally and crucially, (6) no foolproof way to determine the neighbourhood size K, which determines how local the geometric properties captured by the weights are. Locality implies a small neighbourhood (more so since the implementation is linear) but not so small that no neighbours are available.

Roweis and Saul (2000) apply LLE to a toy problem, to a face images dataset like that of ISOMAP and to a word categorisation example. More thorough evaluations are necessary that elucidate the impact of the caveats mentioned.

4.11 Conclusions

We have defined the problem of dimensionality reduction as the search for an economic coordinate representation of a submanifold of a high-dimensional Euclidean space, a problem so far not yet solved in a satisfactory and general way. We have then given an overview of several techniques for dimensionality reduction.

In general, the central place occupied by PCA has not been taken by any nonlinear technique. PCA remains the favourite feature extractor, particularly for very high-dimensional data (such as images), where it is typically used in a preprocessing stage. Projection pursuit is another linear dimensionality reduction technique, guided by an arbitrary criterion rather than by PCA's (maximal variance or, equivalently, minimal L_2-error). Kernel PCA is an attractively simple nonlinear extension of PCA that has performed well in some applications as feature extractor. However, more complete performance comparisons are necessary, and more importantly some theoretical insight in the interpretation of the nonlinear components, the effect of the kernel and the definition of dimensionality reduction and reconstruction mappings.

Local dimensionality reduction, particularly PCA-based, is a good and reasonably fast approach that combines some of the benefits of PCA with a nonlinear model: economy of parameters and flexibility. However, it is not very accurate unless many local models are used. Global nonlinear dimensionality reduction (e.g. autoassociators) requires many parameters and training is difficult, requiring much time and data and being very prone to bad local minima.

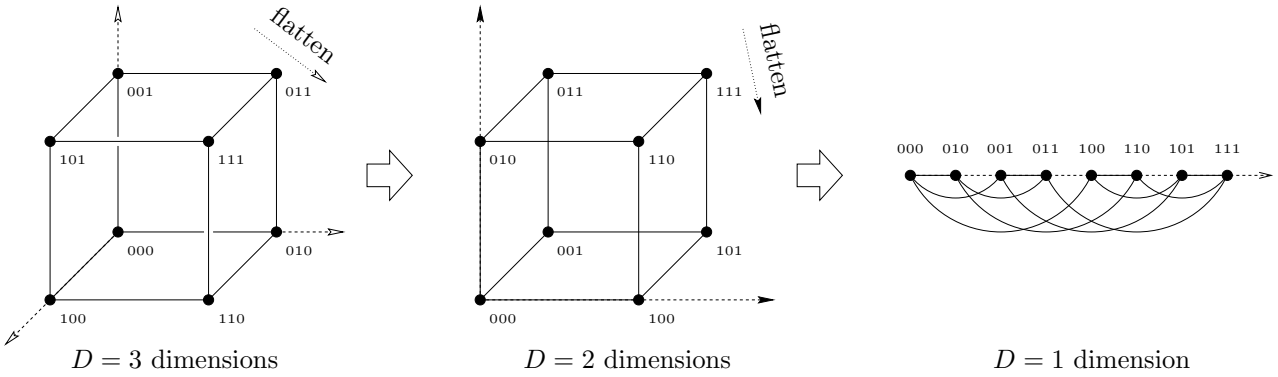
The definition of principal curves makes them intuitively appealing. However, while they have been used for small dimensions (Hastie and Stuetzle, 1989; Banfield and Raftery, 1992), they are not mature enough algorithmically for high-dimensional dimensionality reduction.

Two classes of methods, those based on topological vector quantisation (section 4.9) and those based on MDS with either straight-line or geodetic distances (section 4.10), have as primary product a collection of reference vectors in observed space and an associated collection of low-dimensional vectors preserving topology and metric structure, respectively. The disadvantage of these methods is that the data manifold and the dimensionality reduction and reconstruction mappings are defined implicitly through the pairs $\{(\boldsymbol{\mu}_m, \mathbf{x}_m)\}_{m=1}^M$ (for vector quantisation) or $\{(\mathbf{t}_n, \mathbf{x}_n)\}_{n=1}^N$ (for MDS). Defining such mappings explicitly requires fitting in a supervised way a mapping approximator to that set of pairs. In an ideal situation, the dimensionality of the $\mathcal{X}$ space is appropriate, and the associations $\mathbf{x}_m \leftrightarrow \mathbf{t}_m$ are correct and are instances of a one-to-one mapping $\mathcal{X} \underset{\mathbf{F}}{\overset{\mathbf{f}}{\longleftrightarrow}} \mathcal{M} \subset \mathcal{T}$ (a bijection). But even in this best case, fitting a global flexible model to either of the mappings, $\mathbf{f}$ or $\mathbf{F}$, is likely to end up in bad local minima, depending sensitively on the starting point of the optimiser, the optimiser itself and the architecture of the approximator. Some knowledge about the manifold must be injected, probably via a regularisation term in the objective function, but it is not clear how well the final mappings will generalise to unseen data. For MDS it is possible to define the low-dimensional vectors via an explicit dimensionality reduction mapping, although the problem of bad local minima is still serious. Two further problems of distance-based methods are (1) their high computational complexity, at least quadratic on the training set size; and (2) the fact that the training data should be uniformly distributed over the data manifold, since such methods try to model the geometry of that manifold—for which the training set acts as a scaffold—but not the density of the data.

None of the methods discussed in this chapter define a density model for the data (except PCA and the elastic net in their probabilistic interpretation). While this is not necessary to obtain dimensionality reduction and reconstruction mappings, it is to know the distribution over the manifold and the latent space. Self-organising maps give an indication of the density over the manifold via the distribution of the reference vectors on it, but as mentioned in section 4.9.1 they do not properly define a density model. The advantages of probabilistic methods over non-probabilistic ones are discussed in chapter 11. One somewhat surprising property of dimensionality reduction with probabilistic models is that, due to the noise model, the dimensionality reduction and reconstruction mappings are not the inverse of each other (specifically, $\mathbf{F} \circ \mathbf{f}$ is not the identity, as happens for probabilistic PCA; see section 2.9.1). However, basing the dimensionality reduction on an Euclidean distance criterion (to the low-dimensional, nonlinear manifold) leads to discontinuous mappings.

Two central ideas underlie several of the methods:

- Points close together in data space should be mapped close together in low-dimensional space (vector



$D = 3$ dimensions $D = 2$ dimensions $D = 1$ dimension

Figure 4.22: The topological structure of a quantised high-dimensional Euclidean space can be preserved in lower dimensions (even one dimension) by *folding* or *flattening* it and preserving the connections. In the low-dimensional representation, the connections still reflect the topology of the high-dimensional space, not of the low-dimensional one. The representation can be further generalised by labelling the connections with *distance* or *similarity* values.

quantisation methods, such as self-organising maps and the elastic net, and MDS methods).

- The manifold should pass through the middle of the data (principal curves, nonprobabilistic PCA).

Two major issues that remain open are:

- To overcome the curse of the dimensionality, which results in many parameters that demand huge sample sizes to obtain reasonable results. Most of the techniques reviewed still suffer of this to an extent.
- To determine the intrinsic dimensionality of a distribution given a sample of it. Knowing it would reduce the possibility of over- or underfitting. Model selection techniques, such as those based on the value of an error criterion as a function of the number of dimensions used (as in PCA or MDS), are unreliable.

4.12 Can dimensionality reduction be achieved with discrete variables?

We have discussed dimensionality reduction of continuous observed variables in terms of continuous latent variables, as in PCA, autoassociators or methods based on multidimensional scaling. This is due to the fact that topographic mappings are naturally defined in continuous Euclidean spaces thanks to their intrinsic definition of distance, whatever the dimensionality of the space. Discrete spaces can work like continuous spaces by assuming that they are a sample or discretisation of an underlying continuous space and ensuring that the learning algorithm respects the topographic structure of that space, as happens with GTM or self-organising maps. In such cases, the higher the resolution of the sample, the better the collection of discrete values will approximate the underlying continuous space. Therefore, this use of discrete variables is essentially not different from using continuous latent variables—it is just that we cannot use high-dimensional continuous variables directly in an analytically exact, or otherwise desirable, way. In particular, the dimension of the manifold induced in observed space is the same as that of the underlying low-dimensional continuous space (e.g. 2 for a two-dimensional grid in a self-organising map).

We mention here a different possibility, without going into any detail of how it could actually be implemented. If a Euclidean space of dimension D is discretised into regions, we can represent the neighbourhood relations between these regions (and thus their topological structure) with a graph, as in fig. 4.22(left). This graph, which is D-dimensional, can be flattened to any lower dimension, up to 1 (fig. 4.22(right)) while still keeping the topological information of the D-dimensional space via the graph edges. In the figure, the arrangement on the right is apparently one-dimensional but can be unfolded to reveal a three-dimensional structure; this is revealed by the presence of long-range connections in the one-dimensional arrangement (extending farther than the one-dimensional nearest neighbours). The graph edges can be labelled with numerical values representing distance or similarity.

One cannot help comparing this to the complex pattern of short- and long-range connections existing in different areas of the brain. The primary visual cortex of mammals, usually considered as a two-dimensional

sheet of neurons, is known to have orderly representations of a number of external stimuli, such as position in the visual field, eye of origin, orientation, direction of movement, spatial frequency and disparity. In fact, dimensionality reduction models such as self-organising maps and the elastic net have been very successful at replicating such organisation (Swindale, 1996). In the hippocampus, the pattern of connections is more complex, with no two-dimensional topography, which might indicate that it is coding higher dimensions. It would be interesting to investigate the alternative possibility of a discrete, two-dimensional collection of neurons that actually unfolds into a higher-dimensional stimuli space.

Bibliography

S. Amari. Natural gradient learning for over- and under-complete bases in ICA. *Neural Computation*, 11(8): 1875–1883, Nov. 1999.

S. Amari and A. Cichoki. Adaptive blind signal processing—neural network approaches. *Proc. IEEE*, 86(10): 2026–2048, Oct. 1998.

T. W. Anderson. Asymptotic theory for principal component analysis. *Annals of Mathematical Statistics*, 34 (1):122–148, Mar. 1963.

T. W. Anderson and H. Rubin. Statistical inference in factor analysis. In J. Neyman, editor, *Proc. 3rd Berkeley Symp. Mathematical Statistics and Probability*, volume V, pages 111–150, Berkeley, 1956. University of California Press.

