CHAPTER 7 Dimensionality Reduction

We saw in Chapter 6 that high-dimensional data has some peculiar characteristics, some of which are counterintuitive. For example, in high dimensions the center of the space is devoid of points, with most of the points being scattered along the surface of the space or in the corners. There is also an apparent proliferation of orthogonal axes. As a consequence high-dimensional data can cause problems for data mining and analysis, although in some cases high-dimensionality can help, for example, for nonlinear classification. Nevertheless, it is important to check whether the dimensionality can be reduced while preserving the essential properties of the full data matrix. This can aid data visualization as well as data mining. In this chapter we study methods that allow us to obtain optimal lower-dimensional projections of the data.

7.1 BACKGROUND

Let the data $\mathbf{D}$ consist of n points over d attributes, that is, it is an $n \times d$ matrix, given as

$$\mathbf{D} = \begin{pmatrix} & X_1 & X_2 & \cdots & X_d \\ \mathbf{x}_1 & x_{11} & x_{12} & \cdots & x_{1d} \\ \mathbf{x}_2 & x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_n & x_{n1} & x_{n2} & \cdots & x_{nd} \end{pmatrix}$$

Each point $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{id})^T$ is a vector in the ambient d-dimensional vector space spanned by the d standard basis vectors $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_d$, where $\mathbf{e}_i$ corresponds to the ith attribute X_i. Recall that the standard basis is an orthonormal basis for the data space, that is, the basis vectors are pairwise orthogonal, $\mathbf{e}_i^T \mathbf{e}_j = 0$, and have unit length $\|\mathbf{e}_i\| = 1$.

As such, given any other set of d orthonormal vectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_d$, with $\mathbf{u}_i^T \mathbf{u}_j = 0$ and $\|\mathbf{u}_i\| = 1$ (or $\mathbf{u}_i^T \mathbf{u}_i = 1$), we can re-express each point $\mathbf{x}$ as the linear combination

$$\mathbf{x} = a_1 \mathbf{u}_1 + a_2 \mathbf{u}_2 + \cdots + a_d \mathbf{u}_d \tag{7.1}$$

where the vector $\mathbf{a} = (a_1, a_2, \ldots, a_d)^T$ represents the coordinates of $\mathbf{x}$ in the new basis. The above linear combination can also be expressed as a matrix multiplication:

$$\mathbf{x} = \mathbf{U}\mathbf{a} \tag{7.2}$$

where $\mathbf{U}$ is the $d \times d$ matrix, whose ith column comprises the ith basis vector $\mathbf{u}_i$:

$$\mathbf{U} = \begin{pmatrix} | & | & & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_d \\ | & | & & | \end{pmatrix}$$

The matrix $\mathbf{U}$ is an *orthogonal* matrix, whose columns, the basis vectors, are *orthonormal*, that is, they are pairwise orthogonal and have unit length

$$\mathbf{u}_i^T \mathbf{u}_j = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

Because $\mathbf{U}$ is orthogonal, this means that its inverse equals its transpose:

$$\mathbf{U}^{-1} = \mathbf{U}^T$$

which implies that $\mathbf{U}^T\mathbf{U} = \mathbf{I}$, where $\mathbf{I}$ is the $d \times d$ identity matrix.

Multiplying Eq. (7.2) on both sides by $\mathbf{U}^T$ yields the expression for computing the coordinates of $\mathbf{x}$ in the new basis

$$\begin{aligned} \mathbf{U}^T\mathbf{x} &= \mathbf{U}^T\mathbf{U}\mathbf{a} \\ \mathbf{a} &= \mathbf{U}^T\mathbf{x} \end{aligned} \tag{7.3}$$

Example 7.1. Figure 7.1a shows the centered Iris dataset, with $n = 150$ points, in the $d = 3$ dimensional space comprising the `sepal length` (X_1), `sepal width` (X_2), and `petal length` (X_3) attributes. The space is spanned by the standard basis vectors

$$\mathbf{e}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \qquad \mathbf{e}_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \qquad \mathbf{e}_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

Figure 7.1b shows the same points in the space comprising the new basis vectors

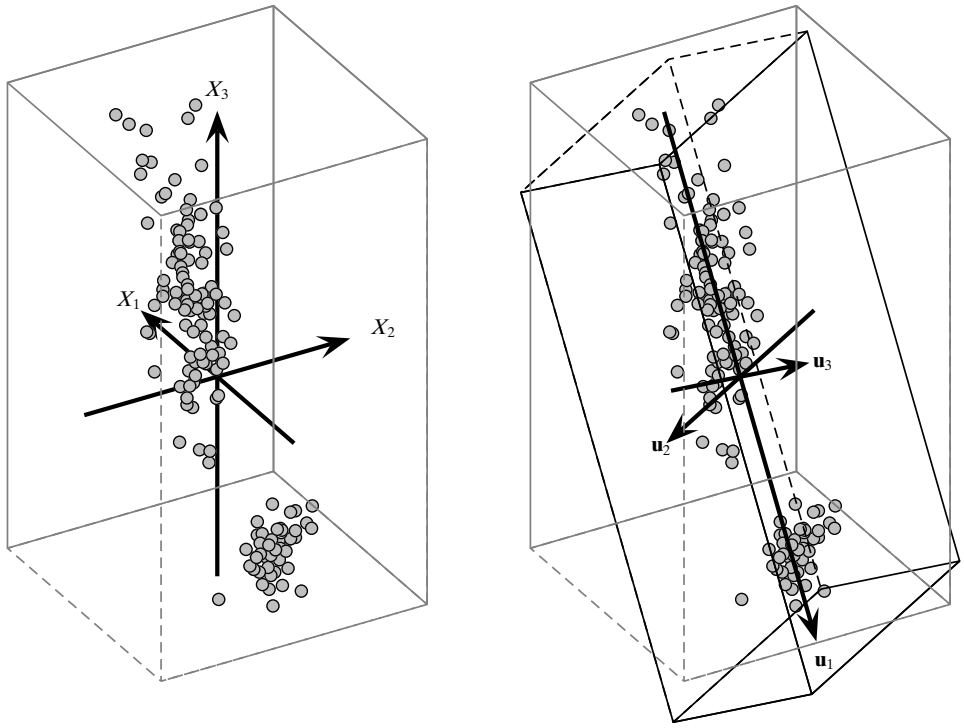
$$\mathbf{u}_1 = \begin{pmatrix} -0.390 \\ 0.089 \\ -0.916 \end{pmatrix} \qquad \mathbf{u}_2 = \begin{pmatrix} -0.639 \\ -0.742 \\ 0.200 \end{pmatrix} \qquad \mathbf{u}_3 = \begin{pmatrix} -0.663 \\ 0.664 \\ 0.346 \end{pmatrix}$$

For example, the new coordinates of the centered point $\mathbf{x} = (-0.343, -0.754, 0.241)^T$ can be computed as

$$\mathbf{a} = \mathbf{U}^T\mathbf{x} = \begin{pmatrix} -0.390 & 0.089 & -0.916 \\ -0.639 & -0.742 & 0.200 \\ -0.663 & 0.664 & 0.346 \end{pmatrix} \begin{pmatrix} -0.343 \\ -0.754 \\ 0.241 \end{pmatrix} = \begin{pmatrix} -0.154 \\ 0.828 \\ -0.190 \end{pmatrix}$$

One can verify that $\mathbf{x}$ can be written as the linear combination

$$\mathbf{x} = -0.154\mathbf{u}_1 + 0.828\mathbf{u}_2 - 0.190\mathbf{u}_3$$



(a) Original Basis (b) Optimal Basis

Figure 7.1. Iris data: optimal basis in three dimensions.

Because there are potentially infinite choices for the set of orthonormal basis vectors, one natural question is whether there exists an *optimal* basis, for a suitable notion of optimality. Further, it is often the case that the input dimensionality d is very large, which can cause various problems owing to the curse of dimensionality (see Chapter 6). It is natural to ask whether we can find a reduced dimensionality subspace that still preserves the essential characteristics of the data. That is, we are interested in finding the optimal r-dimensional representation of $\mathbf{D}$, with $r \ll d$. In other words, given a point $\mathbf{x}$, and assuming that the basis vectors have been sorted in decreasing order of importance, we can truncate its linear expansion [Eq. (7.1)] to just r terms, to obtain

$$\mathbf{x}' = a_1\mathbf{u}_1 + a_2\mathbf{u}_2 + \cdots + a_r\mathbf{u}_r = \sum_{i=1}^{r} a_i\mathbf{u}_i \tag{7.4}$$

Here $\mathbf{x}'$ is the projection of $\mathbf{x}$ onto the first r basis vectors, which can be written in matrix notation as follows:

$$\mathbf{x}' = \begin{pmatrix} | & | & & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_r \\ | & | & & | \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_r \end{pmatrix} = \mathbf{U}_r\mathbf{a}_r \tag{7.5}$$

where $\mathbf{U}_r$ is the matrix comprising the first r basis vectors, and $\mathbf{a}_r$ is vector comprising the first r coordinates. Further, because $\mathbf{a} = \mathbf{U}^T\mathbf{x}$ from Eq. (7.3), restricting it to the first r terms, we get

$$\mathbf{a}_r = \mathbf{U}_r^T\mathbf{x} \tag{7.6}$$

Plugging this into Eq. (7.5), the projection of $\mathbf{x}$ onto the first r basis vectors can be compactly written as

$$\mathbf{x}' = \mathbf{U}_r\mathbf{U}_r^T\mathbf{x} = \mathbf{P}_r\mathbf{x} \tag{7.7}$$

where $\mathbf{P}_r = \mathbf{U}_r\mathbf{U}_r^T$ is the *orthogonal projection matrix* for the subspace spanned by the first r basis vectors. That is, $\mathbf{P}_r$ is symmetric and $\mathbf{P}_r^2 = \mathbf{P}_r$. This is easy to verify because $\mathbf{P}_r^T = (\mathbf{U}_r\mathbf{U}_r^T)^T = \mathbf{U}_r\mathbf{U}_r^T = \mathbf{P}_r$, and $\mathbf{P}_r^2 = (\mathbf{U}_r\mathbf{U}_r^T)(\mathbf{U}_r\mathbf{U}_r^T) = \mathbf{U}_r\mathbf{U}_r^T = \mathbf{P}_r$, where we use the observation that $\mathbf{U}_r^T\mathbf{U}_r = \mathbf{I}_{r\times r}$, the $r \times r$ identity matrix. The projection matrix $\mathbf{P}_r$ can also be written as the decomposition

$$\mathbf{P}_r = \mathbf{U}_r\mathbf{U}_r^T = \sum_{i=1}^{r}\mathbf{u}_i\mathbf{u}_i^T \tag{7.8}$$

From Eqs. (7.1) and (7.4), the projection of $\mathbf{x}$ onto the remaining dimensions comprises the *error vector*

$$\boldsymbol{\epsilon} = \sum_{i=r+1}^{d} a_i\mathbf{u}_i = \mathbf{x} - \mathbf{x}'$$

It is worth noting that that $\mathbf{x}'$ and $\boldsymbol{\epsilon}$ are orthogonal vectors:

$$\mathbf{x}'^T\boldsymbol{\epsilon} = \sum_{i=1}^{r}\sum_{j=r+1}^{d} a_i a_j \mathbf{u}_i^T\mathbf{u}_j = 0$$

This is a consequence of the basis being orthonormal. In fact, we can make an even stronger statement. The subspace spanned by the first r basis vectors

$$S_r = span\,(\mathbf{u}_1, \ldots, \mathbf{u}_r)$$

and the subspace spanned by the remaining basis vectors

$$S_{d-r} = span\,(\mathbf{u}_{r+1}, \ldots, \mathbf{u}_d)$$

are *orthogonal subspaces*, that is, all pairs of vectors $\mathbf{x} \in S_r$ and $\mathbf{y} \in S_{d-r}$ must be orthogonal. The subspace S_{d-r} is also called the *orthogonal complement* of S_r.

Example 7.2. Continuing Example 7.1, approximating the centered point $\mathbf{x} = (-0.343, -0.754, 0.241)^T$ by using only the first basis vector $\mathbf{u}_1 = (-0.390, 0.089, -0.916)^T$, we have

$$\mathbf{x}' = a_1\mathbf{u}_1 = -0.154\mathbf{u}_1 = \begin{pmatrix} 0.060 \\ -0.014 \\ 0.141 \end{pmatrix}$$

The projection of $\mathbf{x}$ on $\mathbf{u}_1$ could have been obtained directly from the projection matrix

$$\mathbf{P}_1 = \mathbf{u}_1\mathbf{u}_1^T = \begin{pmatrix} -0.390 \\ 0.089 \\ -0.916 \end{pmatrix} \begin{pmatrix} -0.390 & 0.089 & -0.916 \end{pmatrix}$$

$$= \begin{pmatrix} 0.152 & -0.035 & 0.357 \\ -0.035 & 0.008 & -0.082 \\ 0.357 & -0.082 & 0.839 \end{pmatrix}$$

That is

$$\mathbf{x}' = \mathbf{P}_1\mathbf{x} = \begin{pmatrix} 0.060 \\ -0.014 \\ 0.141 \end{pmatrix}$$

The error vector is given as

$$\boldsymbol{\epsilon} = a_2\mathbf{u}_2 + a_3\mathbf{u}_3 = \mathbf{x} - \mathbf{x}' = \begin{pmatrix} -0.40 \\ -0.74 \\ 0.10 \end{pmatrix}$$

One can verify that $\mathbf{x}'$ and $\boldsymbol{\epsilon}$ are orthogonal, i.e.,

$$\mathbf{x}'^T\boldsymbol{\epsilon} = \begin{pmatrix} 0.060 & -0.014 & 0.141 \end{pmatrix} \begin{pmatrix} -0.40 \\ -0.74 \\ 0.10 \end{pmatrix} = 0$$

The goal of dimensionality reduction is to seek an r-dimensional basis that gives the best possible approximation $\mathbf{x}_i'$ over all the points $\mathbf{x}_i \in \mathbf{D}$. Alternatively, we may seek to minimize the error $\boldsymbol{\epsilon}_i = \mathbf{x}_i - \mathbf{x}_i'$ over all the points.

