8.6 Principal component analysis

We have already discussed the problems which can arise in attempts to perform pattern recognition in high-dimensional spaces, and the potential improvements which can be achieved by first mapping the data into a space of lower dimensionality. In general, a reduction in the dimensionality of the input space will be accompanied by a loss of some of the information which discriminates between different classes (or, more generally, which determines the target values). The goal in dimensionality reduction is therefore to preserve as much of the relevant information as possible. We have already discussed one approach to dimensionality reduction based on the selection of a subset of a given set of features or inputs. Here we consider techniques for combining inputs together to make a (generally smaller) set of features. The procedures we shall discuss in this section rely entirely on the input data itself without reference to the corresponding target data, and can be regarded as a form of *unsupervised* learning. While they are of great practical significance, the neglect of the target data information implies they can also be significantly sub-optimal, as we discuss in Section 8.6.3.

We begin our discussion of unsupervised techniques for dimensionality reduction by restricting our attention to linear transformations. Our goal is to map vectors $\mathbf{x}^n$ in a d-dimensional space $(x_1, \ldots, x_d)$ onto vectors $\mathbf{z}^n$ in an M-dimensional space $(z_1, \ldots, z_M)$, where $M < d$. We first note that the vector $\mathbf{x}$ can be represented, without loss of generality, as a linear combination of a set of d orthonormal vectors $\mathbf{u}_i$

$$\mathbf{x} = \sum_{i=1}^{d} z_i \mathbf{u}_i \tag{8.12}$$

where the vectors $\mathbf{u}_i$ satisfy the orthonormality relation

$$\mathbf{u}_i^{\mathrm{T}}\mathbf{u}_j = \delta_{ij} \tag{8.13}$$

in which δ_{ij} is the Kronecker delta symbol defined on page xiii. Explicit expressions for the coefficients z_i in (8.12) can be found by using (8.13) to give

$$z_i = \mathbf{u}_i^{\mathrm{T}}\mathbf{x} \tag{8.14}$$

which can be regarded as a simple rotation of the coordinate system from the original x's to a new set of coordinates given by the z's (Appendix A). Now suppose that we retain only a subset $M < d$ of the basis vectors $\mathbf{u}_i$, so that we use only M coefficients z_i. The remaining coefficients will be replaced by constants b_i so that each vector $\mathbf{x}$ is approximated by an expression of the form

$$\widetilde{\mathbf{x}} = \sum_{i=1}^{M} z_i\mathbf{u}_i + \sum_{i=M+1}^{d} b_i\mathbf{u}_i. \tag{8.15}$$

This represents a form of dimensionality reduction since the original vector $\mathbf{x}$ which contained d degrees of freedom must now be approximated by a new vector $\mathbf{z}$ which has $M < d$ degrees of freedom. Now consider a whole data set of N vectors $\mathbf{x}^n$ where $n = 1, \ldots, N$. We wish to choose the basis vectors $\mathbf{u}_i$ and the coefficients b_i such that the approximation given by (8.15), with the values of z_i determined by (8.14), gives the best approximation to the original vector $\mathbf{x}$ on average for the whole data set. The error in the vector $\mathbf{x}^n$ introduced by the dimensionality reduction is given by

$$\mathbf{x}^n - \widetilde{\mathbf{x}}^n = \sum_{i=M+1}^{d} (z_i^n - b_i)\mathbf{u}_i. \tag{8.16}$$

We can then define the best approximation to be that which minimizes the sum of the squares of the errors over the whole data set. Thus, we minimize

$$E_M = \frac{1}{2}\sum_{n=1}^{N} \|\mathbf{x}^n - \widetilde{\mathbf{x}}^n\|^2 = \frac{1}{2}\sum_{n=1}^{N}\sum_{i=M+1}^{d} (z_i^n - b_i)^2 \tag{8.17}$$

where we have used the orthonormality relation (8.13). If we set the derivative of E_M with respect to b_i to zero we find

$$b_i = \frac{1}{N}\sum_{n=1}^{N} z_i^n = \mathbf{u}_i^{\mathrm{T}}\overline{\mathbf{x}} \tag{8.18}$$

where we have defined the mean vector $\overline{\mathbf{x}}$ to be

$$\overline{\mathbf{x}} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}^n. \tag{8.19}$$

Using (8.14) and (8.18) we can write the sum-of-squares error (8.17) as

$$E_M = \frac{1}{2}\sum_{i=M+1}^{d}\sum_{n=1}^{N}\left\{\mathbf{u}_i^{\mathrm{T}}(\mathbf{x}^n - \overline{\mathbf{x}})\right\}^2$$

$$= \frac{1}{2}\sum_{i=M+1}^{d}\mathbf{u}_i^{\mathrm{T}}\boldsymbol{\Sigma}\mathbf{u}_i \tag{8.20}$$

where $\boldsymbol{\Sigma}$ is the covariance matrix of the set of vectors $\{\mathbf{x}^n\}$ and is given by

$$\boldsymbol{\Sigma} = \sum_n (\mathbf{x}^n - \overline{\mathbf{x}})(\mathbf{x}^n - \overline{\mathbf{x}})^{\mathrm{T}}. \tag{8.21}$$

There now remains the task of minimizing E_M with respect to the choice of basis vectors $\mathbf{u}_i$. It is shown in Appendix E that the minimum occurs when the basis vectors satisfy

$$\boldsymbol{\Sigma}\mathbf{u}_i = \lambda_i\mathbf{u}_i \tag{8.22}$$

so that they are the eigenvectors of the covariance matrix. Note that, since the covariance matrix is real and symmetric, its eigenvectors can indeed be chosen to be orthonormal as assumed. Substituting (8.22) into (8.20), and making use of the orthonormality relation (8.13), we obtain the value of the error criterion at the minimum in the form

$$E_M = \frac{1}{2}\sum_{i=M+1}^{d}\lambda_i. \tag{8.23}$$

Thus, the minimum error is obtained by choosing the $d-M$ smallest eigenvalues, and their corresponding eigenvectors, as the ones to discard.

