11

Principal Component Analysis And Low-Rank Models

Overview

In this chapter, we consider the problem of analyzing high-dimensional datasets, where each data point consists of multiple features. Such data are naturally modeled as vectors or matrices. Section 11.1 defines the mean of random vectors and matrices. Section 11.2 introduces the covariance matrix, which encodes the variance of any linear combination of the entries of a random vector. In Section 11.3 we explain how to estimate the covariance matrix from data. Section 11.4 describes principal component analysis (PCA), a popular method to extract the directions of maximum variance in a dataset. Section 11.5 discusses how to use PCA to find optimal low-dimensional representations of high-dimensional data. In Section 11.6 we introduce low-rank models for matrix-valued data and explain how to fit them using the singular-value decomposition. Finally, in Section 11.7 we show how to use low-rank models to estimate missing entries in a matrix, illustrating the approach with an application to collaborative filtering for movie-rating prediction.

11.1 The Mean In Multiple Dimensions

Consider a d-dimensional random vector

$$\tilde{x} := \begin{bmatrix} \tilde{x}[1] \\ \tilde{x}[2] \\ \cdots \\ \tilde{x}[d] \end{bmatrix}, \tag{11.1}$$

which models a dataset with d features. We define the mean of $\tilde{x}$ as the vector formed by the means of its entries.

Definition 11.1 (Mean of a random vector)**.** *The mean of a d-dimensional random vector $\tilde{x}$ is the vector of entry-wise means:*

$$\mathrm{E}\left[\tilde{x}\right] := \begin{bmatrix} \mathrm{E}\left[\tilde{x}[1]\right] \\ \mathrm{E}\left[\tilde{x}[2]\right] \\ \cdots \\ \mathrm{E}\left[\tilde{x}[d]\right] \end{bmatrix}. \tag{11.2}$$

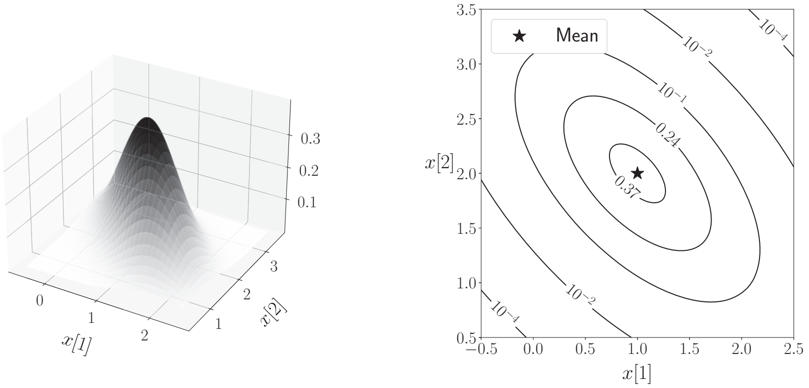


Figure 11.1 Mean of a Gaussian random vector. Joint pdf of a two-dimensional Gaussian random vector (left) and two-dimensional plot of its contour lines (right). The mean of a random vector (indicated by the star marker) is at the center of the contour lines.

As established in the following theorem, the mean is the point in d-dimensional space that minimizes the average squared Euclidean distance to the random vector. It can therefore be interpreted as the center of the distribution of the random vector (with the caveat that it may be distorted by extreme values, as discussed in Section 7.5). We often refer to the operation of subtracting the mean as *centering*.

Theorem 11.2. *For any d-dimensional random vector $\tilde{x}$ with finite mean,*

$$\mathrm{E}\left[\tilde{x}\right] = \arg\min_{a \in \mathbb{R}^d} \mathrm{E}\left[\left|\left|\tilde{x} - a\right|\right|_2^2\right]. \tag{11.3}$$

Proof By linearity of expectation, the mean squared ℓ_2 distance between $\tilde{x}$ and a can be decomposed into d terms equal to the mean squared difference between $\tilde{x}[i]$ and $a[i]$ for $1 \leq i \leq d$,

$$\mathrm{E}\left[\left|\left|\tilde{x} - a\right|\right|_2^2\right] = \mathrm{E}\left[\sum_{i=1}^{d} \left(\tilde{x}[i] - a[i]\right)^2\right] \tag{11.4}$$

$$= \sum_{i=1}^{d} \mathrm{E}\left[\left(\tilde{x}[i] - a[i]\right)^2\right]. \tag{11.5}$$

The result then follows from Theorem 7.30 and Definition 11.1. ■

The mean of a Gaussian random vector equals its mean parameter. It is at the center of the contour surfaces of its joint pdf, as illustrated in Figure 11.1.

Lemma 11.3 (Mean of a Gaussian random vector)**.** *The mean of a d-dimensional Gaussian random vector $\tilde{x}$ is equal to its mean parameter.*

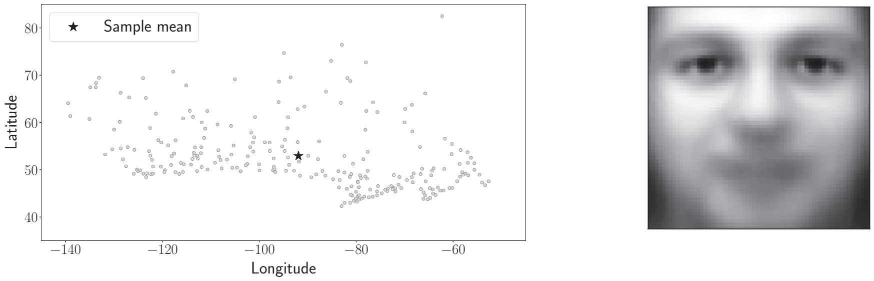


Figure 11.2 Sample mean of a dataset. The left plot shows a scatterplot of the latitude and longitude of the main 248 cities in Canada, represented by two-dimensional vectors as explained in Example 11.5. The star indicates the sample mean of the dataset. The right image is the sample mean of the faces dataset in Example 11.6.

Proof Let $\mu \in \mathbb{R}^d$ denote the mean parameter of $\tilde{x}$. For any $1 \leq i \leq d$, by Theorem 5.25 the ith entry $\tilde{x}[i]$ is a Gaussian random variable with mean parameter $\mu[i]$. Consequently, the mean of $\tilde{x}[i]$ is $\mu[i]$ by Lemma 7.25. ■

To estimate the mean of a random vector, we compute its sample mean, which consists of the sample mean of each entry.

Definition 11.4 (Sample mean of multidimensional data)**.** *Let* $X := \{x_1, x_2, \ldots, x_n\}$ *denote a d-dimensional real-valued dataset. The sample mean is the entry-wise average*

$$m(X) := \frac{1}{n}\sum_{i=1}^{n} x_i. \tag{11.6}$$

Example 11.5 (Canadian cities: Sample mean)**.** Dataset 19 contains the locations (latitude and longitude) of the cities in Canada with more than 1,000 inhabitants. The left plot in Figure 11.2 shows a scatterplot of the data. The sample mean is

$$m(X) = \begin{bmatrix} -91.9 \\ 52.9 \end{bmatrix}, \tag{11.7}$$

where the first entry is the sample mean of the longitude and the second entry is the sample mean of the latitude.

. . . .

Example 11.6 (Faces: Sample mean)**.** The Olivetti Faces dataset (Dataset 20) contains 400 64 × 64 images of 40 different subjects (10 per subject). We interpret each image as a 4,096-dimensional vector, where each entry corresponds to a pixel. Figure 11.2 shows the sample mean of the dataset (on the right). This is the *average face* in the dataset.