7.2 PRINCIPAL COMPONENT ANALYSIS

Principal Component Analysis (PCA) is a technique that seeks a r-dimensional basis that best captures the variance in the data. The direction with the largest projected variance is called the first principal component. The orthogonal direction that captures the second largest projected variance is called the second principal component, and so on. As we shall see, the direction that maximizes the variance is also the one that minimizes the mean squared error.

7.2.1 Best Line Approximation

We will start with $r = 1$, that is, the one-dimensional subspace or line $\mathbf{u}$ that best approximates $\mathbf{D}$ in terms of the variance of the projected points. This will lead to the general PCA technique for the best $1 \leq r \leq d$ dimensional basis for $\mathbf{D}$.

Without loss of generality, we assume that $\mathbf{u}$ has magnitude $\|\mathbf{u}\|^2 = \mathbf{u}^T\mathbf{u} = 1$; otherwise it is possible to keep on increasing the projected variance by simply

increasing the magnitude of $\mathbf{u}$. We also assume that the data has been centered so that it has mean $\boldsymbol{\mu} = \mathbf{0}$.

The projection of $\mathbf{x}_i$ on the vector $\mathbf{u}$ is given as

$$\mathbf{x}_i' = \left(\frac{\mathbf{u}^T\mathbf{x}_i}{\mathbf{u}^T\mathbf{u}}\right)\mathbf{u} = (\mathbf{u}^T\mathbf{x}_i)\mathbf{u} = a_i\mathbf{u}$$

where the scalar

$$a_i = \mathbf{u}^T\mathbf{x}_i$$

gives the coordinate of $\mathbf{x}_i'$ along $\mathbf{u}$. Note that because the mean point is $\boldsymbol{\mu} = \mathbf{0}$, its coordinate along $\mathbf{u}$ is $\mu_{\mathbf{u}} = 0$.

We have to choose the direction $\mathbf{u}$ such that the variance of the projected points is maximized. The projected variance along $\mathbf{u}$ is given as

$$\begin{aligned}
\sigma_{\mathbf{u}}^2 &= \frac{1}{n}\sum_{i=1}^{n}(a_i - \mu_{\mathbf{u}})^2 \\
&= \frac{1}{n}\sum_{i=1}^{n}(\mathbf{u}^T\mathbf{x}_i)^2 \\
&= \frac{1}{n}\sum_{i=1}^{n}\mathbf{u}^T\left(\mathbf{x}_i\mathbf{x}_i^T\right)\mathbf{u} \\
&= \mathbf{u}^T\left(\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i\mathbf{x}_i^T\right)\mathbf{u} \\
&= \mathbf{u}^T\boldsymbol{\Sigma}\mathbf{u}
\end{aligned} \tag{7.9}$$

where $\boldsymbol{\Sigma}$ is the covariance matrix for the centered data $\mathbf{D}$.

To maximize the projected variance, we have to solve a constrained optimization problem, namely to maximize $\sigma_{\mathbf{u}}^2$ subject to the constraint that $\mathbf{u}^T\mathbf{u} = 1$. This can be solved by introducing a Lagrangian multiplier α for the constraint, to obtain the unconstrained maximization problem

$$\max_{\mathbf{u}} J(\mathbf{u}) = \mathbf{u}^T\boldsymbol{\Sigma}\mathbf{u} - \alpha(\mathbf{u}^T\mathbf{u} - 1) \tag{7.10}$$

Setting the derivative of $J(\mathbf{u})$ with respect to $\mathbf{u}$ to the zero vector, we obtain

$$\begin{aligned}
\frac{\partial}{\partial\mathbf{u}}J(\mathbf{u}) &= \mathbf{0} \\
\frac{\partial}{\partial\mathbf{u}}\left(\mathbf{u}^T\boldsymbol{\Sigma}\mathbf{u} - \alpha(\mathbf{u}^T\mathbf{u} - 1)\right) &= \mathbf{0} \\
2\boldsymbol{\Sigma}\mathbf{u} - 2\alpha\mathbf{u} &= \mathbf{0} \\
\boldsymbol{\Sigma}\mathbf{u} &= \alpha\mathbf{u}
\end{aligned} \tag{7.11}$$

This implies that α is an eigenvalue of the covariance matrix $\boldsymbol{\Sigma}$, with the associated eigenvector $\mathbf{u}$. Further, taking the dot product with $\mathbf{u}$ on both sides of Eq. (7.11) yields

$$\mathbf{u}^T\boldsymbol{\Sigma}\mathbf{u} = \mathbf{u}^T\alpha\mathbf{u}$$

From Eq. (7.9), we then have

$$\sigma_{\mathbf{u}}^2 = \alpha \mathbf{u}^T \mathbf{u}$$
$$\text{or } \sigma_{\mathbf{u}}^2 = \alpha \tag{7.12}$$

To maximize the projected variance $\sigma_{\mathbf{u}}^2$, we should thus choose the largest eigenvalue of $\mathbf{\Sigma}$. In other words, the dominant eigenvector $\mathbf{u}_1$ specifies the direction of most variance, also called the *first principal component*, that is, $\mathbf{u} = \mathbf{u}_1$. Further, the largest eigenvalue λ_1 specifies the projected variance, that is, $\sigma_{\mathbf{u}}^2 = \alpha = \lambda_1$.

Minimum Squared Error Approach

We now show that the direction that maximizes the projected variance is also the one that minimizes the average squared error. As before, assume that the dataset $\mathbf{D}$ has been centered by subtracting the mean from each point. For a point $\mathbf{x}_i \in \mathbf{D}$, let $\mathbf{x}_i'$ denote its projection along the direction $\mathbf{u}$, and let $\boldsymbol{\epsilon}_i = \mathbf{x}_i - \mathbf{x}_i'$ denote the error vector. The mean squared error (*MSE*) optimization condition is defined as

$$MSE(\mathbf{u}) = \frac{1}{n}\sum_{i=1}^{n} \|\boldsymbol{\epsilon}_i\|^2 \tag{7.13}$$
$$= \frac{1}{n}\sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{x}_i'\|^2$$
$$= \frac{1}{n}\sum_{i=1}^{n} (\mathbf{x}_i - \mathbf{x}_i')^T(\mathbf{x}_i - \mathbf{x}_i')$$
$$= \frac{1}{n}\sum_{i=1}^{n} \left(\|\mathbf{x}_i\|^2 - 2\mathbf{x}_i^T\mathbf{x}_i' + (\mathbf{x}_i')^T\mathbf{x}_i' \right) \tag{7.14}$$

Noting that $\mathbf{x}_i' = (\mathbf{u}^T\mathbf{x}_i)\mathbf{u}$, we have

$$= \frac{1}{n}\sum_{i=1}^{n} \left(\|\mathbf{x}_i\|^2 - 2\mathbf{x}_i^T(\mathbf{u}^T\mathbf{x}_i)\mathbf{u} + \left((\mathbf{u}^T\mathbf{x}_i)\mathbf{u}\right)^T(\mathbf{u}^T\mathbf{x}_i)\mathbf{u} \right)$$
$$= \frac{1}{n}\sum_{i=1}^{n} \left(\|\mathbf{x}_i\|^2 - 2(\mathbf{u}^T\mathbf{x}_i)(\mathbf{x}_i^T\mathbf{u}) + (\mathbf{u}^T\mathbf{x}_i)(\mathbf{x}_i^T\mathbf{u})\mathbf{u}^T\mathbf{u} \right)$$
$$= \frac{1}{n}\sum_{i=1}^{n} \left(\|\mathbf{x}_i\|^2 - (\mathbf{u}^T\mathbf{x}_i)(\mathbf{x}_i^T\mathbf{u}) \right)$$
$$= \frac{1}{n}\sum_{i=1}^{n} \|\mathbf{x}_i\|^2 - \frac{1}{n}\sum_{i=1}^{n} \mathbf{u}^T(\mathbf{x}_i\mathbf{x}_i^T)\mathbf{u}$$
$$= \frac{1}{n}\sum_{i=1}^{n} \|\mathbf{x}_i\|^2 - \mathbf{u}^T\left(\frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i\mathbf{x}_i^T\right)\mathbf{u}$$
$$= \sum_{i=1}^{n} \frac{\|\mathbf{x}_i\|^2}{n} - \mathbf{u}^T\mathbf{\Sigma}\mathbf{u} \tag{7.15}$$

Note that by Eq. (1.4) the total variance of the centered data (i.e., with $\boldsymbol{\mu} = \mathbf{0}$) is given as

$$var(\mathbf{D}) = \frac{1}{n}\sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{0}\|^2 = \frac{1}{n}\sum_{i=1}^{n} \|\mathbf{x}_i\|^2$$

Further, by Eq. (2.28), we have

$$var(\mathbf{D}) = tr(\boldsymbol{\Sigma}) = \sum_{i=1}^{d} \sigma_i^2$$

Thus, we may rewrite Eq. (7.15) as

$$MSE(\mathbf{u}) = var(\mathbf{D}) - \mathbf{u}^T\boldsymbol{\Sigma}\mathbf{u} = \sum_{i=1}^{d} \sigma_i^2 - \mathbf{u}^T\boldsymbol{\Sigma}\mathbf{u}$$

Because the first term, $var(\mathbf{D})$, is a constant for a given dataset $\mathbf{D}$, the vector $\mathbf{u}$ that minimizes $MSE(\mathbf{u})$ is thus the same one that maximizes the second term, the projected variance $\mathbf{u}^T\boldsymbol{\Sigma}\mathbf{u}$. Because we know that $\mathbf{u}_1$, the dominant eigenvector of $\boldsymbol{\Sigma}$, maximizes the projected variance, we have

$$MSE(\mathbf{u}_1) = var(\mathbf{D}) - \mathbf{u}_1^T\boldsymbol{\Sigma}\mathbf{u}_1 = var(\mathbf{D}) - \mathbf{u}_1^T\lambda_1\mathbf{u}_1 = var(\mathbf{D}) - \lambda_1 \tag{7.16}$$

Thus, the principal component $\mathbf{u}_1$, which is the direction that maximizes the projected variance, is also the direction that minimizes the mean squared error.

Example 7.3. Figure 7.2 shows the first principal component, that is, the best one-dimensional approximation, for the three dimensional Iris dataset shown in Figure 7.1a. The covariance matrix for this dataset is given as

$$\boldsymbol{\Sigma} = \begin{pmatrix} 0.681 & -0.039 & 1.265 \\ -0.039 & 0.187 & -0.320 \\ 1.265 & -0.320 & 3.092 \end{pmatrix}$$

The variance values σ_i^2 for each of the original dimensions are given along the main diagonal of $\boldsymbol{\Sigma}$. For example, $\sigma_1^2 = 0.681$, $\sigma_2^2 = 0.187$, and $\sigma_3^2 = 3.092$. The largest eigenvalue of $\boldsymbol{\Sigma}$ is $\lambda_1 = 3.662$, and the corresponding dominant eigenvector is $\mathbf{u}_1 = (-0.390, 0.089, -0.916)^T$. The unit vector $\mathbf{u}_1$ thus maximizes the projected variance, which is given as $J(\mathbf{u}_1) = \alpha = \lambda_1 = 3.662$. Figure 7.2 plots the principal component $\mathbf{u}_1$. It also shows the error vectors $\boldsymbol{\epsilon}_i$, as thin gray line segments.

The total variance of the data is given as

$$var(\mathbf{D}) = \frac{1}{n}\sum_{i=1}^{n} \|\mathbf{x}\|^2 = \frac{1}{150} \cdot 594.04 = 3.96$$

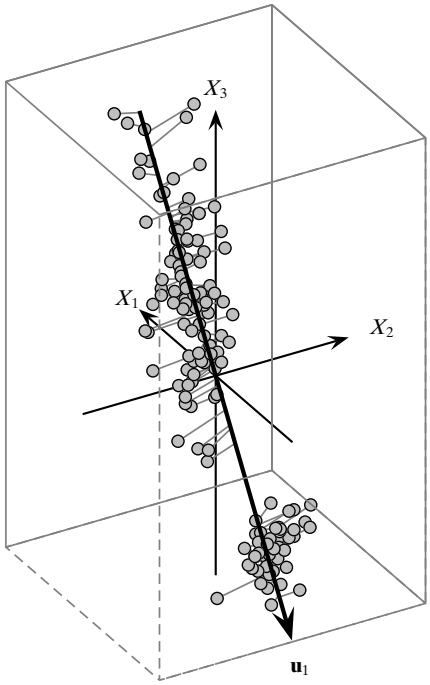


Figure 7.2. Best one-dimensional or line approximation.