S. Arnfield. Artificial EPG palate image. The Reading EPG, 1995. Available online at `http://www.linguistics.reading.ac.uk/research/speechlab/epg/palate.jpg`, Feb. 1, 2000.

H. Asada and J.-J. E. Slotine. *Robot Analysis and Control*. John Wiley & Sons, New York, London, Sydney, 1986.

D. Asimov. The grand tour: A tool for viewing multidimensional data. *SIAM J. Sci. Stat. Comput.*, 6:128–143, 1985.

B. S. Atal, J. J. Chang, M. V. Mathews, and J. W. Tukey. Inversion of articulatory-to-acoustic transformation in the vocal tract by a computer-sorting technique. *J. Acoustic Soc. Amer.*, 63(5):1535–1555, May 1978.

C. G. Atkeson. Learning arm kinematics and dynamics. *Annu. Rev. Neurosci.*, 12:157–183, 1989.

H. Attias. EM algorithms for independent component analysis. In Niranjan (1998), pages 132–141.

H. Attias. Independent factor analysis. *Neural Computation*, 11(4):803–851, May 1999.

F. Aurenhammer. Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM Computing Surveys*, 23(3):345–405, Sept. 1991.

A. Azzalini and A. Capitanio. Statistical applications of the multivariate skew-normal distribution. *Journal of the Royal Statistical Society, B*, 61(3):579–602, 1999.

R. J. Baddeley. Searching for filters with "interesting" output distributions: An uninteresting direction to explore? *Network: Computation in Neural Systems*, 7(2):409–421, 1996.

R. Bakis. Coarticulation modeling with continuous-state HMMs. In *Proc. IEEE Workshop Automatic Speech Recognition*, pages 20–21, Arden House, New York, 1991. Harriman.

R. Bakis. An articulatory-like speech production model with controlled use of prior knowledge. Frontiers in Speech Processing: Robust Speech Analysis '93, Workshop CDROM, NIST Speech Disc 15 (also available from the Linguistic Data Consortium), Aug. 6 1993.

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

J. D. Banfield and A. E. Raftery. Ice floe identification in satellite images using mathematical morphology and clustering about principal curves. *J. Amer. Stat. Assoc.*, 87(417):7–16, Mar. 1992.

J. Barker, L. Josifovski, M. Cooke, and P. Green. Soft decisions in missing data techniques for robust automatic speech recognition. In *Proc. of the International Conference on Spoken Language Processing (ICSLP'00)*, Beijing, China, Oct. 16–20 2000.

J. P. Barker and F. Berthommier. Evidence of correlation between acoustic and visual features of speech. In Ohala et al. (1999), pages 199–202.

M. F. Barnsley. *Fractals Everywhere*. Academic Press, New York, 1988.

D. J. Bartholomew. The foundations of factor analysis. *Biometrika*, 71(2):221–232, Aug. 1984.

D. J. Bartholomew. Foundations of factor analysis: Some practical implications. *Brit. J. of Mathematical and Statistical Psychology*, 38:1–10 (discussion in pp. 127–140), 1985.

D. J. Bartholomew. *Latent Variable Models and Factor Analysis*. Charles Griffin & Company Ltd., London, 1987.

A. Basilevsky. *Statistical Factor Analysis and Related Methods*. Wiley Series in Probability and Mathematical Statistics. John Wiley & Sons, New York, London, Sydney, 1994.

H.-U. Bauer, M. Herrmann, and T. Villmann. Neural maps and topographic vector quantization. *Neural Networks*, 12(4–5):659–676, June 1999.

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

J. Behboodian. On the modes of a mixture of two normal distributions. *Technometrics*, 12(1):131–139, Feb. 1970.

A. J. Bell and T. J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129–1159, 1995.

A. J. Bell and T. J. Sejnowski. The "independent components" of natural scenes are edge filters. *Vision Res.*, 37(23):3327–3338, Dec. 1997.

R. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, 1957.

R. Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, Princeton, 1961.

Y. Bengio and F. Gingras. Recurrent neural networks for missing or asynchronous data. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems*, volume 8, pages 395–401. MIT Press, Cambridge, MA, 1996.

C. Benoît, M.-T. Lallouache, T. Mohamadi, and C. Abry. A set of French visemes for visual speech synthesis. In G. Bailly and C. Benoît, editors, *Talking Machines: Theories, Models and Designs*, pages 485–504. North Holland-Elsevier Science Publishers, Amsterdam, New York, Oxford, 1992.

P. M. Bentler and J. S. Tanaka. Problems with EM algorithms for ML factor analysis. *Psychometrika*, 48(2): 247–251, June 1983.

J. O. Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer Series in Statistics. Springer-Verlag, Berlin, second edition, 1985.

M. Berkane, editor. *Latent Variable Modeling and Applications to Causality*. Number 120 in Springer Series in Statistics. Springer-Verlag, Berlin, 1997.

J. M. Bernardo and A. F. M. Smith. *Bayesian Theory*. Wiley Series in Probability and Mathematical Statistics. John Wiley & Sons, Chichester, 1994.

N. Bernstein. *The Coordination and Regulation of Movements*. Pergamon, Oxford, 1967.

D. P. Bertsekas. *Dynamic Programming. Deterministic and Stochastic Models*. Prentice-Hall, Englewood Cliffs, N.J., 1987.

J. Besag and P. J. Green. Spatial statistics and Bayesian computation. *Journal of the Royal Statistical Society, B*, 55(1):25–37, 1993.

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

E. L. Bienenstock, L. N. Cooper, and P. W. Munro. Theory for the development of neuron selectivity: Orientation specificity and binocular interaction in visual cortex. *J. Neurosci.*, 2(1):32–48, Jan. 1982.

C. M. Bishop. Mixture density networks. Technical Report NCRG/94/004, Neural Computing Research Group, Aston University, Feb. 1994. Available online at `http://www.ncrg.aston.ac.uk/Papers/postscript/NCRG_94_004.ps.Z`.

C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, New York, Oxford, 1995.

C. M. Bishop. Bayesian PCA. In Kearns et al. (1999), pages 382–388.

C. M. Bishop, G. E. Hinton, and I. G. D. Strachan. GTM through time. In *IEE Fifth International Conference on Artificial Neural Networks*, pages 111–116, 1997a.

C. M. Bishop and I. T. Nabney. Modeling conditional probability distributions for periodic variables. *Neural Computation*, 8(5):1123–1133, July 1996.

C. M. Bishop, M. Svensén, and C. K. I. Williams. Magnification factors for the SOM and GTM algorithms. In *WSOM'97: Workshop on Self-Organizing Maps*, pages 333–338, Finland, June 4–6 1997b. Helsinki University of Technology.

C. M. Bishop, M. Svensén, and C. K. I. Williams. Developments of the generative topographic mapping. *Neurocomputing*, 21(1–3):203–224, Nov. 1998a.

C. M. Bishop, M. Svensén, and C. K. I. Williams. GTM: The generative topographic mapping. *Neural Computation*, 10(1):215–234, Jan. 1998b.

C. M. Bishop and M. E. Tipping. A hierarchical latent variable model for data visualization. *IEEE Trans. on Pattern Anal. and Machine Intel.*, 20(3):281–293, Mar. 1998.

A. Bjerhammar. *Theory of Errors and Generalized Matrix Inverses*. North Holland-Elsevier Science Publishers, Amsterdam, New York, Oxford, 1973.

C. S. Blackburn and S. Young. A self-learning predictive model of articulator movements during speech production. *J. Acoustic Soc. Amer.*, 107(3):1659–1670, Mar. 2000.

T. L. Boullion and P. L. Odell. *Generalized Inverse Matrices*. John Wiley & Sons, New York, London, Sydney, 1971.

H. Bourlard and Y. Kamp. Autoassociation by the multilayer perceptrons and singular value decomposition. *Biol. Cybern.*, 59(4–5):291–294, 1988.

H. Bourlard and N. Morgan. *Connectionist Speech Recognition. A Hybrid Approach*. Kluwer Academic Publishers Group, Dordrecht, The Netherlands, 1994.

M. Brand. Structure learning in conditional probability models via an entropic prior and parameter extinction. *Neural Computation*, 11(5):1155–1182, July 1999.

C. Bregler and S. M. Omohundro. Surface learning with applications to lip-reading. In Cowan et al. (1994), pages 43–50.

C. Bregler and S. M. Omohundro. Nonlinear image interpolation using manifold learning. In Tesauro et al. (1995), pages 973–980.

L. J. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, Aug. 1996.

S. P. Brooks. Markov chain Monte Carlo method and its application. *The Statistician*, 47(1):69–100, 1998.

C. P. Browman and L. M. Goldstein. Articulatory phonology: An overview. *Phonetica*, 49(3–4):155–180, 1992.

E. N. Brown, L. M. Frank, D. Tang, M. C. Quirk, and M. A. Wilson. A statistical paradigm for neural spike train decoding applied to position prediction from ensemble firing patterns of rat hippocampal place cells. *J. Neurosci.*, 18(18):7411–7425, Sept. 1998.

G. J. Brown and M. Cooke. Computational auditory scene analysis. *Computer Speech and Language*, 8(4): 297–336, Oct. 1994.

D. Byrd, E. Flemming, C. A. Mueller, and C. C. Tan. Using regions and indices in EPG data reduction. *Journal of Speech and Hearing Research*, 38(4):821–827, Aug. 1995.

J.-F. Cardoso. Infomax and maximum likelihood for blind source separation. *IEEE Letters on Signal Processing*, 4(4):112–114, Apr. 1997.

J.-F. Cardoso. Blind signal separation: Statistical principles. *Proc. IEEE*, 86(10):2009–2025, Oct. 1998.

M. Á. Carreira-Perpiñán. A review of dimension reduction techniques. Technical Report CS–96–09, Dept. of Computer Science, University of Sheffield, UK, Dec. 1996. Available online at `http://www.dcs.shef.ac.uk/~miguel/papers/cs-96-09.html`.

M. Á. Carreira-Perpiñán. Density networks for dimension reduction of continuous data: Analytical solutions. Technical Report CS–97–09, Dept. of Computer Science, University of Sheffield, UK, Apr. 1997. Available online at `http://www.dcs.shef.ac.uk/~miguel/papers/cs-97-09.html`.