. . . .

As we show in Section 11.2, manipulating random vectors often produces *random matrices* with entries that are random variables. For example, $\tilde{x}\tilde{x}^T$ is a $d \times d$ random matrix. We define the mean of these objects as the matrix of entry-wise means.

Definition 11.7 (Mean of a random matrix). *Let $\widetilde{M}$ be a $d_1 \times d_2$ random matrix, where each entry $\widetilde{M}[i,j]$ is a random variable for $1 \leq i \leq d_1$ and $1 \leq j \leq d_2$. The mean of $\widetilde{M}$ is the deterministic matrix*

$$\mathrm{E}[\widetilde{M}] := \begin{bmatrix} \mathrm{E}\left[\widetilde{M}[1,1]\right] & \mathrm{E}\left[\widetilde{M}[1,2]\right] & \cdots & \mathrm{E}\left[\widetilde{M}[1,d_2]\right] \\ \mathrm{E}\left[\widetilde{M}[2,1]\right] & \mathrm{E}\left[\widetilde{M}[2,2]\right] & \cdots & \mathrm{E}\left[\widetilde{M}[2,d_2]\right] \\ & & \cdots & \\ \mathrm{E}\left[\widetilde{M}[d_1,1]\right] & \mathrm{E}\left[\widetilde{M}[d_1,2]\right] & \cdots & \mathrm{E}\left[\widetilde{M}[d_1,d_2]\right] \end{bmatrix}. \tag{11.8}$$

Linearity of expectation holds for random vectors and random matrices. This follows directly from linearity of expectation for scalars (Theorem 7.17).

Lemma 11.8 (Linearity of expectation for random vectors and matrices). *Let $\tilde{x}$ a d-dimensional random vector, and let $b \in \mathbb{R}^k$ and $A \in \mathbb{R}^{k \times d}$ for some positive integer k, then*

$$\mathrm{E}\left[A\tilde{x} + b\right] = A\mathrm{E}\left[\tilde{x}\right] + b. \tag{11.9}$$

Similarly let, $\widetilde{M}$ be a $d_1 \times d_2$ random matrix, and let $B \in \mathbb{R}^{k \times d_2}$ and $A \in \mathbb{R}^{k \times d_1}$ for some positive integer k, then

$$\mathrm{E}[A\widetilde{M} + B] = A\mathrm{E}[\widetilde{M}] + B. \tag{11.10}$$

Proof We prove the result for vectors, the proof for matrices is the same. By linearity of expectation for scalars, the ith entry of $\mathrm{E}\left[A\tilde{x} + b\right]$ equals

$$\begin{aligned} \mathrm{E}\left[A\tilde{x} + b\right][i] &= \mathrm{E}\left[(A\tilde{x} + b)[i]\right] && (11.11) \\ &= \mathrm{E}\left[\sum_{j=1}^{d} A[i,j]\tilde{x}[j] + b[i]\right] && (11.12) \\ &= \sum_{j=1}^{d} A[i,j]\mathrm{E}\left[\tilde{x}[j]\right] + b[i] && (11.13) \\ &= (A\mathrm{E}\left[\tilde{x}\right] + b)[i]. && (11.14) \end{aligned}$$

■

11.2 The Covariance Matrix

As explained in Section 7.7, the variance of a random variable quantifies how much it varies on average around its mean. In order to characterize the fluctuations of a random vector $\tilde{x}$, we can compute the variance of linear combinations of its entries. Any such linear combination can be expressed as a random variable

$$\sum_{i=1}^{d} a[i]\tilde{x}[i] = a^T \tilde{x} \tag{11.15}$$

for some deterministic vector $a \in \mathbb{R}^d$. By linearity of expectation, the variance of $a^T\tilde{x}$ equals

$$\begin{aligned}
\operatorname{Var}\left[a^T\tilde{x}\right] &= \operatorname{E}\left[\left(a^T\tilde{x} - \operatorname{E}\left[a^T\tilde{x}\right]\right)^2\right] && (11.16)\\
&= \operatorname{E}\left[(a^T \operatorname{ct}(\tilde{x}))^2\right] && (11.17)\\
&= a^T \operatorname{E}\left[\operatorname{ct}(\tilde{x})\operatorname{ct}(\tilde{x})^T\right] a, && (11.18)
\end{aligned}$$

where $\operatorname{ct}(\tilde{x}) := \tilde{x} - \operatorname{E}[\tilde{x}]$ is the result of centering $\tilde{x}$ by subtracting its mean. The matrix $\operatorname{E}[\operatorname{ct}(\tilde{x})\operatorname{ct}(\tilde{x})^T]$ in (11.18) does not depend on the vector a; it is always the same. We call this matrix the *covariance matrix* of the vector. The covariance matrix encodes the variance of any linear combination of the entries of the random vector. Its diagonal entries equal the variance of the corresponding entry of $\tilde{x}$,

$$\operatorname{E}\left[\left(\operatorname{ct}(\tilde{x})\operatorname{ct}(\tilde{x})^T\right)[i,i]\right] = \operatorname{E}\left[\operatorname{ct}(\tilde{x}[i])^2\right] = \operatorname{Var}[\tilde{x}[i]], \qquad 1 \le i \le d. \tag{11.19}$$

Its off-diagonal entries are equal to the covariance between the corresponding entries of $\tilde{x}$,

$$\operatorname{E}\left[\left(\operatorname{ct}(\tilde{x})\operatorname{ct}(\tilde{x})^T\right)[i,j]\right] = \operatorname{E}\left[\operatorname{ct}(\tilde{x}[i])\operatorname{ct}(\tilde{x}[j])\right] = \operatorname{Cov}[\tilde{x}[i],\tilde{x}[j]], \quad 1 \le i,j \le d.$$

Definition 11.9 (Covariance matrix)**.** *The covariance matrix of a d-dimensional random vector $\tilde{x}$ is the $d \times d$ matrix*

$$\begin{aligned}
\Sigma_{\tilde{x}} &:= \operatorname{E}\left[\operatorname{ct}(\tilde{x})\operatorname{ct}(\tilde{x})^T\right] && (11.20)\\
&= \begin{bmatrix}
\operatorname{Var}[\tilde{x}[1]] & \operatorname{Cov}[\tilde{x}[1],\tilde{x}[2]] & \cdots & \operatorname{Cov}[\tilde{x}[1],\tilde{x}[d]]\\
\operatorname{Cov}[\tilde{x}[1],\tilde{x}[2]] & \operatorname{Var}[\tilde{x}[2]] & \cdots & \operatorname{Cov}[\tilde{x}[2],\tilde{x}[d]]\\
\vdots & \vdots & \ddots & \vdots\\
\operatorname{Cov}[\tilde{x}[1],\tilde{x}[d]] & \operatorname{Cov}[\tilde{x}[2],\tilde{x}[d]] & \cdots & \operatorname{Var}[\tilde{x}[d]]
\end{bmatrix}, && (11.21)
\end{aligned}$$

where $\operatorname{ct}(\tilde{x}) := \tilde{x} - \operatorname{E}[\tilde{x}]$.

The covariance-matrix parameter of a Gaussian random vector is, unsurprisingly, equal to its covariance matrix.