We can also directly obtain the total variance as the trace of the covariance matrix:

$$var(\mathbf{D}) = tr(\mathbf{\Sigma}) = \sigma_1^2 + \sigma_2^2 + \sigma_3^2 = 0.681 + 0.187 + 3.092 = 3.96$$

Thus, using Eq. (7.16), the minimum value of the mean squared error is given as

$$MSE(\mathbf{u}_1) = var(\mathbf{D}) - \lambda_1 = 3.96 - 3.662 = 0.298$$

7.2.2 Best 2-dimensional Approximation

We are now interested in the best two-dimensional approximation to $\mathbf{D}$. As before, assume that $\mathbf{D}$ has already been centered, so that $\boldsymbol{\mu} = \mathbf{0}$. We already computed the direction with the most variance, namely $\mathbf{u}_1$, which is the eigenvector corresponding to the largest eigenvalue λ_1 of $\mathbf{\Sigma}$. We now want to find another direction $\mathbf{v}$, which also maximizes the projected variance, but is orthogonal to $\mathbf{u}_1$. According to Eq. (7.9) the projected variance along $\mathbf{v}$ is given as

$$\sigma_{\mathbf{v}}^2 = \mathbf{v}^T \mathbf{\Sigma} \mathbf{v}$$

We further require that $\mathbf{v}$ be a unit vector orthogonal to $\mathbf{u}_1$, that is,

$$\mathbf{v}^T \mathbf{u}_1 = 0$$
$$\mathbf{v}^T \mathbf{v} = 1$$

The optimization condition then becomes

$$\max_{\mathbf{v}} J(\mathbf{v}) = \mathbf{v}^T\boldsymbol{\Sigma}\mathbf{v} - \alpha(\mathbf{v}^T\mathbf{v} - 1) - \beta(\mathbf{v}^T\mathbf{u}_1 - 0) \tag{7.17}$$

Taking the derivative of $J(\mathbf{v})$ with respect to $\mathbf{v}$, and setting it to the zero vector, gives

$$2\boldsymbol{\Sigma}\mathbf{v} - 2\alpha\mathbf{v} - \beta\mathbf{u}_1 = \mathbf{0} \tag{7.18}$$

If we multiply on the left by $\mathbf{u}_1^T$ we get

$$\begin{aligned} 2\mathbf{u}_1^T\boldsymbol{\Sigma}\mathbf{v} - 2\alpha\mathbf{u}_1^T\mathbf{v} - \beta\mathbf{u}_1^T\mathbf{u}_1 &= 0 \\ 2\mathbf{v}^T\boldsymbol{\Sigma}\mathbf{u}_1 - \beta &= 0, \text{which implies that} \\ \beta &= 2\mathbf{v}^T\lambda_1\mathbf{u}_1 = 2\lambda_1\mathbf{v}^T\mathbf{u}_1 = 0 \end{aligned}$$

In the derivation above we used the fact that $\mathbf{u}_1^T\boldsymbol{\Sigma}\mathbf{v} = \mathbf{v}^T\boldsymbol{\Sigma}\mathbf{u}_1$, and that $\mathbf{v}$ is orthogonal to $\mathbf{u}_1$. Plugging $\beta = 0$ into Eq. (7.18) gives us

$$\begin{aligned} 2\boldsymbol{\Sigma}\mathbf{v} - 2\alpha\mathbf{v} &= \mathbf{0} \\ \boldsymbol{\Sigma}\mathbf{v} &= \alpha\mathbf{v} \end{aligned}$$

This means that $\mathbf{v}$ is another eigenvector of $\boldsymbol{\Sigma}$. Also, as in Eq. (7.12), we have $\sigma_{\mathbf{v}}^2 = \alpha$. To maximize the variance along $\mathbf{v}$, we should choose $\alpha = \lambda_2$, the second largest eigenvalue of $\boldsymbol{\Sigma}$, with the *second principal component* being given by the corresponding eigenvector, that is, $\mathbf{v} = \mathbf{u}_2$.

Total Projected Variance

Let $\mathbf{U}_2$ be the matrix whose columns correspond to the two principal components, given as

$$\mathbf{U}_2 = \begin{pmatrix} | & | \\ \mathbf{u}_1 & \mathbf{u}_2 \\ | & | \end{pmatrix}$$

Given the point $\mathbf{x}_i \in \mathbf{D}$ its coordinates in the two-dimensional subspace spanned by $\mathbf{u}_1$ and $\mathbf{u}_2$ can be computed via Eq. (7.6), as follows:

$$\mathbf{a}_i = \mathbf{U}_2^T\mathbf{x}_i$$

Assume that each point $\mathbf{x}_i \in \mathbb{R}^d$ in $\mathbf{D}$ has been projected to obtain its coordinates $\mathbf{a}_i \in \mathbb{R}^2$, yielding the new dataset $\mathbf{A}$. Further, because $\mathbf{D}$ is assumed to be centered, with $\boldsymbol{\mu} = \mathbf{0}$, the coordinates of the projected mean are also zero because $\mathbf{U}_2^T\boldsymbol{\mu} = \mathbf{U}_2^T\mathbf{0} = \mathbf{0}$.

The total variance for $\mathbf{A}$ is given as

$$\begin{aligned} var(\mathbf{A}) &= \frac{1}{n}\sum_{i=1}^{n} \|\mathbf{a}_i - \mathbf{0}\|^2 \\ &= \frac{1}{n}\sum_{i=1}^{n} \left(\mathbf{U}_2^T\mathbf{x}_i\right)^T \left(\mathbf{U}_2^T\mathbf{x}_i\right) \\ &= \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i^T \left(\mathbf{U}_2\mathbf{U}_2^T\right)\mathbf{x}_i \\ &= \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i^T \mathbf{P}_2 \mathbf{x}_i \end{aligned} \tag{7.19}$$

where $\mathbf{P}_2$ is the orthogonal projection matrix [Eq. (7.8)] given as

$$\mathbf{P}_2 = \mathbf{U}_2\mathbf{U}_2^T = \mathbf{u}_1\mathbf{u}_1^T + \mathbf{u}_2\mathbf{u}_2^T$$

Substituting this into Eq. (7.19), the projected total variance is given as

$$\begin{aligned} var(\mathbf{A}) &= \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i^T \mathbf{P}_2 \mathbf{x}_i \qquad (7.20) \\ &= \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i^T \left(\mathbf{u}_1\mathbf{u}_1^T + \mathbf{u}_2\mathbf{u}_2^T\right)\mathbf{x}_i \\ &= \frac{1}{n}\sum_{i=1}^{n} (\mathbf{u}_1^T\mathbf{x}_i)(\mathbf{x}_i^T\mathbf{u}_1) + \frac{1}{n}\sum_{i=1}^{n} (\mathbf{u}_2^T\mathbf{x}_i)(\mathbf{x}_i^T\mathbf{u}_2) \\ &= \mathbf{u}_1^T\boldsymbol{\Sigma}\mathbf{u}_1 + \mathbf{u}_2^T\boldsymbol{\Sigma}\mathbf{u}_2 \qquad (7.21) \end{aligned}$$

Because $\mathbf{u}_1$ and $\mathbf{u}_2$ are eigenvectors of $\boldsymbol{\Sigma}$, we have $\boldsymbol{\Sigma}\mathbf{u}_1 = \lambda_1\mathbf{u}_1$ and $\boldsymbol{\Sigma}\mathbf{u}_2 = \lambda_2\mathbf{u}_2$, so that

$$var(\mathbf{A}) = \mathbf{u}_1^T\boldsymbol{\Sigma}\mathbf{u}_1 + \mathbf{u}_2^T\boldsymbol{\Sigma}\mathbf{u}_2 = \mathbf{u}_1^T\lambda_1\mathbf{u}_1 + \mathbf{u}_2^T\lambda_2\mathbf{u}_2 = \lambda_1 + \lambda_2 \tag{7.22}$$

Thus, the sum of the eigenvalues is the total variance of the projected points, and the first two principal components maximize this variance.

Mean Squared Error

We now show that the first two principal components also minimize the mean square error objective. The mean square error objective is given as

$$\begin{aligned} MSE &= \frac{1}{n}\sum_{i=1}^{n} \left\|\mathbf{x}_i - \mathbf{x}_i'\right\|^2 \\ &= \frac{1}{n}\sum_{i=1}^{n} \left(\|\mathbf{x}_i\|^2 - 2\mathbf{x}_i^T\mathbf{x}_i' + (\mathbf{x}_i')^T\mathbf{x}_i' \right), \text{ using Eq. (7.14)} \\ &= var(\mathbf{D}) + \frac{1}{n}\sum_{i=1}^{n} \left(-2\mathbf{x}_i^T\mathbf{P}_2\mathbf{x}_i + (\mathbf{P}_2\mathbf{x}_i)^T\mathbf{P}_2\mathbf{x}_i\right), \text{ using Eq. (7.7) that } \mathbf{x}_i' = \mathbf{P}_2\mathbf{x}_i \end{aligned}$$

$$= var(\mathbf{D}) - \frac{1}{n}\sum_{i=1}^{n}\left(\mathbf{x}_i^T\mathbf{P}_2\mathbf{x}_i\right)$$

$$= var(\mathbf{D}) - var(\mathbf{A}), \text{ using Eq. (7.20)} \tag{7.23}$$

Thus, the MSE objective is minimized precisely when the total projected variance $var(\mathbf{A})$ is maximized. From Eq. (7.22), we have

$$MSE = var(\mathbf{D}) - \lambda_1 - \lambda_2$$

Example 7.4. For the Iris dataset from Example 7.1, the two largest eigenvalues are $\lambda_1 = 3.662$, and $\lambda_2 = 0.239$, with the corresponding eigenvectors:

$$\mathbf{u}_1 = \begin{pmatrix} -0.390 \\ 0.089 \\ -0.916 \end{pmatrix} \qquad \mathbf{u}_2 = \begin{pmatrix} -0.639 \\ -0.742 \\ 0.200 \end{pmatrix}$$

The projection matrix is given as

$$\mathbf{P}_2 = \mathbf{U}_2\mathbf{U}_2^T = \begin{pmatrix} | & | \\ \mathbf{u}_1 & \mathbf{u}_2 \\ | & | \end{pmatrix}\begin{pmatrix} — & \mathbf{u}_1^T & — \\ — & \mathbf{u}_2^T & — \end{pmatrix} = \mathbf{u}_1\mathbf{u}_1^T + \mathbf{u}_2\mathbf{u}_2^T$$

$$= \begin{pmatrix} 0.152 & -0.035 & 0.357 \\ -0.035 & 0.008 & -0.082 \\ 0.357 & -0.082 & 0.839 \end{pmatrix} + \begin{pmatrix} 0.408 & 0.474 & -0.128 \\ 0.474 & 0.551 & -0.148 \\ -0.128 & -0.148 & 0.04 \end{pmatrix}$$

$$= \begin{pmatrix} 0.560 & 0.439 & 0.229 \\ 0.439 & 0.558 & -0.230 \\ 0.229 & -0.230 & 0.879 \end{pmatrix}$$

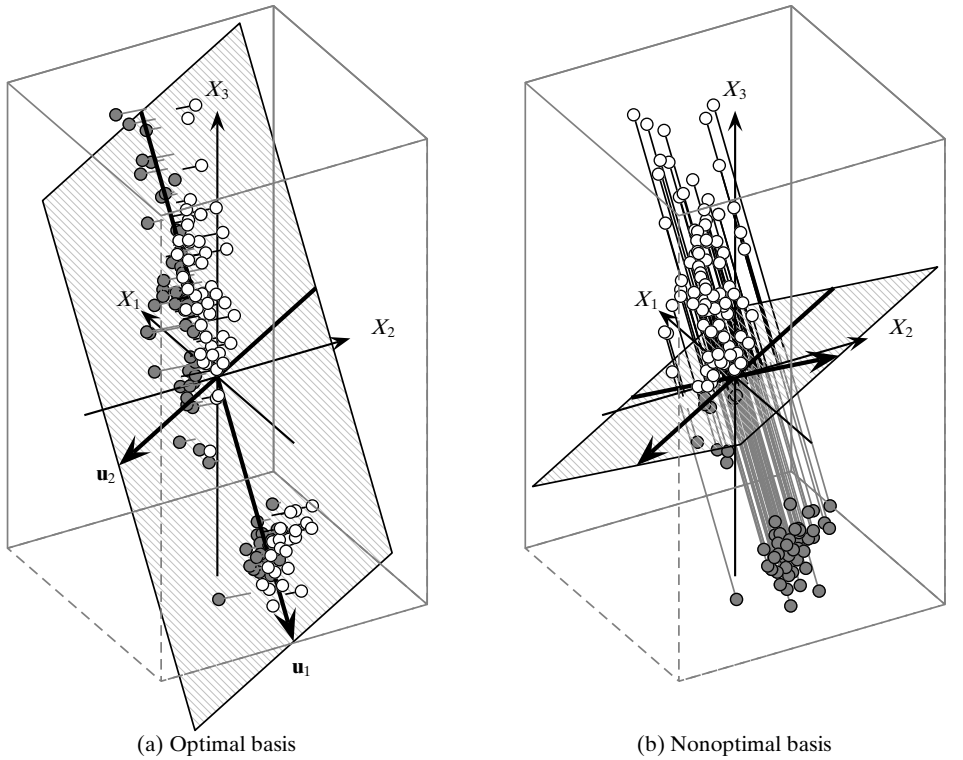
Thus, each point $\mathbf{x}_i$ can be approximated by its projection onto the first two principal components $\mathbf{x}_i' = \mathbf{P}_2\mathbf{x}_i$. Figure 7.3a plots this optimal 2-dimensional subspace spanned by $\mathbf{u}_1$ and $\mathbf{u}_2$. The error vector $\boldsymbol{\epsilon}_i$ for each point is shown as a thin line segment. The gray points are behind the 2-dimensional subspace, whereas the white points are in front of it. The total variance captured by the subspace is given as

$$\lambda_1 + \lambda_2 = 3.662 + 0.239 = 3.901$$

The mean squared error is given as

$$MSE = var(\mathbf{D}) - \lambda_1 - \lambda_2 = 3.96 - 3.662 - 0.239 = 0.059$$

Figure 7.3b plots a nonoptimal 2-dimensional subspace. As one can see the optimal subspace maximizes the variance, and minimizes the squared error, whereas the nonoptimal subspace captures less variance, and has a high mean squared error value, which can be pictorially seen from the lengths of the error vectors (line segments). In fact, this is the worst possible 2-dimensional subspace; its MSE is 3.662.