M. Á. Carreira-Perpiñán. Mode-finding for mixtures of Gaussian distributions. Technical Report CS–99–03, Dept. of Computer Science, University of Sheffield, UK, Mar. 1999a. Revised August 4, 2000. Available online at `http://www.dcs.shef.ac.uk/~miguel/papers/cs-99-03.html`.

M. Á. Carreira-Perpiñán. One-to-many mappings, continuity constraints and latent variable models. In *Proc. of the IEE Colloquium on Applied Statistical Pattern Recognition*, Birmingham, UK, 1999b.

M. Á. Carreira-Perpiñán. Mode-finding for mixtures of Gaussian distributions. *IEEE Trans. on Pattern Anal. and Machine Intel.*, 22(11):1318–1323, Nov. 2000a.

M. Á. Carreira-Perpiñán. Reconstruction of sequential data with probabilistic models and continuity constraints. In Solla et al. (2000), pages 414–420.

M. Á. Carreira-Perpiñán and S. Renals. Dimensionality reduction of electropalatographic data using latent variable models. *Speech Communication*, 26(4):259–282, Dec. 1998a.

M. Á. Carreira-Perpiñán and S. Renals. Experimental evaluation of latent variable models for dimensionality reduction. In Niranjan (1998), pages 165–173.

M. Á. Carreira-Perpiñán and S. Renals. A latent variable modelling approach to the acoustic-to-articulatory mapping problem. In Ohala et al. (1999), pages 2013–2016.

M. Á. Carreira-Perpiñán and S. Renals. Practical identifiability of finite mixtures of multivariate Bernoulli distributions. *Neural Computation*, 12(1):141–152, Jan. 2000.

J. Casti. Flight over Wall St. *New Scientist*, 154(2078):38–41, Apr. 19 1997.

T. Chen and R. R. Rao. Audio-visual integration in multimodal communication. *Proc. IEEE*, 86(5):837–852, May 1998.

H. Chernoff. The use of faces to represent points in k-dimensional space graphically. *J. Amer. Stat. Assoc.*, 68(342):361–368, June 1973.

D. A. Cohn. Neural network exploration using optimal experiment design. *Neural Networks*, 9(6):1071–1083, Aug. 1996.

C. H. Coker. A model of articulatory dynamics and control. *Proc. IEEE*, 64(4):452–460, 1976.

P. Comon. Independent component analysis: A new concept? *Signal Processing*, 36(3):287–314, Apr. 1994.

S. C. Constable, R. L. Parker, and C. G. Constable. Occam's inversion—a practical algorithm for generating smooth models from electromagnetic sounding data. *Geophysics*, 52(3):289–300, 1987.

D. Cook, A. Buja, and J. Cabrera. Projection pursuit indexes based on orthonormal function expansions. *Journal of Computational and Graphical Statistics*, 2(3):225–250, 1993.

M. Cooke and D. P. W. Ellis. The auditory organization of speech and other sources in listeners and computational models. *Speech Communication*, 2000. To appear.

M. Cooke, P. Green, L. Josifovski, and A. Vizinho. Robust automatic speech recognition with missing and unreliable acoustic data. *Speech Communication*, 34(3):267–285, June 2001.

D. Cornford, I. T. Nabney, and D. J. Evans. Bayesian retrieval of scatterometer wind fields. Technical Report NCRG/99/015, Neural Computing Research Group, Aston University, 1999a. Submitted to J. of Geophysical Research. Available online at `ftp://cs.aston.ac.uk/cornford/bayesret.ps.gz`.

D. Cornford, I. T. Nabney, and C. K. I. Williams. Modelling frontal discontinuities in wind fields. Technical Report NCRG/99/001, Neural Computing Research Group, Aston University, Jan. 1999b. Submitted to Nonparametric Statistics. Available online at `http://www.ncrg.aston.ac.uk/Papers/postscript/NCRG_99_001.ps.Z`.

R. Courant and D. Hilbert. *Methods of Mathematical Physics*, volume 1. Interscience Publishers, New York, 1953.

T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley Series in Telecommunications. John Wiley & Sons, New York, London, Sydney, 1991.

J. D. Cowan, G. Tesauro, and J. Alspector, editors. *Advances in Neural Information Processing Systems*, volume 6, 1994. Morgan Kaufmann, San Mateo.

T. F. Cox and M. A. A. Cox. *Multidimensional Scaling*. Chapman & Hall, London, New York, 1994.

J. J. Craig. *Introduction to Robotics. Mechanics and Control*. Series in Electrical and Computer Engineering: Control Engineering. Addison-Wesley, Reading, MA, USA, second edition, 1989.

N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge University Press, 2000.

P. Dayan. Arbitrary elastic topologies and ocular dominance. *Neural Computation*, 5(3):392–401, 1993.

P. Dayan, G. E. Hinton, R. M. Neal, and R. S. Zemel. The Helmholtz machine. *Neural Computation*, 7(5): 889–904, Sept. 1995.

M. H. DeGroot. *Probability and Statistics*. Addison-Wesley, Reading, MA, USA, 1986.

D. DeMers and G. W. Cottrell. Non-linear dimensionality reduction. In S. J. Hanson, J. D. Cowan, and C. L. Giles, editors, *Advances in Neural Information Processing Systems*, volume 5, pages 580–587. Morgan Kaufmann, San Mateo, 1993.

D. DeMers and K. Kreutz-Delgado. Learning global direct inverse kinematics. In J. Moody, S. J. Hanson, and R. P. Lippmann, editors, *Advances in Neural Information Processing Systems*, volume 4, pages 589–595. Morgan Kaufmann, San Mateo, 1992.

D. DeMers and K. Kreutz-Delgado. Canonical parameterization of excess motor degrees of freedom with self-organizing maps. *IEEE Trans. Neural Networks*, 7(1):43–55, Jan. 1996.

D. DeMers and K. Kreutz-Delgado. Learning global properties of nonredundant kinematic mappings. *Int. J. of Robotics Research*, 17(5):547–560, May 1998.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the *EM* algorithm. *Journal of the Royal Statistical Society, B*, 39(1):1–38, 1977.

L. Deng. A dynamic, feature-based approach to the interface between phonology and phonetics for speech modeling and recognition. *Speech Communication*, 24(4):299–323, July 1998.

L. Deng, G. Ramsay, and D. Sun. Production models as a structural basis for automatic speech recognition. *Speech Communication*, 22(2–3):93–111, Aug. 1997.

P. Diaconis and D. Freedman. Asymptotics of graphical projection pursuit. *Annals of Statistics*, 12(3):793–815, Sept. 1984.

K. I. Diamantaras and S.-Y. Kung. *Principal Component Neural Networks. Theory and Applications*. Wiley Series on Adaptive and Learning Systems for Signal Processing, Communications, and Control. John Wiley & Sons, New York, London, Sydney, 1996.

T. G. Dietterich. Machine learning research: Four current directions. *AI Magazine*, 18(4):97–136, winter 1997.

M. P. do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, Englewood Cliffs, N.J., 1976.

R. D. Dony and S. Haykin. Optimally adaptive transform coding. *IEEE Trans. on Image Processing*, 4(10): 1358–1370, Oct. 1995.

R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York, London, Sydney, 1973.

R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison, editors. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998.

R. Durbin and G. Mitchison. A dimension reduction framework for understanding cortical maps. *Nature*, 343 (6259):644–647, Feb. 15 1990.

R. Durbin, R. Szeliski, and A. Yuille. An analysis of the elastic net approach to the traveling salesman problem. *Neural Computation*, 1(3):348–358, Fall 1989.

R. Durbin and D. Willshaw. An analogue approach to the traveling salesman problem using an elastic net method. *Nature*, 326(6114):689–691, Apr. 16 1987.

H. W. Engl, M. Hanke, and A. Neubauer. *Regularization of Inverse Problems*. Kluwer Academic Publishers Group, Dordrecht, The Netherlands, 1996.

K. Erler and G. H. Freeman. An HMM-based speech recognizer using overlapping articulatory features. *J. Acoustic Soc. Amer.*, 100(4):2500–2513, Oct. 1996.

G. Eslava and F. H. C. Marriott. Some criteria for projection pursuit. *Statistics and Computing*, 4:13–20, 1994.

C. Y. Espy-Wilson, S. E. Boyce, M. Jackson, S. Narayanan, and A. Alwan. Acoustic modeling of American English /r/. *J. Acoustic Soc. Amer.*, 108(1):343–356, July 2000.

J. Etezadi-Amoli and R. P. McDonald. A second generation nonlinear factor analysis. *Psychometrika*, 48(3): 315–342, Sept. 1983.

D. J. Evans, D. Cornford, and I. T. Nabney. Structured neural network modelling of multi-valued functions for wind vector retrieval from satellite scatterometer measurements. *Neurocomputing*, 30(1–4):23–30, Jan. 2000.

B. S. Everitt. *An Introduction to Latent Variable Models*. Monographs on Statistics and Applied Probability. Chapman & Hall, London, New York, 1984.

B. S. Everitt and D. J. Hand. *Finite Mixture Distributions*. Monographs on Statistics and Applied Probability. Chapman & Hall, London, New York, 1981.

K. J. Falconer. *Fractal Geometry: Mathematical Foundations and Applications*. John Wiley & Sons, Chichester, 1990.

K. Fan. On a theorem of Weyl concerning the eigenvalues of linear transformations II. *Proc. Natl. Acad. Sci. USA*, 36:31–35, 1950.

G. Fant. *Acoustic Theory of Speech Production*. Mouton, The Hague, Paris, second edition, 1970.

E. Farnetani, W. J. Hardcastle, and A. Marchal. Cross-language investigation of lingual coarticulatory processes using EPG. In J.-P. Tubach and J.-J. Mariani, editors, *Proc. EUROSPEECH'89*, volume 2, pages 429–432, Paris, France, Sept. 26–28 1989.

W. Feller. *An Introduction to Probability Theory and Its Applications*, volume 2 of *Wiley Series in Probability and Mathematical Statistics*. John Wiley & Sons, New York, London, Sydney, third edition, 1971.

D. J. Field. What is the goal of sensory coding? *Neural Computation*, 6(4):559–601, July 1994.

J. L. Flanagan. *Speech Analysis, Synthesis and Perception*. Number 3 in Kommunication und Kybernetik in Einzeldarstellungen. Springer-Verlag, Berlin, second edition, 1972.