Theorem 11.10 (Covariance matrix of a Gaussian random vector)**.** *The covariance matrix of a Gaussian random vector is equal to its covariance-matrix parameter.*

Proof Let $\Sigma \in \mathbb{R}^{d \times d}$ denote the covariance-matrix parameter of $\tilde{x}$, and $\Sigma_{\tilde{x}} \in \mathbb{R}^{d \times d}$ its actual covariance matrix. For any $1 \leq i \leq d$, by Theorem 5.25 the ith entry $\tilde{x}[i]$ is a Gaussian random variable with variance parameter $\Sigma[i, i]$. By Lemma 7.43 this implies that each diagonal entry $\Sigma[i, i]$ is equal to the variance of $\tilde{x}[i]$, and therefore to the corresponding diagonal entry $\Sigma_{\tilde{x}}[i, i]$ of the covariance matrix.

For any $1 \leq i, j \leq d$, by Theorem 5.25 the joint pdf of the subvector

$$\begin{bmatrix} \tilde{x}[i] \\ \tilde{x}[j] \end{bmatrix} \tag{11.22}$$

is Gaussian with covariance-matrix parameter

$$\begin{bmatrix} \Sigma[i, i] & \Sigma[i, j] \\ \Sigma[j, i] & \Sigma[j, j] \end{bmatrix}. \tag{11.23}$$

In the proof of Theorem 8.16 (see also Example 8.8), we show that the correlation coefficient of $\tilde{x}[i]$ and $\tilde{x}[j]$ is equal to $\Sigma[i, j]$ divided by the standard deviations of $\tilde{x}[i]$ and $\tilde{x}[j]$. Therefore, $\Sigma[i, j]$ is equal to the covariance of $\tilde{x}[i]$ and $\tilde{x}[j]$, and consequently to $\Sigma_{\tilde{x}}[i, j]$. ■

As explained at the beginning of this section, the covariance matrix enables us to compute the variance of any linear combination of the entries of a random vector.

Theorem 11.11 (Variance of a linear combination)**.** *For any random vector $\tilde{x}$ with covariance matrix $\Sigma_{\tilde{x}}$, and any deterministic vector $a \in \mathbb{R}^d$,*

$$\text{Var}\left[a^T \tilde{x}\right] = a^T \Sigma_{\tilde{x}} a. \tag{11.24}$$

Proof The result follows from the derivation in (11.18). ■

Example 11.12 (Cheese sandwich)**.** A deli in New York is worried about the fluctuations in the cost of their signature cheese sandwich. The ingredients of the sandwich are bread, a local cheese, and an imported cheese. They model the price of each ingredient as an entry in a three dimensional random vector $\tilde{x}$. The entries $\tilde{x}[1]$, $\tilde{x}[2]$, and $\tilde{x}[3]$ represent the price (in cents per gram) of the bread, the local cheese and the imported cheese, respectively. From past data, the covariance matrix of $\tilde{x}$ is estimated to equal

$$\Sigma_{\tilde{x}} = \begin{bmatrix} 1 & 0.8 & 0 \\ 0.8 & 1 & 0 \\ 0 & 0 & 1.2 \end{bmatrix}. \tag{11.25}$$

The deli considers two recipes. Recipe 1 uses 100g of bread, 50g of local cheese, and 50g of imported cheese. Recipe 2 uses 100g of bread, 100g of local cheese, and no imported cheese. By Theorem 11.11 the standard deviation in the price of Recipe 1 is

$$\sigma_{100\tilde{x}[1]+50\tilde{x}[2]+50\tilde{x}[3]} = \sqrt{\begin{bmatrix}100 & 50 & 50\end{bmatrix} \Sigma_{\tilde{x}} \begin{bmatrix}100\\ 50\\ 50\end{bmatrix}} = 153 \text{ cents.} \tag{11.26}$$

The standard deviation in the price of Recipe 2 is

$$\sigma_{100\tilde{x}[1]+100\tilde{x}[2]} = \sqrt{\begin{bmatrix}100 & 100 & 0\end{bmatrix} \Sigma_{\tilde{x}} \begin{bmatrix}100\\ 100\\ 0\end{bmatrix}} = 190 \text{ cents.} \tag{11.27}$$

Even though the price of the imported cheese is more volatile than that of the local cheese, adding it to the recipe lowers the variance of the cost because it is uncorrelated with the other ingredients.

. . . .

A key application of Theorem 11.11 is determining the variance of a random vector in a specific direction. Let $b \in \mathbb{R}^d$ be a deterministic vector with unit ℓ_2 norm. We can decompose any d-dimensional random vector $\tilde{x}$ into a component collinear with b and a component orthogonal to b:

$$\tilde{x} = \underbrace{(b^T\tilde{x})b}_{\text{collinear with } b} + \underbrace{\tilde{x} - (b^T\tilde{x})b}_{\text{orthogonal to } b}. \tag{11.28}$$

By Theorem 11.11, we can easily compute the variance of the amplitude $b^T\tilde{x}$ of the collinear component using the covariance matrix

$$\text{Var}[b^T\tilde{x}] = b^T\Sigma_{\tilde{x}}b. \tag{11.29}$$

Example 11.13 (Variance of a random vector in a specific direction)**.** We consider the two-dimensional Gaussian random vector $\tilde{x}$ depicted in Figure 11.1. Its mean and covariance-matrix parameters equal

$$\mu := \begin{bmatrix}1\\ 2\end{bmatrix}, \qquad \Sigma := \begin{bmatrix}0.5 & -0.3\\ -0.3 & 0.5\end{bmatrix}. \tag{11.30}$$

We are interested in the variance of the $\tilde{x}$ in the direction of the vector

$$b := \begin{bmatrix}1\\ 0\end{bmatrix}. \tag{11.31}$$

Figure 11.3 shows the distribution of the random variable $b^T \operatorname{ct}(\tilde{x})$, which captures the fluctuations of $\tilde{x}$ around its mean in the direction of b ($\operatorname{ct}(\tilde{x}) := \tilde{x} - \mu$ denotes the corresponding centered random vector). The variance of $\tilde{x}$ in the direction of