(a) Optimal basis (b) Nonoptimal basis

Figure 7.3. Best two-dimensional approximation.

7.2.3 Best *r*-dimensional Approximation

We are now interested in the best r-dimensional approximation to $\mathbf{D}$, where $2 < r \leq d$. Assume that we have already computed the first $j-1$ principal components or eigenvectors, $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{j-1}$, corresponding to the $j-1$ largest eigenvalues of $\boldsymbol{\Sigma}$, for $1 \leq j \leq r$. To compute the jth new basis vector $\mathbf{v}$, we have to ensure that it is normalized to unit length, that is, $\mathbf{v}^T\mathbf{v} = 1$, and is orthogonal to all previous components $\mathbf{u}_i$, i.e., $\mathbf{u}_i^T\mathbf{v} = 0$, for $1 \leq i < j$. As before, the projected variance along $\mathbf{v}$ is given as

$$\sigma_{\mathbf{v}}^2 = \mathbf{v}^T\boldsymbol{\Sigma}\mathbf{v}$$

Combined with the constraints on $\mathbf{v}$, this leads to the following maximization problem with Lagrange multipliers:

$$\max_{\mathbf{v}} J(\mathbf{v}) = \mathbf{v}^T\boldsymbol{\Sigma}\mathbf{v} - \alpha(\mathbf{v}^T\mathbf{v} - 1) - \sum_{i=1}^{j-1} \beta_i(\mathbf{u}_i^T\mathbf{v} - 0)$$

Taking the derivative of $J(\mathbf{v})$ with respect to $\mathbf{v}$ and setting it to the zero vector gives

$$2\boldsymbol{\Sigma}\mathbf{v} - 2\alpha\mathbf{v} - \sum_{i=1}^{j-1} \beta_i\mathbf{u}_i = \mathbf{0} \tag{7.24}$$

If we multiply on the left by $\mathbf{u}_k^T$, for $1 \leq k < j$, we get

$$2\mathbf{u}_k^T\boldsymbol{\Sigma}\mathbf{v} - 2\alpha\mathbf{u}_k^T\mathbf{v} - \beta_k\mathbf{u}_k^T\mathbf{u}_k - \sum_{\substack{i=1 \\ i\neq k}}^{j-1}\beta_i\mathbf{u}_k^T\mathbf{u}_i = 0$$

$$2\mathbf{v}^T\boldsymbol{\Sigma}\mathbf{u}_k - \beta_k = 0$$

$$\beta_k = 2\mathbf{v}^T\lambda_k\mathbf{u}_k = 2\lambda_k\mathbf{v}^T\mathbf{u}_k = 0$$

where we used the fact that $\boldsymbol{\Sigma}\mathbf{u}_k = \lambda_k\mathbf{u}_k$, as $\mathbf{u}_k$ is the eigenvector corresponding to the kth largest eigenvalue λ_k of $\boldsymbol{\Sigma}$. Thus, we find that $\beta_i = 0$ for all $i < j$ in Eq. (7.24), which implies that

$$\boldsymbol{\Sigma}\mathbf{v} = \alpha\mathbf{v}$$

To maximize the variance along $\mathbf{v}$, we set $\alpha = \lambda_j$, the jth largest eigenvalue of $\boldsymbol{\Sigma}$, with $\mathbf{v} = \mathbf{u}_j$ giving the jth principal component.

In summary, to find the best r-dimensional approximation to $\mathbf{D}$, we compute the eigenvalues of $\boldsymbol{\Sigma}$. Because $\boldsymbol{\Sigma}$ is positive semidefinite, its eigenvalues must all be non-negative, and we can thus sort them in decreasing order as follows:

$$\lambda_1 \geq \lambda_2 \geq \cdots \lambda_r \geq \lambda_{r+1} \cdots \geq \lambda_d \geq 0$$

We then select the r largest eigenvalues, and their corresponding eigenvectors to form the best r-dimensional approximation.

Total Projected Variance

Let $\mathbf{U}_r$ be the r-dimensional basis vector matrix

$$\mathbf{U}_r = \begin{pmatrix} | & | & & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_r \\ | & | & & | \end{pmatrix}$$

with the projection matrix given as

$$\mathbf{P}_r = \mathbf{U}_r\mathbf{U}_r^T = \sum_{i=1}^{r}\mathbf{u}_i\mathbf{u}_i^T$$

Let $\mathbf{A}$ denote the dataset formed by the coordinates of the projected points in the r-dimensional subspace, that is, $\mathbf{a}_i = \mathbf{U}_r^T\mathbf{x}_i$, and let $\mathbf{x}_i' = \mathbf{P}_r\mathbf{x}_i$ denote the projected point in the original d-dimensional space. Following the derivation for Eqs. (7.19), (7.21), and (7.22), the projected variance is given as

$$var(\mathbf{A}) = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i^T\mathbf{P}_r\mathbf{x}_i = \sum_{i=1}^{r}\mathbf{u}_i^T\boldsymbol{\Sigma}\mathbf{u}_i = \sum_{i=1}^{r}\lambda_i$$

Thus, the total projected variance is simply the sum of the r largest eigenvalues of $\boldsymbol{\Sigma}$.

Mean Squared Error

Based on the derivation for Eq. (7.23), the mean squared error objective in r dimensions can be written as

$$\begin{aligned} MSE &= \frac{1}{n}\sum_{i=1}^{n}\left\|\mathbf{x}_i - \mathbf{x}_i'\right\|^2 \\ &= var(\mathbf{D}) - var(\mathbf{A}) \\ &= var(\mathbf{D}) - \sum_{i=1}^{r}\mathbf{u}_i^T\boldsymbol{\Sigma}\mathbf{u}_i \\ &= var(\mathbf{D}) - \sum_{i=1}^{r}\lambda_i \end{aligned}$$

The first r-principal components maximize the projected variance $var(\mathbf{A})$, and thus they also minimize the MSE.

Total Variance

Note that the total variance of $\mathbf{D}$ is invariant to a change in basis vectors. Therefore, we have the following identity:

$$var(\mathbf{D}) = \sum_{i=1}^{d}\sigma_i^2 = \sum_{i=1}^{d}\lambda_i$$

Choosing the Dimensionality

Often we may not know how many dimensions, r, to use for a good approximation. One criteria for choosing r is to compute the fraction of the total variance captured by the first r principal components, computed as

$$f(r) = \frac{\lambda_1 + \lambda_2 + \cdots + \lambda_r}{\lambda_1 + \lambda_2 + \cdots + \lambda_d} = \frac{\sum_{i=1}^{r}\lambda_i}{\sum_{i=1}^{d}\lambda_i} = \frac{\sum_{i=1}^{r}\lambda_i}{var(\mathbf{D})} \tag{7.25}$$

Given a certain desired variance threshold, say α, starting from the first principal component, we keep on adding additional components, and stop at the smallest value r, for which $f(r) \geq \alpha$. In other words, we select the fewest number of dimensions such that the subspace spanned by those r dimensions captures at least α fraction of the total variance. In practice, α is usually set to 0.9 or higher, so that the reduced dataset captures at least 90% of the total variance.

Algorithm 7.1 gives the pseudo-code for the principal component analysis algorithm. Given the input data $\mathbf{D} \in \mathbb{R}^{n\times d}$, it first centers it by subtracting the mean from each point. Next, it computes the eigenvectors and eigenvalues of the covariance matrix $\boldsymbol{\Sigma}$. Given the desired variance threshold α, it selects the smallest set of dimensions r that capture at least α fraction of the total variance. Finally, it computes the coordinates of each point in the new r-dimensional principal component subspace, to yield the new data matrix $\mathbf{A} \in \mathbb{R}^{n\times r}$.

ALGORITHM 7.1. Principal Component Analysis

PCA ($\mathbf{D}, \alpha$):

1 $\boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i$ // compute mean
2 $\mathbf{Z} = \mathbf{D} - \mathbf{1} \cdot \boldsymbol{\mu}^T$ // center the data
3 $\boldsymbol{\Sigma} = \frac{1}{n}\left(\mathbf{Z}^T\mathbf{Z}\right)$ // compute covariance matrix
4 $(\lambda_1, \lambda_2, \ldots, \lambda_d) = \text{eigenvalues}(\boldsymbol{\Sigma})$ // compute eigenvalues
5 $\mathbf{U} = \begin{pmatrix}\mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_d\end{pmatrix} = \text{eigenvectors}(\boldsymbol{\Sigma})$ // compute eigenvectors
6 $f(r) = \frac{\sum_{i=1}^{r}\lambda_i}{\sum_{i=1}^{d}\lambda_i}$, for all $r = 1, 2, \ldots, d$ // fraction of total variance
7 Choose smallest r so that $f(r) \geq \alpha$ // choose dimensionality
8 $\mathbf{U}_r = \begin{pmatrix}\mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_r\end{pmatrix}$ // reduced basis
9 $\mathbf{A} = \{\mathbf{a}_i \mid \mathbf{a}_i = \mathbf{U}_r^T\mathbf{x}_i, \text{for } i = 1, \ldots, n\}$ // reduced dimensionality data

Example 7.5. Given the 3-dimensional Iris dataset in Figure 7.1a, its covariance matrix is

$$\boldsymbol{\Sigma} = \begin{pmatrix} 0.681 & -0.039 & 1.265 \\ -0.039 & 0.187 & -0.320 \\ 1.265 & -0.32 & 3.092 \end{pmatrix}$$

The eigenvalues and eigenvectors of $\boldsymbol{\Sigma}$ are given as

$$\lambda_1 = 3.662 \qquad \lambda_2 = 0.239 \qquad \lambda_3 = 0.059$$

$$\mathbf{u}_1 = \begin{pmatrix} -0.390 \\ 0.089 \\ -0.916 \end{pmatrix} \qquad \mathbf{u}_2 = \begin{pmatrix} -0.639 \\ -0.742 \\ 0.200 \end{pmatrix} \qquad \mathbf{u}_3 = \begin{pmatrix} -0.663 \\ 0.664 \\ 0.346 \end{pmatrix}$$

The total variance is therefore $\lambda_1 + \lambda_2 + \lambda_3 = 3.662 + 0.239 + 0.059 = 3.96$. The optimal 3-dimensional basis is shown in Figure 7.1b.

To find a lower dimensional approximation, let $\alpha = 0.95$. The fraction of total variance for different values of r is given as

r	1	2	3
$f(r)$	0.925	0.985	1.0

For example, for $r = 1$, the fraction of total variance is given as $f(1) = \frac{3.662}{3.96} = 0.925$. Thus, we need at least $r = 2$ dimensions to capture 95% of the total variance. This optimal 2-dimensional subspace is shown as the shaded plane in Figure 7.3a. The reduced dimensionality dataset $\mathbf{A}$ is shown in Figure 7.4. It consists of the point coordinates $\mathbf{a}_i = \mathbf{U}_2^T\mathbf{x}_i$ in the new 2-dimensional principal components basis comprising $\mathbf{u}_1$ and $\mathbf{u}_2$.