M. K. Fleming and G. W. Cottrell. Categorization of faces using unsupervised feature extraction. In *Proc. Int. J. Conf. on Neural Networks (IJCNN90)*, volume II, pages 65–70, San Diego, CA, June 17–21 1990.

P. Földiák. Adaptive network for optimal linear feature extraction. In *Proc. Int. J. Conf. on Neural Networks (IJCNN89)*, volume I, pages 401–405, Washington, DC, June 18–22 1989.

D. Fotheringhame and R. Baddeley. Nonlinear principal components analysis of neuronal data. *Biol. Cybern.*, 77(4):283–288, 1997.

I. E. Frank and J. H. Friedman. A statistical view of some chemometrics regression tools. *Technometrics*, 35 (2):109–135 (with comments: pp. 136–148), May 1993.

J. Frankel, K. Richmond, S. King, and P. Taylor. An automatic speech recognition system using neural networks and linear dynamic models to recover and model articulatory traces. In *Proc. of the International Conference on Spoken Language Processing (ICSLP'00)*, Beijing, China, Oct. 16–20 2000.

J. H. Friedman. Exploratory projection pursuit. *J. Amer. Stat. Assoc.*, 82(397):249–266, Mar. 1987.

J. H. Friedman. Multivariate adaptive regression splines. *Annals of Statistics*, 19(1):1–67 (with comments, pp. 67–141), Mar. 1991.

J. H. Friedman and N. I. Fisher. Bump hunting in high-dimensional data. *Statistics and Computing*, 9(2): 123–143 (with discussion, pp. 143–162), Apr. 1999.

J. H. Friedman and W. Stuetzle. Projection pursuit regression. *J. Amer. Stat. Assoc.*, 76(376):817–823, Dec. 1981.

J. H. Friedman, W. Stuetzle, and A. Schroeder. Projection pursuit density estimation. *J. Amer. Stat. Assoc.*, 79(387):599–608, Sept. 1984.

J. H. Friedman and J. W. Tukey. A projection pursuit algorithm for exploratory data analysis. *IEEE Trans. Computers*, C–23:881–889, 1974.

C. Fyfe and R. J. Baddeley. Finding compact and sparse distributed representations of visual images. *Network: Computation in Neural Systems*, 6(3):333–344, Aug. 1995.

J.-L. Gauvain and C.-H. Lee. Maximum a-posteriori estimation for multivariate Gaussian mixture observations of Markov chains. *IEEE Trans. Speech and Audio Process.*, 2:1291–1298, 1994.

A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian Data Analysis*. Texts in Statistical Science. Chapman & Hall, London, New York, 1995.

C. Genest and J. V. Zidek. Combining probability distributions: A critique and an annotated bibliography. *Statistical Science*, 1(1):114–135 (with discussion, pp. 135–148), Feb. 1986.

Z. Ghahramani. Solving inverse problems using an EM approach to density estimation. In M. C. Mozer, P. Smolensky, D. S. Touretzky, J. L. Elman, and A. S. Weigend, editors, *Proceedings of the 1993 Connectionist Models Summer School*, pages 316–323, 1994.

Z. Ghahramani and M. J. Beal. Variational inference for Bayesian mixture of factor analysers. In Solla et al. (2000), pages 449–455.

Z. Ghahramani and G. E. Hinton. The EM algorithm for mixtures of factor analyzers. Technical Report CRG–TR–96–1, University of Toronto, May 21 1996. Available online at `ftp://ftp.cs.toronto.edu/pub/zoubin/tr-96-1.ps.gz`.

Z. Ghahramani and M. I. Jordan. Supervised learning from incomplete data via an EM approach. In Cowan et al. (1994), pages 120–127.

W. Gilks, S. Richardson, and D. J. Spiegelhalter, editors. *Markov Chain Monte Carlo in Practice*. Chapman & Hall, London, New York, 1996.

M. Girolami, A. Cichoki, and S. Amari. A common neural network model for exploratory data analysis and independent component analysis. *IEEE Trans. Neural Networks*, 9(6):1495–1501, 1998.

M. Girolami and C. Fyfe. Stochastic ICA contrast maximization using Oja's nonlinear PCA algorithm. *Int. J. Neural Syst.*, 8(5–6):661–678, Oct./Dec. 1999.

F. Girosi, M. Jones, and T. Poggio. Regularization theory and neural networks architectures. *Neural Computation*, 7(2):219–269, Mar. 1995.

S. J. Godsill and P. J. W. Rayner. *Digital Audio Restoration: A Statistical Model-Based Approach*. Springer-Verlag, Berlin, 1998a.

S. J. Godsill and P. J. W. Rayner. Robust reconstruction and analysis of autoregressive signals in impulsive noise using the Gibbs sampler. *IEEE Trans. Speech and Audio Process.*, 6(4):352–372, July 1998b.

B. Gold and N. Morgan. *Speech and Audio Signal Processing: Processing and Perception of Speech and Music*. John Wiley & Sons, New York, London, Sydney, 2000.

D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.

G. H. Golub and C. F. van Loan. *Matrix Computations*. Johns Hopkins Press, Baltimore, third edition, 1996.

G. J. Goodhill and T. J. Sejnowski. A unifying objective function for topographic mappings. *Neural Computation*, 9(6):1291–1303, Aug. 1997.

R. A. Gopinath, B. Ramabhadran, and S. Dharanipragada. Factor analysis invariant to linear transformations of data. In *Proc. of the International Conference on Spoken Language Processing (ICSLP'98)*, Sydney, Australia, Nov. 30 – Dec. 4 1998.

W. P. Gouveia and J. A. Scales. Resolution of seismic waveform inversion: Bayes versus Occam. *Inverse Problems*, 13(2):323–349, Apr. 1997.

W. P. Gouveia and J. A. Scales. Bayesian seismic waveform inversion: Parameter estimation and uncertainty analysis. *J. of Geophysical Research*, 130(B2):2759–2779, 1998.

I. S. Gradshteyn and I. M. Ryzhik. *Table of Integrals, Series, and Products*. Academic Press, San Diego, fifth edition, 1994. Corrected and enlarged edition, edited by Alan Jeffrey.

R. M. Gray. Vector quantization. *IEEE ASSP Magazine*, pages 4–29, Apr. 1984.

R. M. Gray and D. L. Neuhoff. Quantization. *IEEE Trans. Inf. Theory*, 44(6):2325–2383, Oct. 1998.

M. Gyllenberg, T. Koski, E. Reilink, and M. Verlaan. Non-uniqueness in probabilistic numerical identification of bacteria. *J. Appl. Prob.*, 31:542–548, 1994.

P. Hall. On polynomial-based projection indices for exploratory projection pursuit. *Annals of Statistics*, 17 (2):589–605, June 1989.

W. J. Hardcastle, F. E. Gibbon, and W. Jones. Visual display of tongue-palate contact: Electropalatography in the assessment and remediation of speech disorders. *Brit. J. of Disorders of Communication*, 26:41–74, 1991a.

W. J. Hardcastle, F. E. Gibbon, and K. Nicolaidis. EPG data reduction methods and their implications for studies of lingual coarticulation. *J. of Phonetics*, 19:251–266, 1991b.

W. J. Hardcastle and N. Hewlett, editors. *Coarticulation: Theory, Data, and Techniques*. Cambridge Studies in Speech Science and Communication. Cambridge University Press, Cambridge, U.K., 1999.

W. J. Hardcastle, W. Jones, C. Knight, A. Trudgeon, and G. Calder. New developments in electropalatography: A state-of-the-art report. *J. Clinical Linguistics and Phonetics*, 3:1–38, 1989.

H. H. Harman. *Modern Factor Analysis*. University of Chicago Press, Chicago, second edition, 1967.

A. C. Harvey. *Forecasting, Structural Time Series Models and the Kalman Filter*. Cambridge University Press, 1991.

T. J. Hastie and W. Stuetzle. Principal curves. *J. Amer. Stat. Assoc.*, 84(406):502–516, June 1989.

T. J. Hastie and R. J. Tibshirani. *Generalized Additive Models*. Number 43 in Monographs on Statistics and Applied Probability. Chapman & Hall, London, New York, 1990.

G. T. Herman. *Image Reconstruction from Projections. The Fundamentals of Computer Tomography*. Academic Press, New York, 1980.

H. Hermansky. Perceptual linear predictive (PLP) analysis of speech. *J. Acoustic Soc. Amer.*, 87(4):1738–1752, Apr. 1990.

H. Hermansky and N. Morgan. RASTA processing of speech. *IEEE Trans. Speech and Audio Process.*, 2(4): 578–589, Oct. 1994.

J. A. Hertz, A. S. Krogh, and R. G. Palmer. *Introduction to the Theory of Neural Computation*. Number 1 in Santa Fe Institute Studies in the Sciences of Complexity Lecture Notes. Addison-Wesley, Reading, MA, USA, 1991.

G. E. Hinton. Products of experts. In D. Wilshaw, editor, *Proc. of the Ninth Int. Conf. on Artificial Neural Networks (ICANN99)*, pages 1–6, Edinburgh, UK, Sept. 7–10 1999. The Institution of Electrical Engineers.

G. E. Hinton, P. Dayan, and M. Revow. Modeling the manifolds of images of handwritten digits. *IEEE Trans. Neural Networks*, 8(1):65–74, Jan. 1997.

T. Holst, P. Warren, and F. Nolan. Categorising [s], [ʃ] and intermediate electropalographic patterns: Neural networks and other approaches. *European Journal of Disorders of Communication*, 30(2):161–174, 1995.

H. Hotelling. Analysis of a complex of statistical variables into principal components. *J. of Educational Psychology*, 24:417–441 and 498–520, 1933.

P. J. Huber. *Robust Statistics*. Wiley Series in Probability and Mathematical Statistics. John Wiley & Sons, New York, London, Sydney, 1981.

P. J. Huber. Projection pursuit. *Annals of Statistics*, 13(2):435–475 (with comments, pp. 475–525), June 1985.

D. Husmeier. *Neural Networks for Conditional Probability Estimation*. Perspectives in Neural Computing. Springer-Verlag, Berlin, 1999.

J.-N. Hwang, S.-R. Lay, M. Maechler, R. D. Martin, and J. Schimert. Regression modeling in back-propagation and projection pursuit learning. *IEEE Trans. Neural Networks*, 5(3):342–353, May 1994.

A. Hyvärinen. New approximations of differential entropy for independent component analysis and projection pursuit. In Jordan et al. (1998), pages 273–279.

A. Hyvärinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Trans. Neural Networks*, 10(3):626–634, Oct. 1999a.

A. Hyvärinen. Survey on independent component analysis. *Neural Computing Surveys*, 2:94–128, 1999b.

A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. Wiley Series on Adaptive and Learning Systems for Signal Processing, Communications, and Control. John Wiley & Sons, New York, London, Sydney, 2001.

A. Hyvärinen and E. Oja. Independent component analysis: Algorithms and applications. *Neural Networks*, 13(4–5):411–430, June 2000.

N. Intrator and L. N. Cooper. Objective function formulation of the BCM theory of visual cortical plasticity: Statistical connections, stability conditions. *Neural Networks*, 5(1):3–17, 1992.

E. Isaacson and H. B. Keller. *Analysis of Numerical Methods*. John Wiley & Sons, New York, London, Sydney, 1966.

M. Isard and A. Blake. CONDENSATION — conditional density propagation for visual tracking. *Int. J. Computer Vision*, 29(1):5–28, 1998.

J. E. Jackson. *A User's Guide to Principal Components*. Wiley Series in Probability and Mathematical Statistics. John Wiley & Sons, New York, London, Sydney, 1991.

R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991.

M. Jamshidian and P. M. Bentler. A quasi-Newton method for minimum trace factor analysis. *J. of Statistical Computation and Simulation*, 62(1–2):73–89, 1998.

N. Japkowicz, S. J. Hanson, and M. A. Gluck. Nonlinear autoassociation is not equivalent to PCA. *Neural Computation*, 12(3):531–545, Mar. 2000.

E. T. Jaynes. Prior probabilities. *IEEE Trans. Systems, Science, and Cybernetics*, SSC-4(3):227–241, 1968.

F. V. Jensen. *An Introduction to Bayesian Networks*. UCL Press, London, 1996.

I. T. Jolliffe. *Principal Component Analysis*. Springer Series in Statistics. Springer-Verlag, Berlin, 1986.

M. C. Jones. *The Projection Pursuit Algorithm for Exploratory Data Analysis*. PhD thesis, University of Bath, 1983.

M. C. Jones and R. Sibson. What is projection pursuit? *Journal of the Royal Statistical Society, A*, 150(1): 1–18 (with comments, pp. 19–36), 1987.

W. Jones and W. J. Hardcastle. New developments in EPG3 software. *European Journal of Disorders of Communication*, 30(2):183–192, 1995.

M. I. Jordan. Motor learning and the degrees of freedom problem. In M. Jeannerod, editor, *Attention and Performance XIII*, pages 796–836. Lawrence Erlbaum Associates, Hillsdale, New Jersey and London, 1990.

M. I. Jordan, editor. *Learning in Graphical Models*, Adaptive Computation and Machine Learning series, 1998. MIT Press. Proceedings of the NATO Advanced Study Institute on Learning in Graphical Models, held in Erice, Italy, September 27 – October 7, 1996.

M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233, Nov. 1999.

M. I. Jordan and R. A. Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6(2):181–214, Mar. 1994.

M. I. Jordan, M. J. Kearns, and S. A. Solla, editors. *Advances in Neural Information Processing Systems*, volume 10, 1998. MIT Press, Cambridge, MA.

M. I. Jordan and D. E. Rumelhart. Forward models: Supervised learning with a distal teacher. *Cognitive Science*, 16(3):307–354, July–Sept. 1992.

K. G. Jöreskog. Some contributions to maximum likelihood factor analysis. *Psychometrika*, 32(4):443–482, Dec. 1967.

K. G. Jöreskog. A general approach to confirmatory maximum likelihood factor analysis. *Psychometrika*, 34 (2):183–202, June 1969.

H. F. Kaiser. The varimax criterion for analytic rotation in factor analysis. *Psychometrika*, 23(3):187–200, Sept. 1958.

N. Kambhatla and T. K. Leen. Dimension reduction by local principal component analysis. *Neural Computation*, 9(7):1493–1516, Oct. 1997.

G. K. Kanji. 100 *Statistical Tests*. Sage Publications, London, 1993.

J. N. Kapur. *Maximum-Entropy Models in Science and Engineering*. John Wiley & Sons, New York, London, Sydney, 1989.

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

R. E. Kass and L. Wasserman. The selection of prior distributions by formal rules. *J. Amer. Stat. Assoc.*, 91 (435):1343–1370, Sept. 1996.

M. S. Kearns, S. A. Solla, and D. A. Cohn, editors. *Advances in Neural Information Processing Systems*, volume 11, 1999. MIT Press, Cambridge, MA.

B. Kégl, A. Krzyzak, T. Linder, and K. Zeger. Learning and design of principal curves. *IEEE Trans. on Pattern Anal. and Machine Intel.*, 22(3):281–297, Mar. 2000.

M. G. Kendall and A. Stuart. *The Advanced Theory of Statistics Vol. 1: Distribution Theory*. Charles Griffin & Company Ltd., London, fourth edition, 1977.

W. M. Kier and K. K. Smith. Tongues, tentacles and trunks: The biomechanics of movement in muscular-hydrostats. *Zoological Journal of the Linnean Society*, 83:307–324, 1985.

S. King and A. Wrench. Dynamical system modelling of articulator movement. In Ohala et al. (1999), pages 2259–2262.

B. E. D. Kingsbury, N. Morgan, and S. Greenberg. Robust speech recognition using the modulation spectrogram. *Speech Communication*, 25(1–3):117–132, Aug. 1998.

T. K. Kohonen. *Self-Organizing Maps*. Number 30 in Springer Series in Information Sciences. Springer-Verlag, Berlin, 1995.

A. C. Kokaram, R. D. Morris, W. J. Fitzgerald, and P. J. W. Rayner. Interpolation of missing data in image sequences. *IEEE Trans. on Image Processing*, 4(11):1509–1519, Nov. 1995.

J. F. Kolen and J. B. Pollack. Back propagation is sensitive to initial conditions. *Complex Systems*, 4(3): 269–280, 1990.

A. C. Konstantellos. Unimodality conditions for Gaussian sums. *IEEE Trans. Automat. Contr.*, AC–25(4): 838–839, Aug. 1980.

M. A. Kramer. Nonlinear principal component analysis using autoassociative neural networks. *Journal of the American Institute of Chemical Engineers*, 37(2):233–243, Feb. 1991.

J. B. Kruskal and M. Wish. *Multidimensional Scaling*. Number 07–011 in Sage University Paper Series on Quantitative Applications in the Social Sciences. Sage Publications, Beverly Hills, 1978.

W. J. Krzanowski. *Principles of Multivariate Analysis: A User's Perspective*. Number 3 in Oxford Statistical Science Series. Oxford University Press, New York, Oxford, 1988.

S. Y. Kung, K. I. Diamantaras, and J. S. Taur. Adaptive principal component extraction (APEX) and applications. *IEEE Trans. Signal Processing*, 42(5):1202–1217, May 1994.

O. M. Kvalheim. The latent variable. *Chemometrics and Intelligent Laboratory Systems*, 14:1–3, 1992.

P. Ladefoged. Articulatory parameters. *Language and Speech*, 23(1):25–30, Jan.–Mar. 1980.

P. Ladefoged. *A Course in Phonetics*. Harcourt College Publishers, Fort Worth, fourth edition, 2000.

N. Laird. Nonparametric maximum likelihood estimation of a mixing distribution. *J. Amer. Stat. Assoc.*, 73 (364):805–811, Dec. 1978.

J. N. Larar, J. Schroeter, and M. M. Sondhi. Vector quantisation of the articulatory space. *IEEE Trans. Acoust., Speech, and Signal Process.*, ASSP-36(12):1812–1818, Dec. 1988.

F. Lavagetto. Time-delay neural networks for estimating lip movements from speech analysis: A useful tool in audio-video synchronization. *IEEE Trans. Circuits and Systems for video technology*, 7(5):786–800, Oct. 1997.

E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys, editors. *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*. Wiley Series in Discrete Mathematics and Optimization. John Wiley & Sons, Chichester, England, 1985.

D. N. Lawley. A modified method of estimation in factor analysis and some large sample results. *Nord. Psykol. Monogr. Ser.*, 3:35–42, 1953.

P. F. Lazarsfeld and N. W. Henry. *Latent Structure Analysis*. Houghton-Mifflin, Boston, 1968.

M. LeBlanc and R. Tibshirani. Adaptive principal surfaces. *J. Amer. Stat. Assoc.*, 89(425):53–64, Mar. 1994.

D. D. Lee and H. Sompolinsky. Learning a continuous hidden variable model for binary data. In Kearns et al. (1999), pages 515–521.

T.-W. Lee, M. Girolami, and T. J. Sejnowski. Independent component analysis using an extended infomax algorithm for mixed subgaussian and supergaussian sources. *Neural Computation*, 11(2):417–441, Feb. 1999.

C. J. Leggeter and P. C. Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models. *Computer Speech and Language*, 9(2):171–185, Apr. 1995.

S. E. Levinson and C. E. Schmidt. Adaptive computation of articulatory parameters from the speech signal. *J. Acoustic Soc. Amer.*, 74(4):1145–1154, Oct. 1983.

M. S. Lewicki and T. J. Sejnowski. Learning overcomplete representations. *Neural Computation*, 12(2):337–365, Feb. 2000.

A. M. Liberman, F. S. Cooper, D. P. Shankweiler, and M. Studdert-Kennedy. Perception of the speech code. *Psychological Review*, 74(6):431–461, 1967.

A. M. Liberman and I. G. Mattingly. The motor theory of speech perception revised. *Cognition*, 21:1–36, 1985.

B. G. Lindsay. The geometry of mixture likelihoods: A general theory. *Annals of Statistics*, 11(1):86–94, Mar. 1983.

R. Linsker. An application of the principle of maximum information preservation to linear systems. In D. S. Touretzky, editor, *Advances in Neural Information Processing Systems*, volume 1, pages 186–194. Morgan Kaufmann, San Mateo, 1989.

R. J. A. Little. Regression with missing X's: A review. *J. Amer. Stat. Assoc.*, 87(420):1227–1237, Dec. 1992.

R. J. A. Little and D. B. Rubin. *Statistical Analysis with Missing Data*. Wiley Series in Probability and Mathematical Statistics. John Wiley & Sons, New York, London, Sydney, 1987.