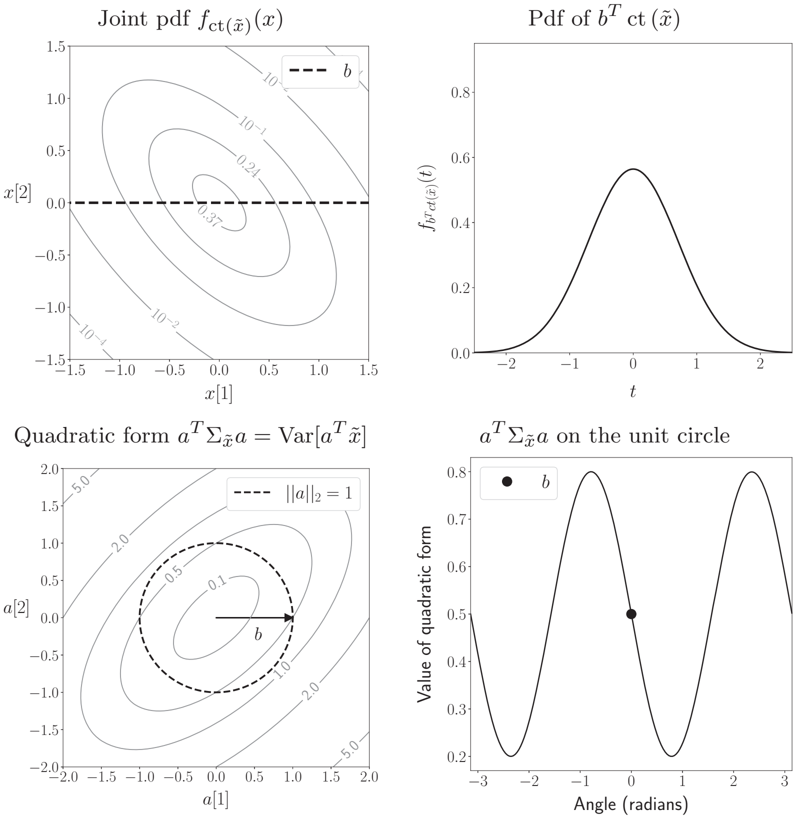


Figure 11.3 Variance in a specific direction. The top left graph shows a dashed line in the direction of the vector b in Example 11.13, superposed onto the contour lines of the joint pdf of the centered vector $\mathrm{ct}(\tilde{x}) := \tilde{x} - \mu$. The top right graph shows the pdf of the component of $\mathrm{ct}(\tilde{x})$ in the direction of b. The bottom left graph shows the contour lines of the quadratic form $a^T \Sigma_{\tilde{x}} a$, as well as the vector b (represented as an arrow). The values on the dashed unit circle encode the variance of $\tilde{x}$ in every possible direction. These values are plotted as a function of the angle in the bottom right graph. The circular marker corresponds to the vector b.

b is the variance of $b^T \mathrm{ct}(\tilde{x})$. By Theorem 11.11 it equals

$$\mathrm{Var}\left[b^T \mathrm{ct}(\tilde{x})\right] = \mathrm{Var}\left[b^T \tilde{x}\right] \tag{11.32}$$

$$= b^T \Sigma\, b = 0.5. \tag{11.33}$$

Figure 11.3 also depicts the contour lines of the quadratic form

$$q(a) := a^T \Sigma\, a. \tag{11.34}$$

The values of this function on the unit circle $\left\{a \in \mathbb{R}^d : ||a||_2 = 1\right\}$ encode the variance of $\tilde{x}$ in every possible direction.

11.3 The Sample Covariance Matrix

In order to describe the geometry of a dataset with d features, we can model the data as samples from a d-dimensional random vector and approximate its mean and covariance matrix. A reasonable estimate for the covariance matrix is the sample covariance matrix, which contains the sample variances and sample covariances of the features.

Definition 11.14 (Sample covariance matrix)**.** *Let $X := \{x_1, x_2, \ldots, x_n\}$ denote n vectors containing d features, and let $X[j] := \{x_1[j], \ldots, x_n[j]\}$ denote the bag of entries corresponding to the jth feature for $1 \leq j \leq d$. The sample covariance matrix of the data equals*

$$\Sigma_X := \frac{1}{n-1} \sum_{i=1}^{n} \operatorname{ct}(x_i) \operatorname{ct}(x_i)^T \tag{11.35}$$

$$= \begin{bmatrix} v(X[1]) & c(X[1], X[2]) & \cdots & c(X[1], X[d]) \\ c(X[1], X[2]) & v(X[2]) & \cdots & c(X[2], X[d]) \\ \vdots & \vdots & \ddots & \vdots \\ c(X[1], X[d]) & c(X[2], X[d]) & \cdots & v(X[d]) \end{bmatrix}, \tag{11.36}$$

where $\operatorname{ct}(x_i) := x_i - m(X)$ *is the result of centering the ith vector using the sample mean $m(X)$ for $1 \leq i \leq d$, $v(X[j])$ is the sample variance of $X[j]$ for $1 \leq j \leq d$, and $c(X[j], X[k])$ is the sample covariance of $X[j]$ and $X[k]$ for $1 \leq j, k \leq d$.*

A similar estimator for the covariance matrix can be obtained by fitting a Gaussian distribution to the data using maximum likelihood. By Theorem 5.26, the maximum-likelihood estimate of the covariance-matrix parameter of a Gaussian random vector equals

$$\Sigma_{\text{ML}} = \frac{1}{n} \sum_{i=1}^{n} \operatorname{ct}(x_i) \operatorname{ct}(x_i)^T, \tag{11.37}$$

which is essentially equal to the sample covariance matrix (unless n is very small).

Example 11.15 (Canadian cities: Sample covariance matrix)**.** The sample covariance matrix of the Canadian-city data in Example 11.5 is

$$\Sigma_X = \begin{bmatrix} 524.9 & -59.8 \\ -59.8 & 53.7 \end{bmatrix}. \tag{11.38}$$

Table 11.1 ***Sample covariance matrix and correlation of temperature data.*** *The sample covariance matrix of the data in Example 11.16 contains the sample variance of the temperature at each location, as well as the sample covariances between the temperatures. The correlation matrix shown below contains the corresponding sample correlation coefficients.*

Covariance matrix

	Tucson, AZ	Hilo, HI	Durham, NC	Ithaca, NY
Tucson, AZ	78.6	14.7	54.8	65.0
Hilo, HI	14.7	8.4	9.5	11.8
Durham, NC	54.8	9.5	89.4	97.4
Ithaca, NY	65.0	11.8	97.4	137.3

Correlation matrix

	Tucson, AZ	Hilo, HI	Durham, NC	Ithaca, NY
Tucson, AZ	1	0.57	0.65	0.63
Hilo, HI	0.57	1	0.35	0.35
Durham, NC	0.65	0.35	1	0.88
Ithaca, NY	0.63	0.35	0.88	1

The longitudes have much higher variance than the latitudes. Latitude and longitude are negatively correlated because people at higher longitudes (in the east) tend to live at lower latitudes (in the south).

. . . .

Example 11.16 (Temperatures in the United States)**.** We consider hourly temperature data at several weather stations in the United States in 2015, extracted from Dataset 9. The sample covariance matrix for four stations is shown in Table 11.1. The diagonal entries of the covariance matrix reveal that the temperature in Hawaii has much lower variance than the rest, whereas the temperature at Ithaca in upstate New York has very high variance. The off-diagonal entries show that all the temperatures are positively correlated.

Table 11.1 also shows the sample correlation coefficients obtained by normalizing each sample covariance using the corresponding sample standard deviations (see Definition 8.10). The correlation between the temperature in Hawaii and the rest is not very high, which makes sense given its geographic location. In contrast, temperatures in Ithaca and Durham are very correlated, as they are both on the East Coast of the United States.

. . . .

Just like the covariance matrix encodes the variance of any linear combination of a random vector, the sample covariance matrix encodes the sample variance of any linear combination of the data.

Theorem 11.17. *For any dataset $X = \{x_1, \ldots, x_n\}$ of d-dimensional data and any vector $a \in \mathbb{R}^d$, let*

$$X_a := \left\{a^T x_1, \ldots, a^T x_n\right\} \tag{11.39}$$

be the set of inner products between a and the elements in X. Then the sample variance of X_a equals

$$v\left(X_a\right) = a^T \Sigma_X a. \tag{11.40}$$

Proof The proof mimics the proof of Theorem 11.11, replacing the mean by the sample mean:

$$\begin{aligned}
v\left(X_a\right) &= \frac{1}{n-1}\sum_{i=1}^{n}\left(a^T x_i - \frac{1}{n}\sum_{j=1}^{n} a^T x_j\right)^2 && (11.41)\\
&= \frac{1}{n-1}\sum_{i=1}^{n}\left(a^T\left(x_i - \frac{1}{n}\sum_{j=1}^{n} x_j\right)\right)^2 && (11.42)\\
&= \frac{1}{n-1}\sum_{i=1}^{n}\left(a^T \operatorname{ct}\left(x_i\right)\right)^2 && (11.43)\\
&= \frac{1}{n-1}\sum_{i=1}^{n} a^T \operatorname{ct}\left(x_i\right) \operatorname{ct}\left(x_i\right)^T a && (11.44)\\
&= a^T\left(\frac{1}{n-1}\sum_{i=1}^{n} \operatorname{ct}\left(x_i\right) \operatorname{ct}\left(x_i\right)^T\right) a && (11.45)\\
&= a^T \Sigma_X a. && (11.46)
\end{aligned}$$

■

In order to describe the geometry of a dataset, we can compute its sample variance in a specific direction by applying Theorem 11.17.