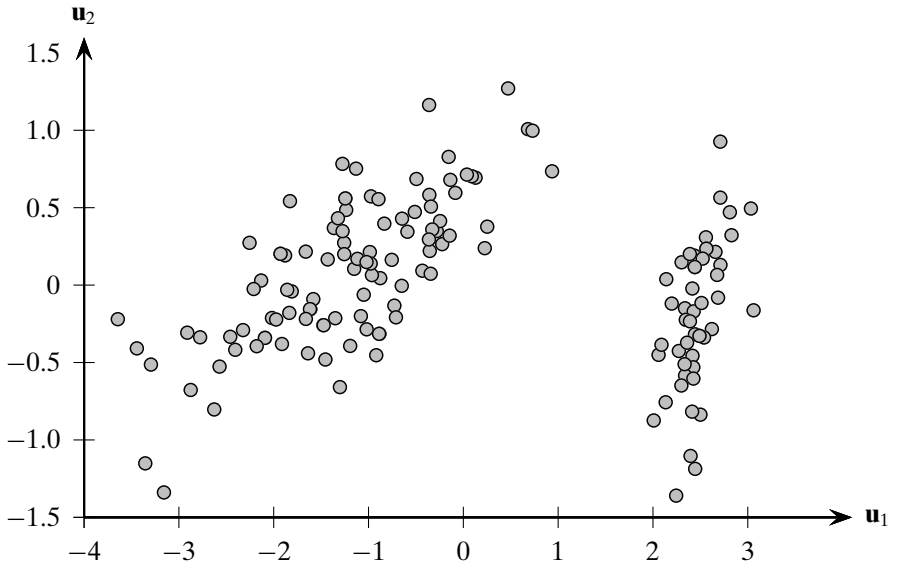


Figure 7.4. Reduced dimensionality dataset: Iris principal components.

7.2.4 Geometry of PCA

Geometrically, when $r = d$, PCA corresponds to a orthogonal change of basis, so that the total variance is captured by the sum of the variances along each of the principal directions $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_d$, and further, all covariances are zero. This can be seen by looking at the collective action of the full set of principal components, which can be arranged in the $d \times d$ orthogonal matrix

$$\mathbf{U} = \begin{pmatrix} | & | & & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_d \\ | & | & & | \end{pmatrix}$$

with $\mathbf{U}^{-1} = \mathbf{U}^T$.

Each principal component $\mathbf{u}_i$ corresponds to an eigenvector of the covariance matrix $\mathbf{\Sigma}$, that is,

$$\mathbf{\Sigma}\mathbf{u}_i = \lambda_i \mathbf{u}_i \text{ for all } 1 \leq i \leq d$$

which can be written compactly in matrix notation as follows:

$$\mathbf{\Sigma}\begin{pmatrix} | & | & & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_d \\ | & | & & | \end{pmatrix} = \begin{pmatrix} | & | & & | \\ \lambda_1\mathbf{u}_1 & \lambda_2\mathbf{u}_2 & \cdots & \lambda_d\mathbf{u}_d \\ | & | & & | \end{pmatrix}$$

$$\mathbf{\Sigma}\mathbf{U} = \mathbf{U}\begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_d \end{pmatrix}$$

$$\mathbf{\Sigma}\mathbf{U} = \mathbf{U}\mathbf{\Lambda} \tag{7.26}$$

If we multiply Eq. (7.26) on the left by $\mathbf{U}^{-1} = \mathbf{U}^T$ we obtain

$$\mathbf{U}^T\boldsymbol{\Sigma}\mathbf{U} = \mathbf{U}^T\mathbf{U}\boldsymbol{\Lambda} = \boldsymbol{\Lambda} = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_d \end{pmatrix}$$

This means that if we change the basis to $\mathbf{U}$, we change the covariance matrix $\boldsymbol{\Sigma}$ to a similar matrix $\boldsymbol{\Lambda}$, which in fact is the covariance matrix in the new basis. The fact that $\boldsymbol{\Lambda}$ is diagonal confirms that after the change of basis, all of the covariances vanish, and we are left with only the variances along each of the principal components, with the variance along each new direction $\mathbf{u}_i$ being given by the corresponding eigenvalue λ_i.

It is worth noting that in the new basis, the equation

$$\mathbf{x}^T\boldsymbol{\Sigma}^{-1}\mathbf{x} = 1 \tag{7.27}$$

defines a d-dimensional ellipsoid (or hyper-ellipse). The eigenvectors $\mathbf{u}_i$ of $\boldsymbol{\Sigma}$, that is, the principal components, are the directions for the principal axes of the ellipsoid. The square roots of the eigenvalues, that is, $\sqrt{\lambda_i}$, give the lengths of the semi-axes.

Multiplying Eq. (7.26) on the right by $\mathbf{U}^{-1} = \mathbf{U}^T$, we have

$$\boldsymbol{\Sigma} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T \tag{7.28}$$

Assuming that $\boldsymbol{\Sigma}$ is invertible or nonsingular, we have

$$\boldsymbol{\Sigma}^{-1} = (\mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T)^{-1} = \left(\mathbf{U}^{-1}\right)^T\boldsymbol{\Lambda}^{-1}\mathbf{U}^{-1} = \mathbf{U}\boldsymbol{\Lambda}^{-1}\mathbf{U}^T$$

where

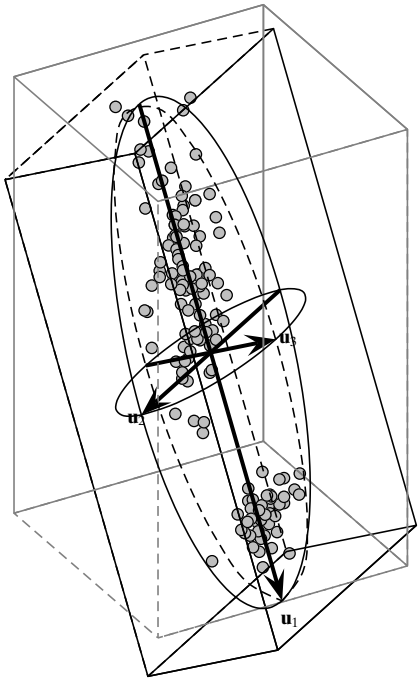
$$\boldsymbol{\Lambda}^{-1} = \begin{pmatrix} \frac{1}{\lambda_1} & 0 & \cdots & 0 \\ 0 & \frac{1}{\lambda_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{\lambda_d} \end{pmatrix}$$

Substituting $\boldsymbol{\Sigma}^{-1}$ in Eq. (7.27), and using the fact that $\mathbf{x} = \mathbf{U}\mathbf{a}$ from Eq. (7.2), where $\mathbf{a} = (a_1, a_2, \ldots, a_d)^T$ represents the coordinates of $\mathbf{x}$ in the new basis, we get

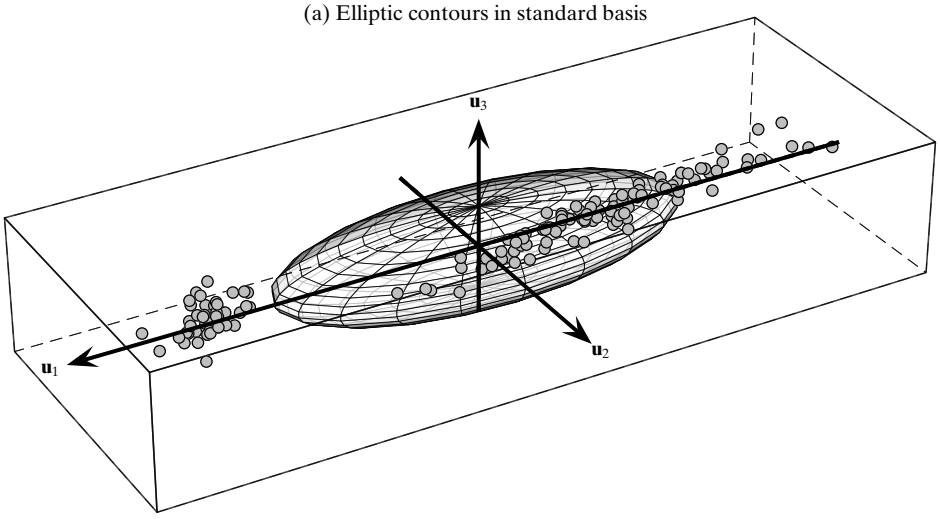
$$\begin{aligned} \mathbf{x}^T\boldsymbol{\Sigma}^{-1}\mathbf{x} &= 1 \\ \left(\mathbf{a}^T\mathbf{U}^T\right)\mathbf{U}\boldsymbol{\Lambda}^{-1}\mathbf{U}^T\left(\mathbf{U}\mathbf{a}\right) &= 1 \\ \mathbf{a}^T\boldsymbol{\Lambda}^{-1}\mathbf{a} &= 1 \\ \sum_{i=1}^{d}\frac{a_i^2}{\lambda_i} &= 1 \end{aligned}$$

which is precisely the equation for an ellipse centered at $\mathbf{0}$, with semi-axes lengths $\sqrt{\lambda_i}$. Thus $\mathbf{x}^T\boldsymbol{\Sigma}^{-1}\mathbf{x} = 1$, or equivalently $\mathbf{a}^T\boldsymbol{\Lambda}^{-1}\mathbf{a} = 1$ in the new principal components basis, defines an ellipsoid in d-dimensions, where the semi-axes lengths equal the standard deviations (squared root of the variance, $\sqrt{\lambda_i}$) along each axis. Likewise, the equation $\mathbf{x}^T\boldsymbol{\Sigma}^{-1}\mathbf{x} = s$, or equivalently $\mathbf{a}^T\boldsymbol{\Lambda}^{-1}\mathbf{a} = s$, for different values of the scalar s, represents concentric ellipsoids.

Example 7.6. Figure 7.5b shows the ellipsoid $\mathbf{x}^T\boldsymbol{\Sigma}^{-1}\mathbf{x} = \mathbf{a}^T\boldsymbol{\Lambda}^{-1}\mathbf{a} = 1$ in the new principal components basis. Each semi-axis length corresponds to the standard deviation $\sqrt{\lambda_i}$ along that axis. Because all pairwise covariances are zero in the principal components basis, the ellipsoid is axis-parallel, that is, each of its axes coincides with a basis vector.



(a) Elliptic contours in standard basis



(b) Axis parallel ellipsoid in principal components basis

Figure 7.5. Iris data: standard and principal components basis in three dimensions.

On the other hand, in the original standard d-dimensional basis for $\mathbf{D}$, the ellipsoid will not be axis-parallel, as shown by the contours of the ellipsoid in Figure 7.5a. Here the semi-axis lengths correspond to half the value range in each direction; the length was chosen so that the ellipsoid encompasses most of the points.

7.3 KERNEL PRINCIPAL COMPONENT ANALYSIS

Principal component analysis can be extended to find nonlinear "directions" in the data using kernel methods. Kernel PCA finds the directions of most variance in the feature space instead of the input space. That is, instead of trying to find linear combinations of the input dimensions, kernel PCA finds linear combinations in the high-dimensional feature space obtained as some nonlinear transformation of the input dimensions. Thus, the linear principal components in the feature space correspond to nonlinear directions in the input space. As we shall see, using the *kernel trick*, all operations can be carried out in terms of the kernel function in input space, without having to transform the data into feature space.

Example 7.7. Consider the nonlinear Iris dataset shown in Figure 7.6, obtained via a nonlinear transformation applied on the centered Iris data. In particular, the `sepal length` (A_1) and `sepal width` attributes (A_2) were transformed as follows:

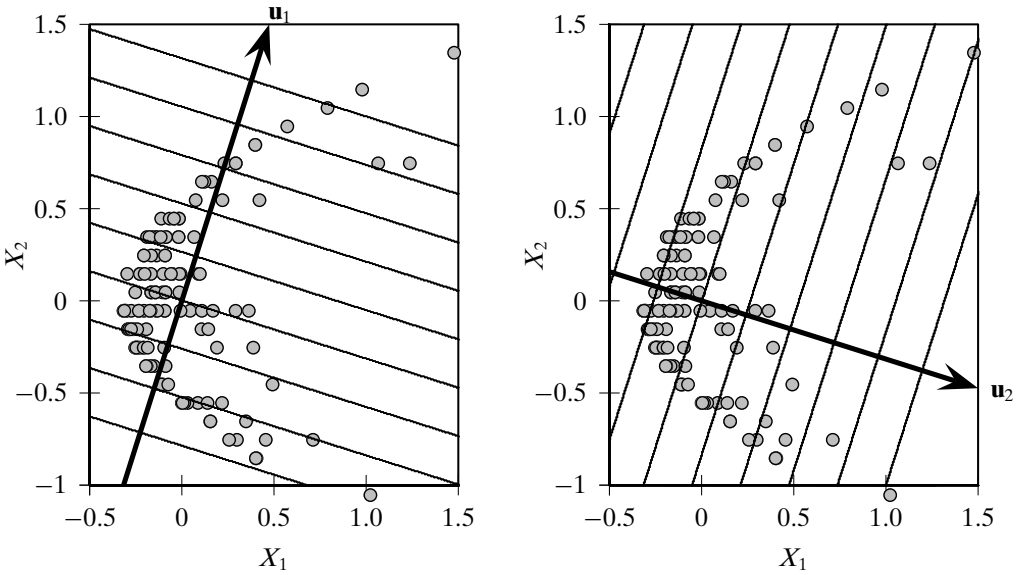
$$X_1 = 0.2A_1^2 + A_2^2 + 0.1A_1A_2$$
$$X_2 = A_2$$

The points show a clear quadratic (nonlinear) relationship between the two variables. Linear PCA yields the following two directions of most variance:

$$\lambda_1 = 0.197 \qquad \lambda_2 = 0.087$$
$$\mathbf{u}_1 = \begin{pmatrix} 0.301 \\ 0.953 \end{pmatrix} \qquad \mathbf{u}_2 = \begin{pmatrix} -0.953 \\ 0.301 \end{pmatrix}$$

These two principal components are illustrated in Figure 7.6. Also shown in the figure are lines of constant projections onto the principal components, that is, the set of all points in the input space that have the same coordinates when projected onto $\mathbf{u}_1$ and $\mathbf{u}_2$, respectively. For instance, the lines of constant projections in Figure 7.6a correspond to the solutions of $\mathbf{u}_1^T\mathbf{x} = s$ for different values of the coordinate s. Figure 7.7 shows the coordinates of each point in the principal components space comprising $\mathbf{u}_1$ and $\mathbf{u}_2$. It is clear from the figures that $\mathbf{u}_1$ and $\mathbf{u}_2$ do not fully capture the nonlinear relationship between X_1 and X_2. We shall see later in this section that kernel PCA is able to capture this dependence better.