S. P. Luttrell. A Bayesian analysis of self-organizing maps. *Neural Computation*, 6(5):767–794, Sept. 1994.

D. J. C. MacKay. Bayesian interpolation. *Neural Computation*, 4(3):415–447, May 1992a.

D. J. C. MacKay. A practical Bayesian framework for backpropagation networks. *Neural Computation*, 4(3): 448–472, May 1992b.

D. J. C. MacKay. Bayesian neural networks and density networks. *Nuclear Instruments and Methods in Physics Research A*, 354(1):73–80, Jan. 1995a.

D. J. C. MacKay. Probable networks and plausible predictions — a review of practical Bayesian methods for supervised neural networks. *Network: Computation in Neural Systems*, 6(3):469–505, 1995b.

D. J. C. MacKay. Maximum likelihood and covariant algorithms for independent component analysis. Draft 3.7, Cavendish Laboratory, University of Cambridge, Dec. 19 1996. Available online at `http://wol.ra.phy.cam.ac.uk/mackay/abstracts/ica.html`.

D. J. C. MacKay. Comparison of approximate methods for handling hyperparameters. *Neural Computation*, 11(5):1035–1068, July 1999.

S. Maeda. A digital simulation method of the vocal tract system. *Speech Communication*, 1(3–4):199–229, 1982.

S. Makeig, T.-P. Jung, A. J. Bell, D. Ghahremani, and T. J. Sejnowski. Blind separation of auditory event-related brain responses into independent components. *Proc. Natl. Acad. Sci. USA*, 94:10979–10984, Sept. 1997.

E. C. Malthouse. Limitations of nonlinear PCA as performed with generic neural networks. *IEEE Trans. Neural Networks*, 9(1):165–173, Jan. 1998.

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

A. Marchal and W. J. Hardcastle. ACCOR: Instrumentation and database for the cross-language study of coarticulation. *Language and Speech*, 36(2, 3):137–153, 1993.

K. V. Mardia, J. T. Kent, and J. M. Bibby. *Multivariate Analysis*. Probability and Mathematical Statistics Series. Academic Press, New York, 1979.

A. D. Marrs and A. R. Webb. Exploratory data analysis using radial basis function latent variable models. In Kearns et al. (1999), pages 529–535.

T. M. Martinetz and K. J. Schulten. Topology representing networks. *Neural Networks*, 7(3):507–522, 1994.

G. P. McCabe. Principal variables. *Technometrics*, 26(2):137–144, 1984.

R. P. McDonald. *Factor Analysis and Related Methods*. Lawrence Erlbaum Associates, Hillsdale, New Jersey and London, 1985.

R. S. McGowan. Recovering articulatory movement from formant frequency trajectories using task dynamics and a genetic algorithm: Preliminary model tests. *Speech Communication*, 14(1):19–48, Feb. 1994.

R. S. McGowan and A. Faber. Introduction to papers on speech recognition and perception from an articulatory point of view. *J. Acoustic Soc. Amer.*, 99(3):1680–1682, Mar. 1996.

G. J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. Wiley Series in Probability and Mathematical Statistics. John Wiley & Sons, 1997.

G. J. McLachlan and D. Peel. *Finite Mixture Models*. Wiley Series in Probability and Mathematical Statistics. John Wiley & Sons, 2000.

X.-L. Meng and D. van Dyk. The EM algorithm — an old folk-song sung to a fast new tune. *Journal of the Royal Statistical Society, B*, 59(3):511–540 (with discussion, pp. 541–567), 1997.

P. Mermelstein. Determination of vocal-tract shape from measured formant frequencies. *J. Acoustic Soc. Amer.*, 41(5):1283–1294, 1967.

P. Mermelstein. Articulatory model for the study of speech production. *J. Acoustic Soc. Amer.*, 53(4):1070–1082, 1973.

L. Mirsky. *An Introduction to Linear Algebra*. Clarendon Press, Oxford, 1955. Reprinted in 1982 by Dover Publications.

B. Moghaddam and A. Pentland. Probabilistic visual learning for object representation. *IEEE Trans. on Pattern Anal. and Machine Intel.*, 19(7):696–710, July 1997.

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

D. F. Morrison. *Multivariate Statistical Methods*. McGraw-Hill, New York, third edition, 1990.

K. Mosegaard and A. Tarantola. Monte-Carlo sampling of solutions to inverse problems. *J. of Geophysical Research—Solid Earth*, 100(B7):12431–12447, 1995.

É. Moulines, J.-F. Cardoso, and E. Gassiat. Maximum likelihood for blind separation and deconvolution of noisy signals using mixture models. In *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Sig. Proc. (ICASSP'97)*, volume 5, pages 3617–3620, Munich, Germany, Apr. 21–24 1997.

J. R. Movellan, P. Mineiro, and R. J. Williams. Modeling path distributions using partially observable diffusion networks: A Monte-Carlo approach. Technical Report 99.01, Department of Cognitive Science, University of California, San Diego, June 1999. Available online at `http://hci.ucsd.edu/cogsci/tech_reports/faculty_pubs/99_01.ps`.

F. Mulier and V. Cherkassky. Self-organization as an iterative kernel smoothing process. *Neural Computation*, 7(6):1165–1177, Nov. 1995.

I. T. Nabney, D. Cornford, and C. K. I. Williams. Bayesian inference for wind field retrieval. *Neurocomputing*, 30(1–4):3–11, Jan. 2000.

J.-P. Nadal and N. Parga. Non-linear neurons in the low-noise limit: A factorial code maximizes information transfer. *Network: Computation in Neural Systems*, 5(4):565–581, Nov. 1994.

R. M. Neal. Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG–TR–93–1, Dept. of Computer Science, University of Toronto, Sept. 1993. Available online at `ftp://ftp.cs.toronto.edu/pub/radford/review.ps.Z`.

R. M. Neal. *Bayesian Learning for Neural Networks*. Springer Series in Statistics. Springer-Verlag, Berlin, 1996.

R. M. Neal and P. Dayan. Factor analysis using delta-rule wake-sleep learning. *Neural Computation*, 9(8): 1781–1803, Nov. 1997.

R. M. Neal and G. E. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. In Jordan (1998), pages 355–368. Proceedings of the NATO Advanced Study Institute on Learning in Graphical Models, held in Erice, Italy, September 27 – October 7, 1996.

W. L. Nelson. Physical principles for economies of skilled movements. *Biol. Cybern.*, 46(2):135–147, 1983.

N. Nguyen. EPG bidimensional data reduction. *European Journal of Disorders of Communication*, 30:175–182, 1995.

N. Nguyen, P. Hoole, and A. Marchal. Regenerating the spectral shape of [s] and [ʃ] from a limited set of articulatory parameters. *J. Acoustic Soc. Amer.*, 96(1):33–39, July 1994.

N. Nguyen, A. Marchal, and A. Content. Modeling tongue-palate contact patterns in the production of speech. *J. of Phonetics*, 24(1):77–97, Jan. 1996.

K. Nicolaidis and W. J. Hardcastle. Articulatory-acoustic analysis of selected English sentences from the EUR-ACCOR corpus. Technical report, SPHERE (Human capital and mobility program), 1994.

K. Nicolaidis, W. J. Hardcastle, A. Marchal, and N. Nguyen-Trong. Comparing phonetic, articulatory, acoustic and aerodynamic signal representations. In M. Cooke, S. Beet, and M. Crawford, editors, *Visual Representations of Speech Signals*, pages 55–82. John Wiley & Sons, 1993.

M. A. L. Nicolelis. Actions from thoughts. *Nature*, 409(6818):403–407, Jan. 18 2001.

M. Niranjan, editor. *Proc. of the 1998 IEEE Signal Processing Society Workshop on Neural Networks for Signal Processing (NNSP98)*, Cambridge, UK, Aug. 31 – Sept. 2 1998.

D. A. Nix and J. E. Hogden. Maximum-likelihood continuity mapping (MALCOM): An alternative to HMMs. In Kearns et al. (1999), pages 744–750.

J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research. Springer-Verlag, 1999.

J. J. Ohala, Y. Hasegawa, M. Ohala, D. Granville, and A. C. Bailey, editors. *Proc. of the 14th International Congress of Phonetic Sciences (ICPhS'99)*, San Francisco, USA, Aug. 1–7 1999.

E. Oja. Principal components, minor components, and linear neural networks. *Neural Networks*, 5(6):927–935, Nov.–Dec. 1992.

B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, June 13 1996.

B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Res.*, 37(23):3311–3325, Dec. 1997.

M. W. Oram, P. Földiák, D. I. Perret, and F. Sengpiel. The 'ideal homunculus': Decoding neural population signals. *Trends Neurosci.*, 21(6):259–265, June 1998.

D. Ormoneit and V. Tresp. Penalized likelihood and Bayesian estimation for improving Gaussian mixture probability density estimates. *IEEE Trans. Neural Networks*, 9(4):639–650, July 1998.

M. Ostendorf, V. V. Digalakis, and O. A. Kimball. From HMM's to segment models: A unified view of stochastic modeling for speech recognition. *IEEE Trans. Speech and Audio Process.*, 4(5):360–378, Sept. 1996.

G. Papcun, J. Hochberg, T. R. Thomas, F. Laroche, J. Zacks, and S. Levy. Inferring articulation and recognizing gestures from acoustics with a neural network trained on x-ray microbeam data. *J. Acoustic Soc. Amer.*, 92(2):688–700, Aug. 1992.

J. Park and I. W. Sandberg. Approximation and radial-basis-function networks. *Neural Computation*, 5(2): 305–316, Mar. 1993.

R. L. Parker. *Geophysical Inverse Theory*. Princeton University Press, Princeton, 1994.

J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, 1988.

B. A. Pearlmutter. Gradient calculation for dynamic recurrent neural networks: A survey. *IEEE Trans. Neural Networks*, 6(5):1212–1228, 1995.

B. A. Pearlmutter and L. C. Parra. A context-sensitive generalization of ICA. In *International Conference on Neural Information Processing (ICONIP–96), Hong Kong*, pages 151–157, Sept. 1996.

K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(6): 559–572, 1901.

D. Peel and G. J. McLachlan. Robust mixture modelling using the t distribution. *Statistics and Computing*, 10(4):339–348, Oct. 2000.

H.-O. Peitgen, H. Jürgens, and D. Saupe. *Chaos and Fractals: New Frontiers of Science*. Springer-Verlag, New York, 1992.