Example 11.18 (Canadian cities: Variance in a specific direction)**.** Our goal is to quantify how the distribution of Canadian cities varies in the southwest-northeast direction corresponding to the unit ℓ_2 norm vector

$$b := \frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix}. \tag{11.47}$$

We denote the inner products between the data and b by

$$X_b := \left\{b^T x_1, \ldots, b^T x_n\right\}. \tag{11.48}$$

By Theorem 11.17, we can compute the sample variance of this set directly from the sample covariance matrix Σ_X computed in Example 11.15:

$$\begin{aligned}
v(X_b) &= \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\end{bmatrix}\Sigma_X \frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix} && (11.49)\\
&= 229, && (11.50)
\end{aligned}$$

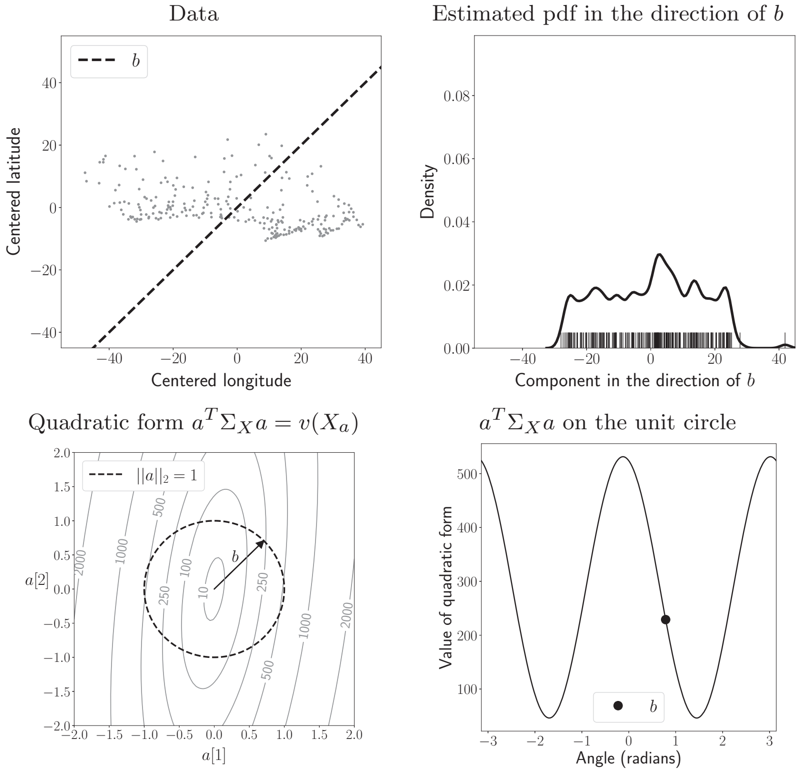


Figure 11.4 Sample variance in a specific direction. The top left scatterplot shows the centered data from Example 11.18 and the direction of the vector b (dashed line). The top right plot shows a rug plot of the set X_b containing the components of the data in that direction, and a kernel-density estimate of its probability density. The bottom left plot shows the contours of the quadratic form $a^T\Sigma_X a$, as well as the vector b (represented by an arrow). The values on the dashed unit circle are the sample variances of the dataset in every possible direction. These values are plotted as a function of the angle in the bottom right graph. The circular marker corresponds to the vector b.

so the standard deviation is 15.1. Figure 11.4 depicts b on the scatterplot of the data, as well as a kernel density estimate of X_b. The figure also shows the contour lines of the quadratic form

$$q(a) := a^T\Sigma_X a, \tag{11.51}$$

and its values on the circle $\left\{a \in \mathbb{R}^d : ||a||_2 = 1\right\}$, which capture the sample variance of the dataset in every possible direction.

. . . .

11.4 Principal Component Analysis

Principal component analysis (PCA) is a very popular technique to analyze high-dimensional datasets. PCA identifies the directions of maximum variance in the data, and decomposes the data into components corresponding to each of these directions. Intuitively, we expect directions in which the data varies more to be more informative, as opposed to directions in which the data are almost constant. Section 11.4.1 defines PCA for random vectors. Section 11.4.2 describes how to perform PCA on a dataset. Section 11.4.3 provides an intuitive explanation of the mathematics behind PCA.

11.4.1 Principal Component Analysis Of A Random Vector

As explained in Section 11.2, the covariance matrix $\Sigma_{\tilde{x}}$ of a random vector $\tilde{x}$ encodes the variance of the vector in every direction. Here we explain how to leverage this to find the direction in which the variance is highest. By Theorem 11.11, the variance in the direction of a vector a is equal to $a^T \Sigma_{\tilde{x}} a$, so in order to find the direction of maximum variance, we need to maximize the function

$$q(a) := a^T \Sigma_{\tilde{x}} a \tag{11.52}$$

over the set of unit ℓ_2 norm vectors $\{a : ||a||_2 = 1\}$. The spectral theorem is a fundamental result in linear algebra, which states that the maximum is achieved at an eigenvector of $\Sigma_{\tilde{x}}$.

Theorem 11.19 (Spectral theorem for symmetric matrices)**.** *If $M \in \mathbb{R}^{d \times d}$ is symmetric, then it has an eigendecomposition of the form*

$$M = \begin{bmatrix} u_1 & u_2 & \cdots & u_d \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ & & \cdots & \\ 0 & 0 & \cdots & \lambda_d \end{bmatrix} \begin{bmatrix} u_1 & u_2 & \cdots & u_d \end{bmatrix}^T, \tag{11.53}$$

where the eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$ are real and the eigenvectors u_1, u_2, $\ldots$, u_n are real and orthogonal. In addition,

$$\lambda_1 = \max_{||a||_2=1} a^T M a, \tag{11.54}$$

$$u_1 = \arg \max_{||a||_2=1} a^T M a, \tag{11.55}$$

$$\lambda_k = \max_{||a||_2=1, a \perp u_1, \ldots, u_{k-1}} a^T M a, \quad 2 \leq k \leq d-1, \tag{11.56}$$

$$u_k = \arg \max_{||a||_2=1, a \perp u_1, \ldots, u_{k-1}} a^T M a, \quad 2 \leq k \leq d-1, \tag{11.57}$$

$$\lambda_d = \min_{||a||_2=1} a^T M a, \tag{11.58}$$

$$u_d = \arg \min_{||a||_2=1} a^T M a. \tag{11.59}$$

In Section 11.4.3 we explain why the maximum and minimum of the function $a^T M a$ on the unit sphere are reached at eigenvectors of M. For a more thorough proof of the spectral theorem, we refer to any advanced textbook on linear algebra.