Let ϕ correspond to a mapping from the input space to the feature space. Each point in feature space is given as the image $\phi(\mathbf{x}_i)$ of the point $\mathbf{x}_i$ in input space. In the input space, the first principal component captures the direction with the most projected variance; it is the eigenvector corresponding to the largest eigenvalue of the



(a) $\lambda_1 = 0.197$ (b) $\lambda_2 = 0.087$

Figure 7.6. Nonlinear Iris dataset: PCA in input space.

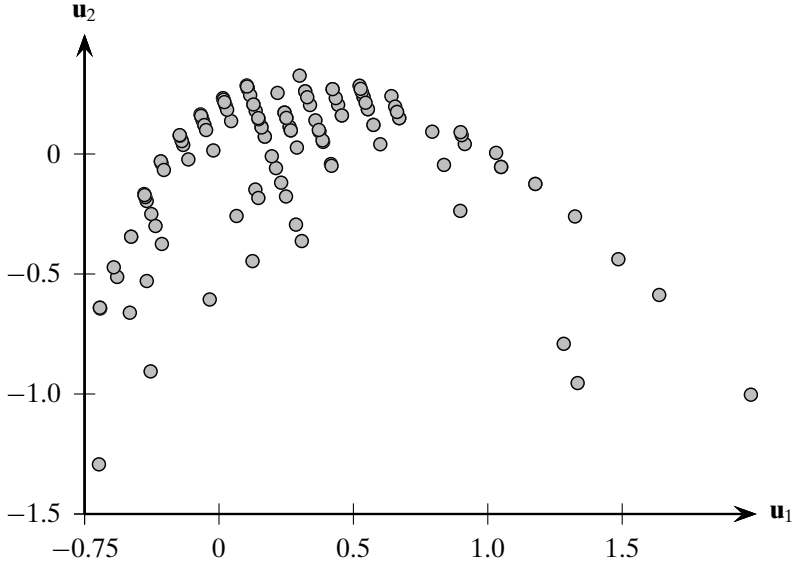


Figure 7.7. Projection onto principal components.

covariance matrix. Likewise, in feature space, we can find the first kernel principal component $\mathbf{u}_1$ (with $\mathbf{u}_1^T\mathbf{u}_1 = 1$), by solving for the eigenvector corresponding to the largest eigenvalue of the covariance matrix in feature space:

$$\boldsymbol{\Sigma}_\phi \mathbf{u}_1 = \lambda_1 \mathbf{u}_1 \tag{7.29}$$

where $\boldsymbol{\Sigma}_\phi$, the covariance matrix in feature space, is given as

$$\boldsymbol{\Sigma}_\phi = \frac{1}{n}\sum_{i=1}^{n}\phi(\mathbf{x}_i)\phi(\mathbf{x}_i)^T \tag{7.30}$$

Here we assume that the points are centered, that is, $\phi(\mathbf{x}_i) = \phi(\mathbf{x}_i) - \boldsymbol{\mu}_\phi$, where $\boldsymbol{\mu}_\phi$ is the mean in feature space.

Plugging in the expansion of $\boldsymbol{\Sigma}_\phi$ from Eq. (7.30) into Eq. (7.29), we get

$$\left(\frac{1}{n}\sum_{i=1}^{n}\phi(\mathbf{x}_i)\phi(\mathbf{x}_i)^T\right)\mathbf{u}_1 = \lambda_1\mathbf{u}_1 \tag{7.31}$$

$$\frac{1}{n}\sum_{i=1}^{n}\phi(\mathbf{x}_i)\left(\phi(\mathbf{x}_i)^T\mathbf{u}_1\right) = \lambda_1\mathbf{u}_1$$

$$\sum_{i=1}^{n}\left(\frac{\phi(\mathbf{x}_i)^T\mathbf{u}_1}{n\,\lambda_1}\right)\phi(\mathbf{x}_i) = \mathbf{u}_1$$

$$\sum_{i=1}^{n}c_i\phi(\mathbf{x}_i) = \mathbf{u}_1 \tag{7.32}$$

where $c_i = \frac{\phi(\mathbf{x}_i)^T\mathbf{u}_1}{n\lambda_1}$ is a scalar value. From Eq. (7.32) we see that the best direction in the feature space, $\mathbf{u}_1$, is just a linear combination of the transformed points, where the scalars c_i show the importance of each point toward the direction of most variance.

We can now substitute Eq. (7.32) back into Eq. (7.31) to get

$$\left(\frac{1}{n}\sum_{i=1}^{n}\phi(\mathbf{x}_i)\phi(\mathbf{x}_i)^T\right)\left(\sum_{j=1}^{n}c_j\phi(\mathbf{x}_j)\right) = \lambda_1\sum_{i=1}^{n}c_i\phi(\mathbf{x}_i)$$

$$\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n}c_j\phi(\mathbf{x}_i)\phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j) = \lambda_1\sum_{i=1}^{n}c_i\phi(\mathbf{x}_i)$$

$$\sum_{i=1}^{n}\left(\phi(\mathbf{x}_i)\sum_{j=1}^{n}c_j\phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)\right) = n\lambda_1\sum_{i=1}^{n}c_i\phi(\mathbf{x}_i)$$

In the preceding equation, we can replace the dot product in feature space, namely $\phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$, by the corresponding kernel function in input space, namely $K(\mathbf{x}_i, \mathbf{x}_j)$, which yields

$$\sum_{i=1}^{n}\left(\phi(\mathbf{x}_i)\sum_{j=1}^{n}c_jK(\mathbf{x}_i, \mathbf{x}_j)\right) = n\lambda_1\sum_{i=1}^{n}c_i\phi(\mathbf{x}_i) \tag{7.33}$$

Note that we assume that the points in feature space are centered, that is, we assume that the kernel matrix $\mathbf{K}$ has already been centered using Eq. (5.14):

$$\mathbf{K} = \left(\mathbf{I} - \frac{1}{n}\mathbf{1}_{n\times n}\right)\mathbf{K}\left(\mathbf{I} - \frac{1}{n}\mathbf{1}_{n\times n}\right)$$

where $\mathbf{I}$ is the $n \times n$ identity matrix, and $\mathbf{1}_{n\times n}$ is the $n \times n$ matrix all of whose elements are 1.

We have so far managed to replace one of the dot products with the kernel function. To make sure that all computations in feature space are only in terms of dot products, we can take any point, say $\phi(\mathbf{x}_k)$ and multiply Eq. (7.33) by $\phi(\mathbf{x}_k)^T$ on both sides to obtain

$$\sum_{i=1}^{n}\left(\phi(\mathbf{x}_k)^T\phi(\mathbf{x}_i)\sum_{j=1}^{n}c_j K(\mathbf{x}_i,\mathbf{x}_j)\right) = n\lambda_1\sum_{i=1}^{n}c_i\phi(\mathbf{x}_k)^T\phi(\mathbf{x}_i)$$

$$\sum_{i=1}^{n}\left(K(\mathbf{x}_k,\mathbf{x}_i)\sum_{j=1}^{n}c_j K(\mathbf{x}_i,\mathbf{x}_j)\right) = n\lambda_1\sum_{i=1}^{n}c_i K(\mathbf{x}_k,\mathbf{x}_i) \tag{7.34}$$

Further, let $\mathbf{K}_i$ denote row i of the centered kernel matrix, written as the column vector

$$\mathbf{K}_i = (K(\mathbf{x}_i,\mathbf{x}_1)\ \ K(\mathbf{x}_i,\mathbf{x}_2)\ \ \cdots\ \ K(\mathbf{x}_i,\mathbf{x}_n))^T$$

Let $\mathbf{c}$ denote the column vector of weights

$$\mathbf{c} = (c_1\ \ c_2\ \ \cdots\ \ c_n)^T$$

We can plug $\mathbf{K}_i$ and $\mathbf{c}$ into Eq. (7.34), and rewrite it as

$$\sum_{i=1}^{n}K(\mathbf{x}_k,\mathbf{x}_i)\mathbf{K}_i^T\mathbf{c} = n\lambda_1\mathbf{K}_k^T\mathbf{c}$$

In fact, because we can choose any of the n points, $\phi(\mathbf{x}_k)$, in the feature space, to obtain Eq. (7.34), we have a set of n equations:

$$\sum_{i=1}^{n}K(\mathbf{x}_1,\mathbf{x}_i)\mathbf{K}_i^T\mathbf{c} = n\lambda_1\mathbf{K}_1^T\mathbf{c}$$

$$\sum_{i=1}^{n}K(\mathbf{x}_2,\mathbf{x}_i)\mathbf{K}_i^T\mathbf{c} = n\lambda_1\mathbf{K}_2^T\mathbf{c}$$

$$\vdots \qquad = \qquad \vdots$$

$$\sum_{i=1}^{n}K(\mathbf{x}_n,\mathbf{x}_i)\mathbf{K}_i^T\mathbf{c} = n\lambda_1\mathbf{K}_n^T\mathbf{c}$$

We can compactly represent all of these n equations as follows:

$$\mathbf{K}^2\mathbf{c} = n\lambda_1\mathbf{K}\mathbf{c}$$

where $\mathbf{K}$ is the centered kernel matrix. Multiplying by $\mathbf{K}^{-1}$ on both sides, we obtain

$$\mathbf{K}^{-1}\mathbf{K}^2\mathbf{c} = n\lambda_1\mathbf{K}^{-1}\mathbf{K}\mathbf{c}$$

$$\mathbf{K}\mathbf{c} = n\lambda_1\mathbf{c}$$

$$\mathbf{K}\mathbf{c} = \eta_1\mathbf{c} \tag{7.35}$$

where $\eta_1 = n\lambda_1$. Thus, the weight vector $\mathbf{c}$ is the eigenvector corresponding to the largest eigenvalue η_1 of the kernel matrix $\mathbf{K}$.

Once $\mathbf{c}$ is found, we can plug it back into Eq. (7.32) to obtain the first kernel principal component $\mathbf{u}_1$. The only constraint we impose is that $\mathbf{u}_1$ should be normalized to be a unit vector, as follows:

$$\mathbf{u}_1^T\mathbf{u}_1 = 1$$

$$\sum_{i=1}^{n}\sum_{j=1}^{n} c_i c_j \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j) = 1$$

$$\mathbf{c}^T\mathbf{K}\mathbf{c} = 1$$

Noting that $\mathbf{K}\mathbf{c} = \eta_1\mathbf{c}$ from Eq. (7.35), we get

$$\mathbf{c}^T(\eta_1\mathbf{c}) = 1$$

$$\eta_1\mathbf{c}^T\mathbf{c} = 1$$

$$\|\mathbf{c}\|^2 = \frac{1}{\eta_1}$$

However, because $\mathbf{c}$ is an eigenvector of $\mathbf{K}$ it will have unit norm. Thus, to ensure that $\mathbf{u}_1$ is a unit vector, we have to scale the weight vector $\mathbf{c}$ so that its norm is $\|\mathbf{c}\| = \sqrt{\frac{1}{\eta_1}}$, which can be achieved by multiplying $\mathbf{c}$ by $\sqrt{\frac{1}{\eta_1}}$.

In general, because we do not map the input points into the feature space via ϕ, it is not possible to directly compute the principal direction, as it is specified in terms of $\phi(\mathbf{x}_i)$, as seen in Eq. (7.32). However, what matters is that we can project any point $\phi(\mathbf{x})$ onto the principal direction $\mathbf{u}_1$, as follows:

$$\mathbf{u}_1^T\phi(\mathbf{x}) = \sum_{i=1}^{n} c_i \phi(\mathbf{x}_i)^T\phi(\mathbf{x}) = \sum_{i=1}^{n} c_i K(\mathbf{x}_i, \mathbf{x})$$

which requires only kernel operations. When $\mathbf{x} = \mathbf{x}_i$ is one of the input points, the projection of $\phi(\mathbf{x}_i)$ onto the principal component $\mathbf{u}_1$ can be written as the dot product

$$\mathbf{a}_i = \mathbf{u}_1^T\phi(\mathbf{x}_i) = \mathbf{K}_i^T\mathbf{c} \tag{7.36}$$

where $\mathbf{K}_i$ is the column vector corresponding to the ith row in the kernel matrix. Thus, we have shown that all computations, either for the solution of the principal component, or for the projection of points, can be carried out using only the kernel function. Finally, we can obtain the additional principal components by solving for the other eigenvalues and eigenvectors of Eq. (7.35). In other words, if we sort the eigenvalues of $\mathbf{K}$ in decreasing order $\eta_1 \geq \eta_2 \geq \cdots \geq \eta_n \geq 0$, we can obtain the jth principal component as the corresponding eigenvector $\mathbf{c}_j$, which has to be normalized so that the norm is $\|\mathbf{c}_j\| = \sqrt{\frac{1}{\eta_j}}$, provided $\eta_j > 0$. Also, because $\eta_j = n\lambda_j$, the variance along the jth principal component is given as $\lambda_j = \frac{\eta_j}{n}$. Algorithm 7.2 gives the pseudo-code for the kernel PCA method.