J. Picone, S. Pike, R. Reagan, T. Kamm, J. Bridle, L. Deng, J. Ma, H. Richards, and M. Schuster. Initial evaluation of hidden dynamic models on conversational speech. In *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Sig. Proc. (ICASSP'99)*, volume 1, pages 109–112, Phoenix, Arizona, USA, May 15–19 1999.

A. Pisani. A nonparametric and scale-independent method for cluster-analysis. 1. The univariate case. *Monthly Notices of the Royal Astronomical Society*, 265(3):706–726, Dec. 1993.

C. Posse. An effective two-dimensional projection pursuit algorithm. *Communications in Statistics — Simulation and Computation*, 19(4):1143–1164, 1990.

C. Posse. Tools for two-dimensional exploratory projection pursuit. *Journal of Computational and Graphical Statistics*, 4:83–100, 1995.

S. Pratt, A. T. Heintzelman, and D. S. Ensrud. The efficacy of using the IBM Speech Viewer vowel accuracy module to treat young children with hearing impairment. *Journal of Speech and Hearing Research*, 29: 99–105, 1993.

F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction*. Monographs in Computer Science. Springer-Verlag, New York, 1985.

W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, Cambridge, U.K., second edition, 1992.

L. Rabiner and B.-H. Juang. *Fundamentals of Speech Recognition*. Signal Processing Series. Prentice-Hall, Englewood Cliffs, N.J., 1993.

M. G. Rahim, C. C. Goodyear, W. B. Kleijn, J. Schroeter, and M. M. Sondhi. On the use of neural networks in articulatory speech synthesis. *J. Acoustic Soc. Amer.*, 93(2):1109–1121, Feb. 1993.

R. A. Redner and H. F. Walker. Mixture densities, maximum likelihood and the EM algorithm. *SIAM Review*, 26(2):195–239, Apr. 1984.

K. Reinhard and M. Niranjan. Parametric subspace modeling of speech transitions. *Speech Communication*, 27(1):19–42, Feb. 1999.

M. Revow, C. K. I. Williams, and G. Hinton. Using generative models for handwritten digit recognition. *IEEE Trans. on Pattern Anal. and Machine Intel.*, 18(6):592–606, June 1996.

H. B. Richards and J. S. Bridle. The HDM: a segmental hidden dynamic model of coarticulation. In *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Sig. Proc. (ICASSP'99)*, volume I, pages 357–360, Phoenix, Arizona, USA, May 15–19 1999.

S. Richardson and P. J. Green. On Bayesian analysis of mixtures with an unknown number of components. *Journal of the Royal Statistical Society, B*, 59(4):731–758, 1997.

B. D. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, Cambridge, U.K., 1996.

S. J. Roberts. Parametric and non-parametric unsupervised cluster analysis. *Pattern Recognition*, 30(2): 261–272, Feb. 1997.

S. J. Roberts, D. Husmeier, I. Rezek, and W. Penny. Bayesian approaches to Gaussian mixture modeling. *IEEE Trans. on Pattern Anal. and Machine Intel.*, 20(11):1133–1142, 1998.

W. J. J. Roberts and Y. Ephraim. Hidden Markov modeling of speech using Toeplitz covariance matrices. *Speech Communication*, 31(1):1–14, May 2000.

A. J. Robinson. An application of recurrent nets to phone probability estimation. *IEEE Trans. Neural Networks*, 5(2):298–305, Mar. 1994.

T. Rögnvaldsson. On Langevin updating in multilayer perceptrons. *Neural Computation*, 6(5):916–926, Sept. 1994.

R. Rohwer and J. C. van der Rest. Minimum description length, regularization, and multimodal data. *Neural Computation*, 8(3):595–609, Apr. 1996.

E. T. Rolls and A. Treves. *Neural Networks and Brain Function*. Oxford University Press, 1998.

K. Rose. Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. *Proc. IEEE*, 86(11):2210–2239, Nov. 1998.

R. C. Rose, J. Schroeter, and M. M. Sondhi. The potential role of speech production models in automatic speech recognition. *J. Acoustic Soc. Amer.*, 99(3):1699–1709 (with comments, pp. 1710–1717), Mar. 1996.

E. Z. Rothkopf. A measure of stimulus similarity and errors in some paired-associate learning tasks. *J. of Experimental Psychology*, 53(2):94–101, 1957.

B. Rotman and G. T. Kneebone. *The Theory of Sets & Transfinite Numbers*. Oldbourne, London, 1966.

S. Roweis. EM algorithms for PCA and SPCA. In Jordan et al. (1998), pages 626–632.

S. Roweis. Constrained hidden Markov models. In Solla et al. (2000), pages 782–788.

S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290 (5500):2323–2326, Dec. 22 2000.

A. E. Roy. *Orbital Motion*. Adam Hilger Ltd., Bristol, 1978.

D. B. Rubin. *Multiple Imputation for Nonresponse in Surveys*. Wiley Series in Probability and Mathematical Statistics. John Wiley & Sons, 1987.

D. B. Rubin and D. T. Thayer. EM algorithms for ML factor analysis. *Psychometrika*, 47(1):69–76, Mar. 1982.

D. B. Rubin and D. T. Thayer. More on EM for ML factor analysis. *Psychometrika*, 48(2):253–257, June 1983.

P. Rubin, T. Baer, and P. Mermelstein. An articulatory synthesizer for perceptual research. *J. Acoustic Soc. Amer.*, 70(2):321–328, Aug. 1981.

E. Saltzman and J. A. Kelso. Skilled actions: a task-dynamic approach. *Psychological Review*, 94(1):84–106, Jan. 1987.

J. W. Sammon, Jr. A nonlinear mapping for data structure analysis. *IEEE Trans. Computers*, C–18(5): 401–409, May 1969.

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

T. D. Sanger. Probability density estimation for the interpretation of neural population codes. *J. Neurophysiol.*, 76(4):2790–2793, Oct. 1996.

L. K. Saul and M. G. Rahim. Markov processes on curves. *Machine Learning*, 41(3):345–363, Dec. 2000a.

L. K. Saul and M. G. Rahim. Maximum likelihood and minimum classification error factor analysis for automatic speech recognition. *IEEE Trans. Speech and Audio Process.*, 8(2):115–125, Mar. 2000b.

E. Saund. Dimensionality-reduction using connectionist networks. *IEEE Trans. on Pattern Anal. and Machine Intel.*, 11(3):304–314, Mar. 1989.

J. A. Scales and M. L. Smith. *Introductory Geophysical Inverse Theory*. Samizdat Press, 1998. Freely available in draft form from `http://samizdat.mines.edu/inverse_theory/`.

F. Scarselli and A. C. Tsoi. Universal approximation using feedforward neural networks: A survey of some existing methods, and some new results. *Neural Networks*, 11(1):15–37, Jan. 1998.

J. L. Schafer. *Analysis of Incomplete Multivariate Data*. Number 72 in Monographs on Statistics and Applied Probability. Chapman & Hall, London, New York, 1997.

B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors. *Advances in Kernel Methods. Support Vector Learning*. MIT Press, 1999a.

B. Schölkopf, S. Mika, C. J. C. Burges, P. Knirsch, K.-R. Müller, G. Rätsch, and A. Smola. Input space vs. feature space in kernel-based methods. *IEEE Trans. Neural Networks*, 10(5):1000–1017, Sept. 1999b.

B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, July 1998.

M. R. Schroeder. Determination of the geometry of the human vocal tract by acoustic measurements. *J. Acoustic Soc. Amer.*, 41(4):1002–1010, 1967.

J. Schroeter and M. M. Sondhi. Dynamic programming search of articulatory codebooks. In *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Sig. Proc. (ICASSP'89)*, volume 1, pages 588–591, Glasgow, UK, May 23–26 1989.

J. Schroeter and M. M. Sondhi. Techniques for estimating vocal-tract shapes from the speech signal. *IEEE Trans. Speech and Audio Process.*, 2(1):133–150, Jan. 1994.

M. Schuster. *On Supervised Learning from Sequential Data with Applications for Speech Recognition*. PhD thesis, Graduate School of Information Science, Nara Institute of Science and Technology, 1999.

D. W. Scott. *Multivariate Density Estimation. Theory, Practice, and Visualization*. Wiley Series in Probability and Mathematical Statistics. John Wiley & Sons, New York, London, Sydney, 1992.

D. W. Scott and J. R. Thompson. Probability density estimation in higher dimensions. In J. E. Gentle, editor, *Computer Science and Statistics: Proceedings of the Fifteenth Symposium on the Interface*, pages 173–179, Amsterdam, New York, Oxford, 1983. North Holland-Elsevier Science Publishers.

R. N. Shepard. Analysis of proximities as a technique for the study of information processing in man. *Human Factors*, 5:33–48, 1963.

K. Shirai and T. Kobayashi. Estimating articulatory motion from speech wave. *Speech Communication*, 5(2): 159–170, June 1986.

M. F. Shlesinger, G. M. Zaslavsky, and U. Frisch, editors. *Lévy Flights and Related Topics in Physics*. Number 450 in Lecture Notes in Physics. Springer-Verlag, Berlin, 1995. Proceedings of the International Workshop held at Nice, France, 27–30 June 1994.

B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Number 26 in Monographs on Statistics and Applied Probability. Chapman & Hall, London, New York, 1986.

L. Sirovich and M. Kirby. Low-dimensional procedure for the identification of human faces. *J. Opt. Soc. Amer. A*, 4(3):519–524, Mar. 1987.

D. S. Sivia. *Data Analysis. A Bayesian Tutorial*. Oxford University Press, New York, Oxford, 1996.

R. Snieder and J. Trampert. *Inverse Problems in Geophysics*. Samizdat Press, 1999. Freely available from `http://samizdat.mines.edu/snieder_trampert/`.

S. A. Solla, T. K. Leen, and K.-R. Müller, editors. *Advances in Neural Information Processing Systems*, volume 12, 2000. MIT Press, Cambridge, MA.

V. N. Sorokin. Determination of vocal-tract shape for vowels. *Speech Communication*, 11(1):71–85, Mar. 1992.

V. N. Sorokin, A. S. Leonov, and A. V. Trushkin. Estimation of stability and accuracy of inverse problem solution for the vocal tract. *Speech Communication*, 30(1):55–74, Jan. 2000.