The linear dimensionality reduction procedure derived above is called the *Karhunen–Loéve transformation* or *principal component analysis* and is discussed at length in Jollife (1986). Each of the eigenvectors $\mathbf{u}_i$ is called a *principal component.* The technique is illustrated schematically in Figure 8.9 for the case of data points in two dimensions.

In practice, the algorithm proceeds by first computing the mean of the vectors $\mathbf{x}^n$ and then subtracting off this mean. Then the covariance matrix is calculated

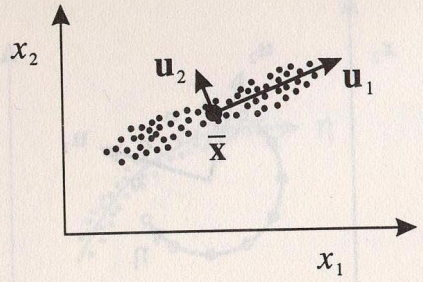

Figure 8.9. Schematic illustration of principal component analysis applied to data in two dimensions. In a linear projection down to one dimension, the optimum choice of projection, in the sense of minimizing the sum-of-squares error, is obtained by first subtracting off the mean $\bar{\mathbf{x}}$ of the data set, and then projecting onto the first eigenvector $\mathbf{u}_1$ of the covariance matrix.

and its eigenvectors and eigenvalues are found. The eigenvectors corresponding to the M *largest* eigenvalues are retained and the input vectors $\mathbf{x}^n$ are projected onto the eigenvectors to give the components of the transformed vectors $\mathbf{z}^n$ in the M-dimensional space. Thus, in Figure 8.9, each two-dimensional data point is transformed to a single variable z_1 representing the projection of the data point onto the eigenvector $\mathbf{u}_1$.

The error introduced by a dimensionality reduction using principal component analysis can be evaluated using (8.23). In some applications the original data has a very high dimensionality and we wish only to retain the first few principal components. In such cases use can be made of efficient algorithms which allow only the required eigenvectors, corresponding to the largest few eigenvalues, to be evaluated (Press *et al.*, 1992).

We have considered linear dimensionality reduction based on the sum-of-squares error criterion. It is possible to consider other criteria including data covariance measures and population entropy. These give rise to the same result for the optimal dimensionality reduction in terms of projections onto the eigenvectors of $\boldsymbol{\Sigma}$ corresponding to the largest eigenvalues (Fukunaga, 1990).

8.6.1 *Intrinsic dimensionality*

Suppose we are given a set of data vectors in a d-dimensional space, and we apply principal component analysis and discover that the first d' eigenvalues have significantly larger values than the remaining $d-d'$ eigenvalues. This tells us that the data can be represented to a relatively high accuracy by projection onto the first d' eigenvectors. We therefore discover that the effective dimensionality of the data is less than the apparent dimensionality d, as a result of correlations within the data. However, principal component analysis is limited by virtue of being a linear technique. It may therefore be unable to capture more complex non-linear correlations, and may therefore overestimate the true dimensionality

a

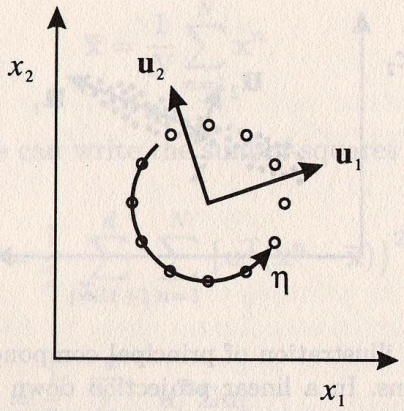

Figure 8.10. Example of a data set in two dimensions which has an intrinsic dimensionality $d' = 1$. The data can be specified not only in terms of the two variables x_1 and x_2, but also in terms of the single parameter η. However, a linear dimensionality reduction technique, such as principal component analysis, is unable to detect the lower dimensionality.

of the data. This is illustrated schematically in Figure 8.10, for data points which lie around the perimeter of a circle. Principal component analysis would give two eigenvectors with equal eigenvalues (as a result of the symmetry of the data). In fact, however, the data could be described equally well by a single parameter η as shown. More generally, a data set in d dimensions is said to have an *intrinsic dimensionality* equal to d' if the data lies entirely within a d'-dimensional subspace (Fukunaga, 1982).

Note that if the data is slightly noisy, then the intrinsic dimensionality may be increased. Figure 8.11 shows some data in two dimensions which is corrupted by a small level of noise. Strictly the data now lives in a two-dimensional space, but can nevertheless by represented to high accuracy by a single parameter.

8.6.2 *Neural networks for dimensionality reduction*

Multi-layer neural networks can themselves be used to perform non-linear dimensionality reduction, thereby overcoming some of the limitations of linear principal component analysis. Consider first a multi-layer perceptron of the form shown in Figure 8.12, having d inputs, d output units and M hidden units, with $M < d$ (Rumelhart *et al.*, 1986). The targets used to train the network are simply the input vectors themselves, so that the network is attempting to map each input vector onto itself. Due to the reduced number of units in the first layer, a perfect reconstruction of all input vectors is not in general possible. The network can be trained by minimizing a sum-of-squares error of the form

$$E = \frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{d}\{y_k(\mathbf{x}^n) - x_k^n\}^2. \tag{8.24}$$