ALGORITHM 7.2. Kernel Principal Component Analysis

KERNELPCA ($\mathbf{D}, K, \alpha$):

$\mathbf{K} = \left\{K(\mathbf{x}_i, \mathbf{x}_j)\right\}_{i,j=1,\ldots,n}$ // compute $n \times n$ kernel matrix

$\mathbf{K} = (\mathbf{I} - \frac{1}{n}\mathbf{1}_{n\times n})\mathbf{K}(\mathbf{I} - \frac{1}{n}\mathbf{1}_{n\times n})$ // center the kernel matrix

$(\eta_1, \eta_2, \ldots, \eta_d) = \text{eigenvalues}(\mathbf{K})$ // compute eigenvalues

$\begin{pmatrix}\mathbf{c}_1 & \mathbf{c}_2 & \cdots & \mathbf{c}_n\end{pmatrix} = \text{eigenvectors}(\mathbf{K})$ // compute eigenvectors

$\lambda_i = \frac{\eta_i}{n}$ for all $i = 1, \ldots, n$ // compute variance for each component

$\mathbf{c}_i = \sqrt{\frac{1}{\eta_i}} \cdot \mathbf{c}_i$ for all $i = 1, \ldots, n$ // ensure that $\mathbf{u}_i^T\mathbf{u}_i = 1$

$f(r) = \frac{\sum_{i=1}^{r}\lambda_i}{\sum_{i=1}^{d}\lambda_i}$, for all $r = 1, 2, \ldots, d$ // fraction of total variance

Choose smallest r so that $f(r) \geq \alpha$ // choose dimensionality

$\mathbf{C}_r = \begin{pmatrix}\mathbf{c}_1 & \mathbf{c}_2 & \cdots & \mathbf{c}_r\end{pmatrix}$ // reduced basis

$\mathbf{A} = \{\mathbf{a}_i \mid \mathbf{a}_i = \mathbf{C}_r^T\mathbf{K}_i, \text{for } i = 1, \ldots, n\}$ // reduced dimensionality data

Example 7.8. Consider the nonlinear Iris data from Example 7.7 with $n = 150$ points. Let us use the homogeneous quadratic polynomial kernel in Eq. (5.8):

$$K(\mathbf{x}_i, \mathbf{x}_j) = \left(\mathbf{x}_i^T\mathbf{x}_j\right)^2$$

The kernel matrix $\mathbf{K}$ has three nonzero eigenvalues:

$$\eta_1 = 31.0 \qquad \eta_2 = 8.94 \qquad \eta_3 = 2.76$$

$$\lambda_1 = \frac{\eta_1}{150} = 0.2067 \qquad \lambda_2 = \frac{\eta_2}{150} = 0.0596 \qquad \lambda_3 = \frac{\eta_3}{150} = 0.0184$$

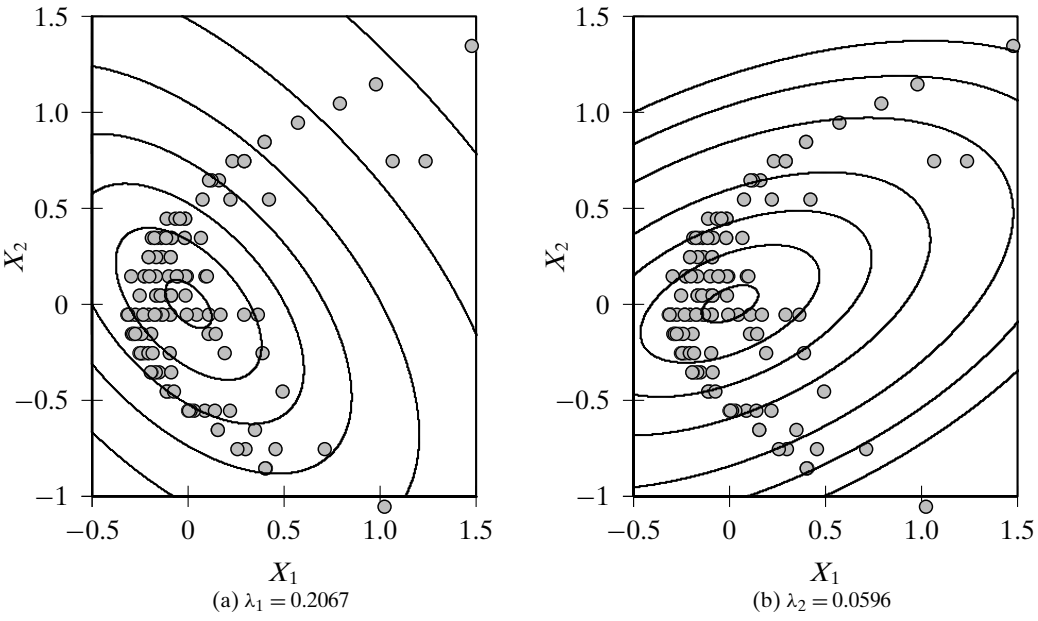
The corresponding eigenvectors $\mathbf{c}_1$, $\mathbf{c}_2$, and $\mathbf{c}_3$ are not shown because they lie in $\mathbb{R}^{150}$.

Figure 7.8 shows the contour lines of constant projection onto the first three kernel principal components. These lines are obtained by solving the equations $\mathbf{u}_i^T\mathbf{x} = \sum_{j=1}^{n} c_{ij}K(\mathbf{x}_j, \mathbf{x}) = s$ for different projection values s, for each of the eigenvectors $\mathbf{c}_i = (c_{i1}, c_{i2}, \ldots, c_{in})^T$ of the kernel matrix. For instance, for the first principal component this corresponds to the solutions $\mathbf{x} = (x_1, x_2)^T$, shown as contour lines, of the following equation:

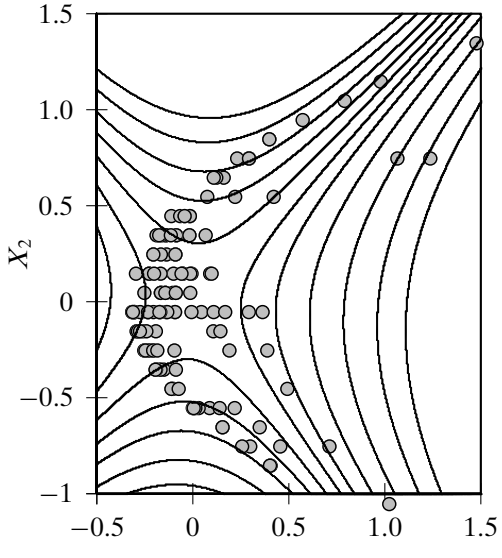
$$1.0426x_1^2 + 0.995x_2^2 + 0.914x_1x_2 = s$$

for each chosen value of s. The principal components are also not shown in the figure, as it is typically not possible or feasible to map the points into feature space, and thus one cannot derive an explicit expression for $\mathbf{u}_i$. However, because the projection onto the principal components can be carried out via kernel operations via Eq. (7.36), Figure 7.9 shows the projection of the points onto the first two kernel principal components, which capture $\frac{\lambda_1+\lambda_2}{\lambda_1+\lambda_2+\lambda_3} = \frac{0.2663}{0.2847} = 93.5\%$ of the total variance.

Incidentally, the use of a linear kernel $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T\mathbf{x}_j$ yields exactly the same principal components as shown in Figure 7.7.



(a) $\lambda_1 = 0.2067$ (b) $\lambda_2 = 0.0596$



(c) $\lambda_3 = 0.0184$

Figure 7.8. Kernel PCA: homogeneous quadratic kernel.

7.4 SINGULAR VALUE DECOMPOSITION

Principal components analysis is a special case of a more general matrix decomposition method called *Singular Value Decomposition (SVD)*. We saw in Eq. (7.28) that PCA yields the following decomposition of the covariance matrix:

$$\mathbf{\Sigma} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T \tag{7.37}$$

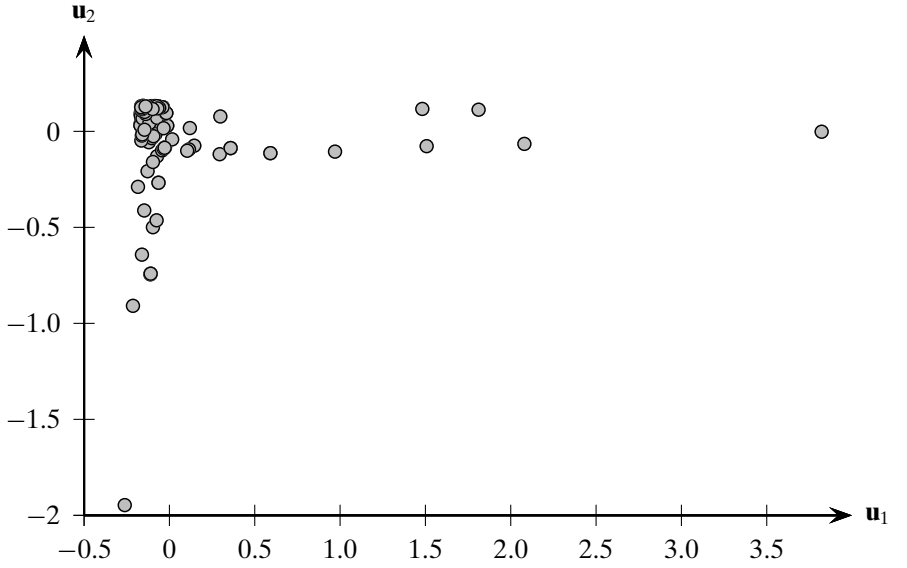


Figure 7.9. Projected point coordinates: homogeneous quadratic kernel.

where the covariance matrix has been factorized into the orthogonal matrix $\mathbf{U}$ containing its eigenvectors, and a diagonal matrix $\mathbf{\Lambda}$ containing its eigenvalues (sorted in decreasing order). SVD generalizes the above factorization for any matrix. In particular for an $n \times d$ data matrix $\mathbf{D}$ with n points and d columns, SVD factorizes $\mathbf{D}$ as follows:

$$\mathbf{D} = \mathbf{L}\mathbf{\Delta}\mathbf{R}^T \tag{7.38}$$

where $\mathbf{L}$ is a orthogonal $n \times n$ matrix, $\mathbf{R}$ is an orthogonal $d \times d$ matrix, and $\mathbf{\Delta}$ is an $n \times d$ "diagonal" matrix. The columns of $\mathbf{L}$ are called the *left singular vectors*, and the columns of $\mathbf{R}$ (or rows of $\mathbf{R}^T$) are called the *right singular vectors*. The matrix $\mathbf{\Delta}$ is defined as

$$\mathbf{\Delta}(i,j) = \begin{cases} \delta_i & \text{If } i = j \\ 0 & \text{If } i \neq j \end{cases}$$

where $i = 1, \ldots, n$ and $j = 1, \ldots, d$. The entries $\mathbf{\Delta}(i,i) = \delta_i$ along the main diagonal of $\mathbf{\Delta}$ are called the *singular values* of $\mathbf{D}$, and they are all non-negative. If the rank of $\mathbf{D}$ is $r \leq \min(n,d)$, then there will be only r nonzero singular values, which we assume are ordered as follows:

$$\delta_1 \geq \delta_2 \geq \cdots \geq \delta_r > 0$$

One can discard those left and right singular vectors that correspond to zero singular values, to obtain the *reduced SVD* as

$$\mathbf{D} = \mathbf{L}_r \mathbf{\Delta}_r \mathbf{R}_r^T \tag{7.39}$$

where $\mathbf{L}_r$ is the $n \times r$ matrix of the left singular vectors, $\mathbf{R}_r$ is the $d \times r$ matrix of the right singular vectors, and $\mathbf{\Delta}_r$ is the $r \times r$ diagonal matrix containing the positive singular vectors. The reduced SVD leads directly to the *spectral decomposition* of $\mathbf{D}$, given as

$$\begin{aligned}
\mathbf{D} &= \mathbf{L}_r \mathbf{\Delta}_r \mathbf{R}_r^T \\
&= \begin{pmatrix} | & | & & | \\ \boldsymbol{l}_1 & \boldsymbol{l}_2 & \cdots & \boldsymbol{l}_r \\ | & | & & | \end{pmatrix} \begin{pmatrix} \delta_1 & 0 & \cdots & 0 \\ 0 & \delta_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \delta_r \end{pmatrix} \begin{pmatrix} — & \mathbf{r}_1^T & — \\ — & \mathbf{r}_2^T & — \\ & \vdots & \\ — & \mathbf{r}_r^T & — \end{pmatrix} \\
&= \delta_1 \boldsymbol{l}_1 \mathbf{r}_1^T + \delta_2 \boldsymbol{l}_2 \mathbf{r}_2^T + \cdots + \delta_r \boldsymbol{l}_r \mathbf{r}_r^T \\
&= \sum_{i=1}^{r} \delta_i \boldsymbol{l}_i \mathbf{r}_i^T
\end{aligned}$$