C. Spearman. General intelligence, objectively determined and measured. *Am. J. Psychol.*, 15:201–293, 1904.

D. F. Specht. A general regression neural network. *IEEE Trans. Neural Networks*, 2(6):568–576, Nov. 1991.

M. Spivak. *Calculus on Manifolds: A Modern Approach to Classical Theorems of Advanced Calculus*. Addison-Wesley, Reading, MA, USA, 1965.

M. Spivak. *Calculus*. Addison-Wesley, Reading, MA, USA, 1967.

M. Stone. Toward a model of three-dimensional tongue movement. *J. of Phonetics*, 19:309–320, 1991.

N. V. Swindale. The development of topography in the visual cortex: A review of models. *Network: Computation in Neural Systems*, 7(2):161–247, May 1996.

A. Tarantola. *Inverse Problem Theory: Methods for Data Fitting and Model Parameter Estimation*. Elsevier Science Publishers B.V., Amsterdam, The Netherlands, 1987.

J. B. Tenenbaum. Mapping a manifold of perceptual observations. In Jordan et al. (1998), pages 682–688.

J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, Dec. 22 2000.

G. Tesauro, D. S. Touretzky, and T. K. Leen, editors. *Advances in Neural Information Processing Systems*, volume 7, 1995. MIT Press, Cambridge, MA.

R. J. Tibshirani. Principal curves revisited. *Statistics and Computing*, 2:183–190, 1992.

A. N. Tikhonov and V. Y. Arsenin. *Solutions of Ill-Posed Problems*. Scripta Series in Mathematics. John Wiley & Sons, New York, London, Sydney, 1977. Translation editor: Fritz John.

M. E. Tipping and C. M. Bishop. Mixtures of probabilistic principal component analyzers. *Neural Computation*, 11(2):443–482, Feb. 1999a.

M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, B*, 61(3):611–622, 1999b.

D. M. Titterington, A. F. M. Smith, and U. E. Makov. *Statistical Analysis of Finite Mixture Distributions*. Wiley Series in Probability and Mathematical Statistics. John Wiley & Sons, New York, London, Sydney, 1985.

L. Tong, R.-W. Liu, V. C. Soon, and Y.-F. Huang. The indeterminacy and identifiability of blind identification. *IEEE Trans. Circuits and Systems*, 38(5):499–509, May 1991.

V. Tresp, R. Neuneier, and S. Ahmad. Efficient methods for dealing with missing data in supervised learning. In Tesauro et al. (1995), pages 689–696.

A. Treves, S. Panzeri, E. T. Rolls, M. Booth, and E. A. Wakeman. Firing rate distributions and efficiency of information transmission of inferior temporal cortex neurons to natural visual stimuli. *Neural Computation*, 11(3):601–632, Mar. 1999.

A. Treves and E. T. Rolls. What determines the capacity of autoassociative memories in the brain? *Network: Computation in Neural Systems*, 2(4):371–397, Nov. 1991.

A. C. Tsoi. Recurrent neural network architectures — an overview. In C. L. Giles and M. Gori, editors, *Adaptive Processing of Temporal Information*, volume 1387 of *Lecture Notes in Artificial Intelligence*, pages 1–26. Springer-Verlag, New York, 1998.

UCLA. Artificial EPG palate image. The UCLA Phonetics Lab. Available online at `http://www.humnet.ucla.edu/humnet/linguistics/faciliti/facilities/physiology/EGP_picture.JPG`, Feb. 1, 2000.

N. Ueda, R. Nakano, Z. Ghahramani, and G. E. Hinton. SMEM algorithm for mixture models. *Neural Computation*, 12(9):2109–2128, Sept. 2000.

A. Utsugi. Hyperparameter selection for self-organizing maps. *Neural Computation*, 9(3):623–635, Apr. 1997a.

A. Utsugi. Topology selection for self-organizing maps. *Network: Computation in Neural Systems*, 7(4): 727–740, 1997b.

A. Utsugi. Bayesian sampling and ensemble learning in generative topographic mapping. *Neural Processing Letters*, 12(3):277–290, Dec. 2000.

A. Utsugi and T. Kumagai. Bayesian analysis of mixtures of factor analyzers. *Neural Computation*, 13(5): 993–1002, May 2001.

V. N. Vapnik and S. Mukherjee. Support vector method for multivariate density estimation. In Solla et al. (2000), pages 659–665.

S. V. Vaseghi. *Advanced Signal Processing and Digital Noise Reduction*. John Wiley & Sons, New York, London, Sydney, second edition, 2000.

S. V. Vaseghi and P. J. W. Rayner. Detection and suppression of impulsive noise in speech-communication systems. *IEE Proc. I (Communications, Speech and Vision)*, 137(1):38–46, Feb. 1990.

T. Villmann, R. Der, M. Hermann, and T. M. Martinetz. Topology preservation in self-organizing feature maps: Exact definition and measurement. *IEEE Trans. Neural Networks*, 8(2):256–266, Mar. 1997.

W. E. Vinje and J. L. Gallant. Sparse coding and decorrelation in primary visual cortex during natural vision. *Science*, 287(5456):1273–1276, Feb. 18 2000.

H. M. Wagner. *Principles of Operations Research with Applications to Managerial Decisions*. Prentice-Hall, Englewood Cliffs, N.J., second edition, 1975.

A. Webb. *Statistical Pattern Recognition*. Edward Arnold, 1999.

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

E. J. Wegman. Hyperdimensional data analysis using parallel coordinates. *J. Amer. Stat. Assoc.*, 85(411): 664–675, Sept. 1990.

J. R. Westbury. *X-Ray Microbeam Speech Production Database User's Handbook Version 1.0*. Waisman Center on Mental Retardation & Human Development, University of Wisconsin, Madison, WI, June 1994. With the assistance of Greg Turner & Jim Dembowski.

J. R. Westbury, M. Hashi, and M. J. Lindstrom. Differences among speakers in lingual articulation for American English /ɹ/. *Speech Communication*, 26(3):203–226, Nov. 1998.

J. Weston, A. Gammerman, M. O. Stitson, V. Vapnik, V. Vovk, and C. Watkins. Support vector density estimation. In Schölkopf et al. (1999a), chapter 18, pages 293–306.

J. Whittaker. *Graphical Models in Applied Multivariate Statistics*. Wiley Series in Probability and Mathematical Statistics. John Wiley & Sons, New York, London, Sydney, 1990.

P. Whittle. On principal components and least square methods of factor analysis. *Skand. Aktur. Tidskr.*, 36: 223–239, 1952.

J. Wiles, P. Bakker, A. Lynton, M. Norris, S. Parkinson, M. Staples, and A. Whiteside. Using bottlenecks in feedforward networks as a dimension reduction technique: An application to optimization tasks. *Neural Computation*, 8(6):1179–1183, Aug. 1996.

J. H. Wilkinson. *The Algebraic Eigenvalue Problem*. Oxford University Press, New York, Oxford, 1965.

P. M. Williams. Using neural networks to model conditional multivariate densities. *Neural Computation*, 8 (4):843–854, May 1996.

B. Willmore, P. A. Watters, and D. J. Tolhurst. A comparison of natural-image-based models of simple-cell coding. *Perception*, 29(9):1017–1040, Sept. 2000.

R. Wilson and M. Spann. A new approach to clustering. *Pattern Recognition*, 23(12):1413–1425, 1990.

J. H. Wolfe. Pattern clustering by multivariate mixture analysis. *Multivariate Behavioral Research*, 5:329–350, July 1970.

D. M. Wolpert and Z. Ghahramani. Computational principles of movement neuroscience. *Nat. Neurosci.*, 3 (Supp.):1212–1217, Nov. 2000.

D. M. Wolpert and M. Kawato. Multiple paired forward and inverse models for motor control. *Neural Networks*, 11(7–8):1317–1329, Oct. 1998.

A. A. Wrench. A multi-channel/multi-speaker articulatory database for continuous speech recognition research. In *Phonus*, volume 5, Saarbrücken, 2000. Institute of Phonetics, University of Saarland.

F. Xie and D. van Compernolle. Speech enhancement by spectral magnitude estimation —a unifying approach. *Speech Communication*, 19(2):89–104, Aug. 1996.

L. Xu, C. C. Cheung, and S. Amari. Learned parametric mixture based ICA algorithm. *Neurocomputing*, 22 (1–3):69–80, Nov. 1998.

E. Yamamoto, S. Nakamura, and K. Shikano. Lip movement synthesis from speech based on hidden Markov models. *Speech Communication*, 26(1–2):105–115, 1998.

H. H. Yang and S. Amari. Adaptive on-line learning algorithms for blind separation — maximum entropy and minimum mutual information. *Neural Computation*, 9(7):1457–1482, Oct. 1997.

H. Yehia and F. Itakura. A method to combine acoustic and morphological constraints in the speech production inverse problem. *Speech Communication*, 18(2):151–174, Apr. 1996.

H. Yehia, T. Kuratate, and E. Vatikiotis-Bateson. Using speech acoustics to drive facial motion. In Ohala et al. (1999), pages 631–634.

H. Yehia, P. Rubin, and E. Vatikiotis-Bateson. Quantitative association of vocal-tract and facial behavior. *Speech Communication*, 26(1–2):23–43, Oct. 1998.

G. Young. Maximum likelihood estimation and factor analysis. *Psychometrika*, 6:49–53, 1940.

S. J. Young. A review of large vocabulary continuous speech recognition. *IEEE Signal Processing Magazine*, 13(5):45–57, Sept. 1996.

K. Zhang, I. Ginzburg, B. L. McNaughton, and T. J. Sejnowski. Interpreting neuronal population activity by reconstruction: Unified framework with application to hippocampal place cells. *J. Neurophysiol.*, 79(2): 1017–1044, Feb. 1998.

R. D. Zhang and J.-G. Postaire. Convexity dependent morphological transformations for mode detection in cluster-analysis. *Pattern Recognition*, 27(1):135–148, 1994.

Y. Zhao and C. G. Atkeson. Implementing projection pursuit learning. *IEEE Trans. Neural Networks*, 7(2): 362–373, Mar. 1996.

I. Zlokarnik. Adding articulatory features to acoustic features for automatic speech recognition. *J. Acoustic Soc. Amer.*, 97(5):3246, May 1995a.

I. Zlokarnik. Articulatory kinematics from the standpoint of automatic speech recognition. *J. Acoustic Soc. Amer.*, 98(5):2930–2931, Nov. 1995b.