By the spectral theorem and Theorem 11.11, the eigendecomposition of the covariance matrix of a random vector yields a *ranking* of orthogonal directions. The largest eigenvalue λ_1 is the highest variance of the random vector in any direction, attained in the direction of the first eigenvector u_1. In directions orthogonal to u_1, the highest variance is the second largest eigenvalue λ_2, attained in the direction of the second eigenvector u_2. In general, when restricted to the orthogonal complement of the span of u_1, …, u_k for $1 \leq k \leq d-1$, the variance is highest in the direction of the $k+1$th eigenvector u_{k+1}, and equals the corresponding eigenvalue.

Theorem 11.20 (Directions of maximum and minimum variance)**.** *Let $\tilde{x}$ be a d-dimensional random vector with covariance matrix $\Sigma_{\tilde{x}}$. The eigenvectors u_1, …, u_d and eigenvalues $\lambda_1 > \ldots > \lambda_d$ of $\Sigma_{\tilde{x}}$ satisfy*

$$\lambda_1 = \max_{||a||_2=1} \operatorname{Var}[a^T \tilde{x}], \tag{11.60}$$

$$u_1 = \arg \max_{||a||_2=1} \operatorname{Var}[a^T \tilde{x}], \tag{11.61}$$

$$\lambda_k = \max_{||a||_2=1, a \perp u_1, \ldots, u_{k-1}} \operatorname{Var}[a^T \tilde{x}], \quad 2 \leq k \leq d-1, \tag{11.62}$$

$$u_k = \arg \max_{||a||_2=1, a \perp u_1, \ldots, u_{k-1}} \operatorname{Var}[a^T \tilde{x}], \quad 2 \leq k \leq d-1, \tag{11.63}$$

$$\lambda_d = \min_{||a||_2=1} \operatorname{Var}[a^T \tilde{x}], \tag{11.64}$$

$$u_d = \arg \min_{||a||_2=1} \operatorname{Var}[a^T \tilde{x}]. \tag{11.65}$$

Proof Covariance matrices are symmetric:

$$\Sigma_{\tilde{x}}^T = \left(\operatorname{E}\left[\tilde{x}\tilde{x}^T\right]\right)^T \tag{11.66}$$

$$= \operatorname{E}\left[\left(\tilde{x}\tilde{x}^T\right)^T\right] \tag{11.67}$$

$$= \operatorname{E}\left[\tilde{x}\tilde{x}^T\right] = \Sigma_{\tilde{x}}. \tag{11.68}$$

Since $\operatorname{Var}[a^T \tilde{x}] = a^T \Sigma_{\tilde{x}} a$ for any $a \in \mathbb{R}^d$ by Theorem 11.11, the result follows from Theorem 11.19 setting $M := \Sigma_{\tilde{x}}$. ■

We call the directions of the eigenvectors *principal directions.* The component of the centered random vector $\operatorname{ct}(\tilde{x}) := \tilde{x} - \operatorname{E}[\tilde{x}]$ in each principal direction is called a *principal component*,

$$\widetilde{w}_i := u_i^T \operatorname{ct}(\tilde{x}), \quad 1 \leq i \leq d. \tag{11.69}$$

The variance captured by each principal component is equal to the eigenvalue of the covariance matrix associated to the corresponding principal direction.

Lemma 11.21. *Let $\tilde{x}$ be a d-dimensional random vector with covariance matrix $\Sigma_{\tilde{x}}$. The variance of the ith principal component $\widetilde{w}_i$ equals the corresponding eigenvalue λ_i of $\Sigma_{\tilde{x}}$.*

Proof The principal direction u_i is by definition an eigenvector of the covariance matrix with eigenvalue λ_i, so

$$\begin{align}
\operatorname{Var}\left[\widetilde{w}_i\right] &= u_i^T \Sigma_{\tilde{x}} u_i \tag{11.70}\\
&= \lambda_i u_i^T u_i \tag{11.71}\\
&= \lambda_i. \tag{11.72}
\end{align}$$

■

The principal components of a random vectors are uncorrelated, which means that there is no linear relationship between them.

Lemma 11.22 (Principal components are uncorrelated)**.** *Let $\tilde{x}$ be a d-dimensional random vector with covariance matrix $\Sigma_{\tilde{x}}$. The principal components $\widetilde{w}_1$, $\widetilde{w}_2$, ..., $\widetilde{w}_d$ of $\tilde{x}$ are uncorrelated.*

Proof Let u_i be the eigenvector of the covariance matrix corresponding to the ith principal component. By Theorem 11.20, the eigenvectors are orthogonal. The mean of the principal components is zero by linearity of expectation,

$$\begin{align}
\operatorname{E}\left[\tilde{w}_i\right] &= \operatorname{E}\left[u_i^T \operatorname{ct}\left(\tilde{x}\right)\right] \tag{11.73}\\
&= u_i^T \operatorname{E}\left[\operatorname{ct}\left(\tilde{x}\right)\right] = 0, \tag{11.74}
\end{align}$$

because $\operatorname{E}\left[\operatorname{ct}\left(\tilde{x}\right)\right] = 0$. Consequently, if $i \neq j$,

$$\begin{align}
\operatorname{Cov}[\widetilde{w}_i \widetilde{w}_j] = \operatorname{E}[\widetilde{w}_i \widetilde{w}_j] &= \operatorname{E}\left[u_i^T \operatorname{ct}\left(\tilde{x}\right) \operatorname{ct}\left(\tilde{x}\right)^T u_j\right] \tag{11.75}\\
&= u_i^T \operatorname{E}[\operatorname{ct}\left(\tilde{x}\right) \operatorname{ct}\left(\tilde{x}\right)^T] u_j \tag{11.76}\\
&= u_i^T \Sigma_{\tilde{x}} u_j \tag{11.77}\\
&= \lambda_j u_i^T u_j \tag{11.78}\\
&= 0. \tag{11.79}
\end{align}$$

■

Example 11.23 (Principal component analysis of a Gaussian random vector)**.** We apply PCA to the Gaussian random vector $\tilde{x}$ in Example 11.13. The first principal direction