The spectral decomposition represents $\mathbf{D}$ as a sum of rank one matrices of the form $\delta_i \boldsymbol{l}_i \mathbf{r}_i^T$. By selecting the q largest singular values $\delta_1, \delta_2, \ldots, \delta_q$ and the corresponding left and right singular vectors, we obtain the best rank q approximation to the original matrix $\mathbf{D}$. That is, if $\mathbf{D}_q$ is the matrix defined as

$$\mathbf{D}_q = \sum_{i=1}^{q} \delta_i \boldsymbol{l}_i \mathbf{r}_i^T$$

then it can be shown that $\mathbf{D}_q$ is the rank q matrix that minimizes the expression

$$\|\mathbf{D} - \mathbf{D}_q\|_F$$

where $\|\mathbf{A}\|_F$ is called the *Frobenius Norm* of the $n \times d$ matrix $\mathbf{A}$, defined as

$$\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{d} \mathbf{A}(i,j)^2}$$

7.4.1 Geometry of SVD

In general, any $n \times d$ matrix $\mathbf{D}$ represents a *linear transformation*, $\mathbf{D}: \mathbb{R}^d \rightarrow \mathbb{R}^n$, from the space of d-dimensional vectors to the space of n-dimensional vectors because for any $\mathbf{x} \in \mathbb{R}^d$ there exists $\mathbf{y} \in \mathbb{R}^n$ such that

$$\mathbf{D}\mathbf{x} = \mathbf{y}$$

The set of all vectors $\mathbf{y} \in \mathbb{R}^n$ such that $\mathbf{D}\mathbf{x} = \mathbf{y}$ over all possible $\mathbf{x} \in \mathbb{R}^d$ is called the *column space* of $\mathbf{D}$, and the set of all vectors $\mathbf{x} \in \mathbb{R}^d$, such that $\mathbf{D}^T\mathbf{y} = \mathbf{x}$ over all $\mathbf{y} \in \mathbb{R}^n$, is called the *row space* of $\mathbf{D}$, which is equivalent to the column space of $\mathbf{D}^T$. In other words, the column space of $\mathbf{D}$ is the set of all vectors that can be obtained as linear combinations of columns of $\mathbf{D}$, and the row space of $\mathbf{D}$ is the set of all vectors that can

be obtained as linear combinations of the rows of $\mathbf{D}$ (or columns of $\mathbf{D}^T$). Also note that the set of all vectors $\mathbf{x} \in \mathbb{R}^d$, such that $\mathbf{Dx} = \mathbf{0}$ is called the *null space* of $\mathbf{D}$, and finally, the set of all vectors $\mathbf{y} \in \mathbb{R}^n$, such that $\mathbf{D}^T\mathbf{y} = \mathbf{0}$ is called the *left null space* of $\mathbf{D}$.

One of the main properties of SVD is that it gives a basis for each of the four fundamental spaces associated with the matrix $\mathbf{D}$. If $\mathbf{D}$ has rank r, it means that it has only r independent columns, and also only r independent rows. Thus, the r left singular vectors $\boldsymbol{l}_1, \boldsymbol{l}_2, \ldots, \boldsymbol{l}_r$ corresponding to the r nonzero singular values of $\mathbf{D}$ in Eq. (7.38) represent a basis for the column space of $\mathbf{D}$. The remaining $n - r$ left singular vectors $\boldsymbol{l}_{r+1}, \ldots, \boldsymbol{l}_n$ represent a basis for the left null space of $\mathbf{D}$. For the row space, the r right singular vectors $\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_r$ corresponding to the r non-zero singular values, represent a basis for the row space of $\mathbf{D}$, and the remaining $d - r$ right singular vectors $\mathbf{r}_j$ $(j = r+1, \ldots, d)$, represent a basis for the null space of $\mathbf{D}$.

Consider the reduced SVD expression in Eq. (7.39). Right multiplying both sides of the equation by $\mathbf{R}_r$ and noting that $\mathbf{R}_r^T\mathbf{R}_r = \mathbf{I}_r$, where $\mathbf{I}_r$ is the $r \times r$ identity matrix, we have

$$\mathbf{DR}_r = \mathbf{L}_r \boldsymbol{\Delta}_r \mathbf{R}_r^T \mathbf{R}_r$$

$$\mathbf{DR}_r = \mathbf{L}_r \boldsymbol{\Delta}_r$$

$$\mathbf{DR}_r = \mathbf{L}_r \begin{pmatrix} \delta_1 & 0 & \cdots & 0 \\ 0 & \delta_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \delta_r \end{pmatrix}$$

$$\mathbf{D} \begin{pmatrix} | & | & & | \\ \mathbf{r}_1 & \mathbf{r}_2 & \cdots & \mathbf{r}_r \\ | & | & & | \end{pmatrix} = \begin{pmatrix} | & | & & | \\ \delta_1 \boldsymbol{l}_1 & \delta_2 \boldsymbol{l}_2 & \cdots & \delta_r \boldsymbol{l}_r \\ | & | & & | \end{pmatrix}$$

From the above, we conclude that

$$\mathbf{Dr}_i = \delta_i \boldsymbol{l}_i \quad \text{for all } i = 1, \ldots, r$$

In other words, SVD is a special factorization of the matrix $\mathbf{D}$, such that any basis vector $\mathbf{r}_i$ for the row space is mapped to the corresponding basis vector $\boldsymbol{l}_i$ in the column space, scaled by the singular value δ_i. As such, we can think of the SVD as a mapping from an orthonormal basis $(\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_r)$ in $\mathbb{R}^d$ (the row space) to an orthonormal basis $(\boldsymbol{l}_1, \boldsymbol{l}_2, \ldots, \boldsymbol{l}_r)$ in $\mathbb{R}^n$ (the column space), with the corresponding axes scaled according to the singular values $\delta_1, \delta_2, \ldots, \delta_r$.

7.4.2 Connection between SVD and PCA

Assume that the matrix $\mathbf{D}$ has been centered, and assume that it has been factorized via SVD [Eq. (7.38)] as $\mathbf{D} = \mathbf{L}\boldsymbol{\Delta}\mathbf{R}^T$. Consider the *scatter matrix* for $\mathbf{D}$, given as $\mathbf{D}^T\mathbf{D}$. We have

$$\begin{aligned} \mathbf{D}^T\mathbf{D} &= \left(\mathbf{L}\boldsymbol{\Delta}\mathbf{R}^T\right)^T \left(\mathbf{L}\boldsymbol{\Delta}\mathbf{R}^T\right) \\ &= \mathbf{R}\boldsymbol{\Delta}^T\mathbf{L}^T\mathbf{L}\boldsymbol{\Delta}\mathbf{R}^T \end{aligned}$$

$$= \mathbf{R}(\boldsymbol{\Delta}^T\boldsymbol{\Delta})\mathbf{R}^T$$
$$= \mathbf{R}\boldsymbol{\Delta}_d^2\mathbf{R}^T \tag{7.40}$$

where $\boldsymbol{\Delta}_d^2$ is the $d \times d$ diagonal matrix defined as $\boldsymbol{\Delta}_d^2(i,i) = \delta_i^2$, for $i = 1, \ldots, d$. Only $r \leq \min(d, n)$ of these eigenvalues are positive, whereas the rest are all zeros.

Because the covariance matrix of centered $\mathbf{D}$ is given as $\boldsymbol{\Sigma} = \frac{1}{n}\mathbf{D}^T\mathbf{D}$, and because it can be decomposed as $\boldsymbol{\Sigma} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T$ via PCA [Eq. (7.37)], we have

$$\mathbf{D}^T\mathbf{D} = n\boldsymbol{\Sigma}$$
$$= n\mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T$$
$$= \mathbf{U}(n\boldsymbol{\Lambda})\mathbf{U}^T \tag{7.41}$$

Equating Eq. (7.40) and Eq. (7.41), we conclude that the right singular vectors $\mathbf{R}$ are the same as the eigenvectors of $\boldsymbol{\Sigma}$. Further, the corresponding singular values of $\mathbf{D}$ are related to the eigenvalues of $\boldsymbol{\Sigma}$ by the expression

$$n\lambda_i = \delta_i^2$$
$$\text{or, } \lambda_i = \frac{\delta_i^2}{n}, \text{ for } i = 1, \ldots, d \tag{7.42}$$

Let us now consider the matrix $\mathbf{D}\mathbf{D}^T$. We have

$$\mathbf{D}\mathbf{D}^T = (\mathbf{L}\boldsymbol{\Delta}\mathbf{R}^T)(\mathbf{L}\boldsymbol{\Delta}\mathbf{R}^T)^T$$
$$= \mathbf{L}\boldsymbol{\Delta}\mathbf{R}^T\mathbf{R}\boldsymbol{\Delta}^T\mathbf{L}^T$$
$$= \mathbf{L}(\boldsymbol{\Delta}\boldsymbol{\Delta}^T)\mathbf{L}^T$$
$$= \mathbf{L}\boldsymbol{\Delta}_n^2\mathbf{L}^T$$

where $\boldsymbol{\Delta}_n^2$ is the $n \times n$ diagonal matrix given as $\boldsymbol{\Delta}_n^2(i,i) = \delta_i^2$, for $i = 1, \ldots, n$. Only r of these singular values are positive, whereas the rest are all zeros. Thus, the left singular vectors in $\mathbf{L}$ are the eigenvectors of the matrix $n \times n$ matrix $\mathbf{D}\mathbf{D}^T$, and the corresponding eigenvalues are given as δ_i^2.

Example 7.9. Let us consider the $n \times d$ centered Iris data matrix $\mathbf{D}$ from Example 7.1, with $n = 150$ and $d = 3$. In Example 7.5 we computed the eigenvectors and eigenvalues of the covariance matrix $\boldsymbol{\Sigma}$ as follows:

$$\lambda_1 = 3.662 \qquad \lambda_2 = 0.239 \qquad \lambda_3 = 0.059$$
$$\mathbf{u}_1 = \begin{pmatrix} -0.390 \\ 0.089 \\ -0.916 \end{pmatrix} \qquad \mathbf{u}_2 = \begin{pmatrix} -0.639 \\ -0.742 \\ 0.200 \end{pmatrix} \qquad \mathbf{u}_3 = \begin{pmatrix} -0.663 \\ 0.664 \\ 0.346 \end{pmatrix}$$

Computing the SVD of $\mathbf{D}$ yields the following nonzero singular values and the corresponding right singular vectors

$$\delta_1 = 23.437 \qquad \delta_2 = 5.992 \qquad \delta_3 = 2.974$$

$$\mathbf{r}_1 = \begin{pmatrix} -0.390 \\ 0.089 \\ -0.916 \end{pmatrix} \qquad \mathbf{r}_2 = \begin{pmatrix} 0.639 \\ 0.742 \\ -0.200 \end{pmatrix} \qquad \mathbf{r}_3 = \begin{pmatrix} -0.663 \\ 0.664 \\ 0.346 \end{pmatrix}$$

We do not show the left singular vectors $\boldsymbol{l}_1, \boldsymbol{l}_2, \boldsymbol{l}_3$ because they lie in $\mathbb{R}^{150}$. Using Eq. (7.42) one can verify that $\lambda_i = \frac{\delta_i^2}{n}$. For example,

$$\lambda_1 = \frac{\delta_1^2}{n} = \frac{23.437^2}{150} = \frac{549.29}{150} = 3.662$$

Notice also that the right singular vectors are equivalent to the principal components or eigenvectors of $\boldsymbol{\Sigma}$, up to isomorphism. That is, they may potentially be reversed in direction. For the Iris dataset, we have $\mathbf{r}_1 = \mathbf{u}_1$, $\mathbf{r}_2 = -\mathbf{u}_2$, and $\mathbf{r}_3 = \mathbf{u}_3$. Here the second right singular vector is reversed in sign when compared to the second principal component.

7.5 FURTHER READING

Principal component analysis was pioneered in Pearson (1901). For a comprehensive description of PCA see Jolliffe (2002). Kernel PCA was first introduced in Schölkopf, Smola, and Müller (1998). For further exploration of non-linear dimensionality reduction methods see Lee and Verleysen (2007). The requisite linear algebra background can be found in Strang (2006).

Jolliffe, I. (2002). *Principal Component Analysis,* 2nd ed. Springer Series in Statistics. New York: Springer Science + Business Media.

Lee, J. A. and Verleysen, M. (2007). *Nonlinear Dimensionality Reduction*. New York: Springer Science + Business Media.

Pearson, K. (1901). "On lines and planes of closest fit to systems of points in space." *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2 (11): 559–572.

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

Strang, G. (2006). *Linear Algebra and Its Applications*, 4th ed. Independence, KY: Thomson Brooks/Cole, Cengage Learning.