$$u_1 = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix} \tag{11.80}$$

is the direction of maximum variance, which equals the corresponding eigenvalue

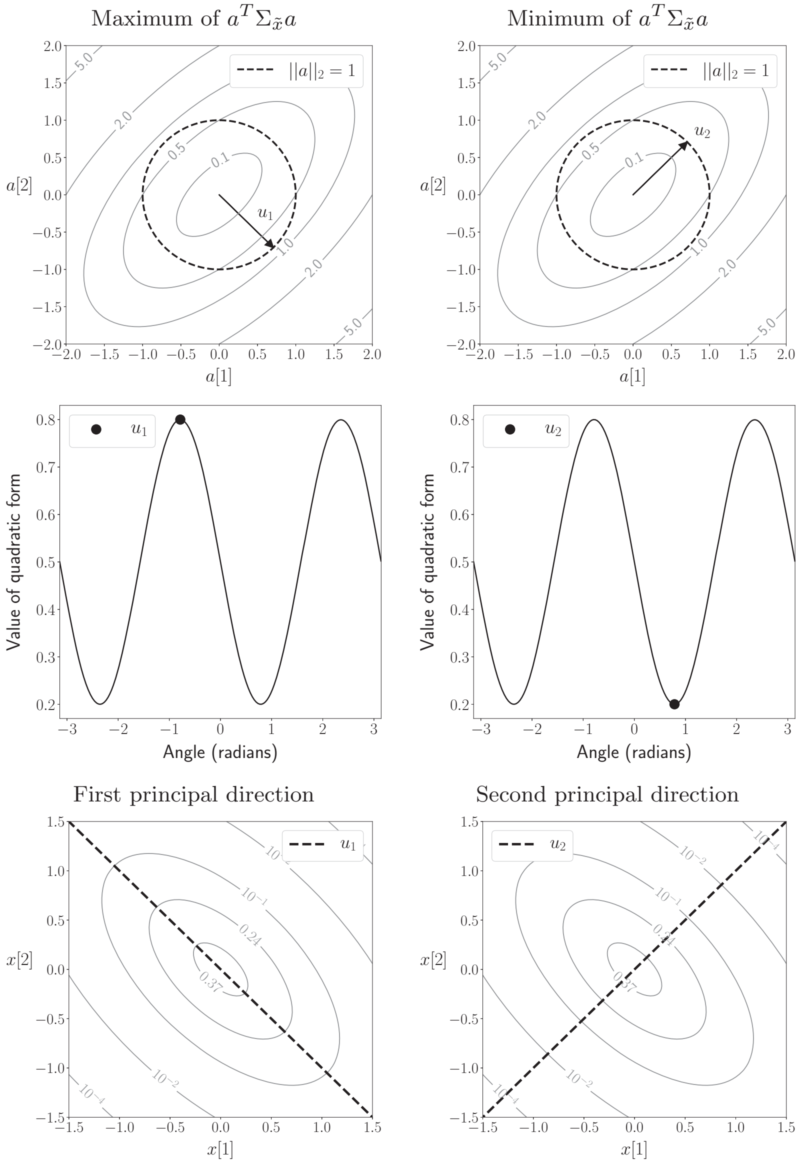


Figure 11.5 Principal directions of a Gaussian random vector. The top row shows the first (left) and second (right) principal directions of the Gaussian random vector $\tilde{x}$ in Example 11.13, derived in Example 11.23. The principal directions are superposed onto the contour lines of the quadratic form $a^T \Sigma_{\tilde{x}} a$. The values on the dashed unit circle are plotted as a function of the angle in the middle row. The first and second principal directions attain the maximum and minimum of the restricted function (depicted by circular markers). The bottom row shows that the principal directions are aligned with the axes of the ellipsoidal contour lines of the joint pdf of the centered vector $\text{ct}(\tilde{x})$.

$\lambda_1 = 0.8$. Since the random vector has only two dimensions, the second principal direction

$$u_2 = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \tag{11.81}$$

is the direction of minimum variance, which again equals the corresponding eigenvalue $\lambda_2 = 0.2$. In Figure 11.5 we see that these two directions indeed attain the maximum and minimum of the quadratic form $a^T \Sigma\, a$ on the unit circle where $||a||_2 = 1$. The bottom row of the figure shows the principal directions superposed onto the contour lines of the joint pdf of the centered random vector $\operatorname{ct}(\tilde{x})$.

As described in Section 5.10.1, the contour lines of Gaussian random vectors are ellipsoidal. PCA enables us to find the axes of these ellipsoids. The major axis is equal to the first principal direction, because it is the direction in which the density is the most *stretched* and hence has the highest variance. The second major axis is equal to the second principal direction, and so on. The analysis of the ellipsoidal contour lines of the joint pdf of $\tilde{x}$ in Example 5.22 confirms that the axes of the ellipses indeed coincide with the principal directions.

The top row of Figure 11.6 shows the joint pdf of the principal components $\widetilde{w}_1 := u_1^T \operatorname{ct}(\tilde{x})$ and $\widetilde{w}_2 := u_2^T \operatorname{ct}(\tilde{x})$, and compares it to the joint pdf of $\tilde{x}$. To obtain the principal components, we first center the joint pdf by subtracting the mean, and then rotate it so that the axes of the ellipsoidal contour lines align with the coordinate axes. This maximizes the variance of the marginal distribution along the horizontal axis and minimizes the variance of the marginal distribution along the vertical axis. The marginal pdfs are shown on the bottom row of Figure 11.6.

. . . .

11.4.2 Principal Component Analysis Of A Dataset

As explained in Section 11.4.1, principal component analysis (PCA) extracts the directions of maximum (and minimum variance) of a random vector from the eigendecomposition of its covariance matrix. The same procedure can be applied to the sample covariance matrix of a dataset.

Definition 11.24 (Principal component analysis)**.** *To perform principal component analysis of a dataset X containing n vectors $x_1, x_2, \ldots, x_n \in \mathbb{R}^d$ with d features each, we apply the following steps:*

1. *We compute the sample covariance matrix of the data Σ_X following Definition 11.14.*
2. *We compute the eigendecomposition of Σ_X. The eigenvectors $u_1, \ldots, u_d$ (ordered according to the corresponding eigenvalues $\lambda_1 \geq \cdots \geq \lambda_d$), are the principal directions of the data.*

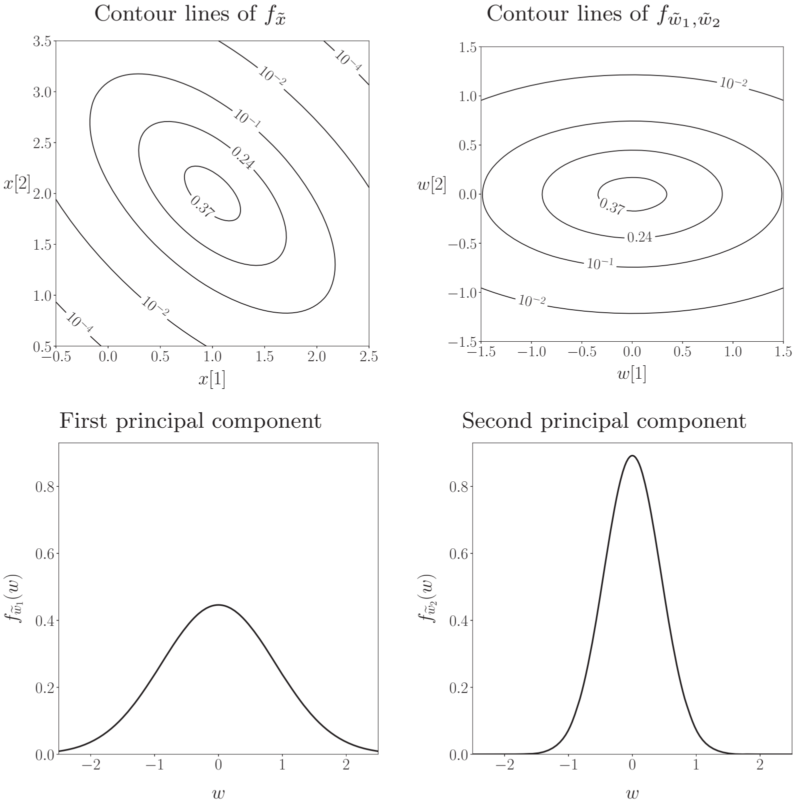


Figure 11.6 Principal components of a Gaussian random vector. The top left graph shows the contour lines of the joint pdf $f_{\tilde{x}}$ of the Gaussian random vector in Example 11.13. The top right graph shows the contour lines of the joint pdf of its first and second principal components $\tilde{w}_1$ and $\tilde{w}_2$, which are obtained by centering the ellipsoidal contour lines of $f_{\tilde{x}}$ and then rotating them so that their axes align with the coordinate axes. This yields a horizontal component with maximum variance ($\tilde{w}_1$) and a vertical component with minimum variance ($\tilde{w}_2$). The bottom row shows the marginal pdfs of $\tilde{w}_1$ and $\tilde{w}_2$.

3 We center the data and compute the principal components

$$w_j[i] := u_j^T \operatorname{ct}(x_i), \quad 1 \leq i \leq n, \ 1 \leq j \leq d, \tag{11.82}$$

where $\operatorname{ct}(x_i) := x_i - m(X)$, *and* $m(X)$ *denotes the sample mean (see Definition 11.4).*

When we perform PCA on a dataset, the resulting principal directions maximize (and minimize) the sample variance.

Theorem 11.25 (Directions of maximum and minimum sample variance)**.** *Let*

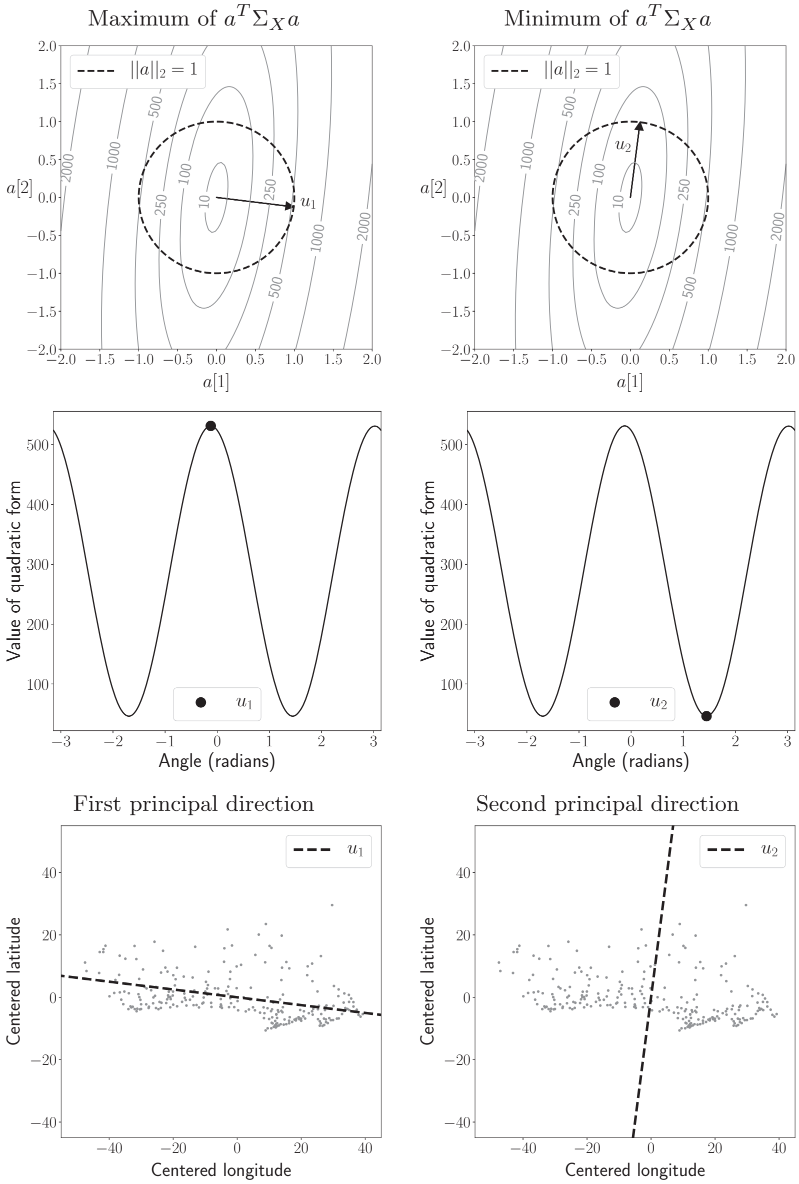


Figure 11.7 Principal directions of a dataset. The top row shows the first (left) and second (right) principal directions of the data in Example 11.18, computed following Definition 11.24. The principal directions are superposed onto the contour lines of the quadratic form $a^T \Sigma_X a$. The values on the dashed unit circle are plotted as a function of the angle in the middle row. The first and second principal directions attain the maximum and minimum of the restricted function (depicted by circular markers). The bottom row shows that the principal directions are aligned with the directions of maximum and minimum sample variance of the data.

X contain n vectors $x_1, x_2, \ldots, x_n \in \mathbb{R}^d$ with sample covariance matrix Σ_X. The eigenvectors u_1, ..., u_d and eigenvalues $\lambda_1 > \ldots > \lambda_d$ of Σ_X are the directions of maximum and minimum sample variance,

$$\lambda_1 = \max_{||a||_2=1} v(X_a), \tag{11.83}$$

$$u_1 = \arg \max_{||a||_2=1} v(X_a), \tag{11.84}$$

$$\lambda_k = \max_{||a||_2=1, a \perp u_1, \ldots, u_{k-1}} v(X_a), \quad 2 \leq k \leq d-1, \tag{11.85}$$

$$u_k = \arg \max_{||a||_2=1, a \perp u_1, \ldots, u_{k-1}} v(X_a), \quad 2 \leq k \leq d-1, \tag{11.86}$$

$$\lambda_d = \min_{||a||_2=1} v(X_a), \tag{11.87}$$

$$u_d = \arg \min_{||a||_2=1} v(X_a), \tag{11.88}$$

where X_a denotes the components of the data in the direction of a

$$X_a := \left\{ a^T x_1, \ldots, a^T x_n \right\}. \tag{11.89}$$

Proof Sample covariance matrices are symmetric:

$$\Sigma_X^T = \left(\frac{1}{n-1} \sum_{i=1}^{n} \operatorname{ct}(x_i) \operatorname{ct}(x_i)^T \right)^T \tag{11.90}$$

$$= \frac{1}{n-1} \sum_{i=1}^{n} \operatorname{ct}(x_i) \operatorname{ct}(x_i)^T = \Sigma_X, \tag{11.91}$$

so the result follows from Theorems 11.19 and 11.17. ■

In words, u_1 is the direction of maximum sample variance, u_2 is the direction of maximum sample variance orthogonal to u_1, and u_k is the direction of maximum variation that is orthogonal to u_1, u_2, ..., u_{k-1}. The sample variances in each of these directions are equal to the corresponding eigenvalues.

Figure 11.7 shows the two principal directions of the two-dimensional data in Example 11.18, computed following Definition 11.24. As established by Theorem 11.25, these directions maximize and minimize the quadratic form $a^T \Sigma_X \, a$ on the unit circle where $||a||_2 = 1$, and therefore identify the directions in which the data have maximum and minimum variance. The principal components are then obtained by centering the data and rotating it so that that the coordinate axes align with the principal directions, as illustrated in Figure 11.8. After the rotation, the sample variance is maximized along the horizontal axis and minimized along the vertical axis. Analogously to Lemma 11.22, the sample correlation between the principal components is zero (see Exercise 11.4).

Up to now we have focused on two-dimensional examples because they are easy to visualize. However, in practice, we don't really need PCA to analyze the structure of two-dimensional data: we can just plot them and take a look! In contrast, analyzing higher dimensional data is much more challenging. The following example considers a dataset with 4,096 features. We can obviously not

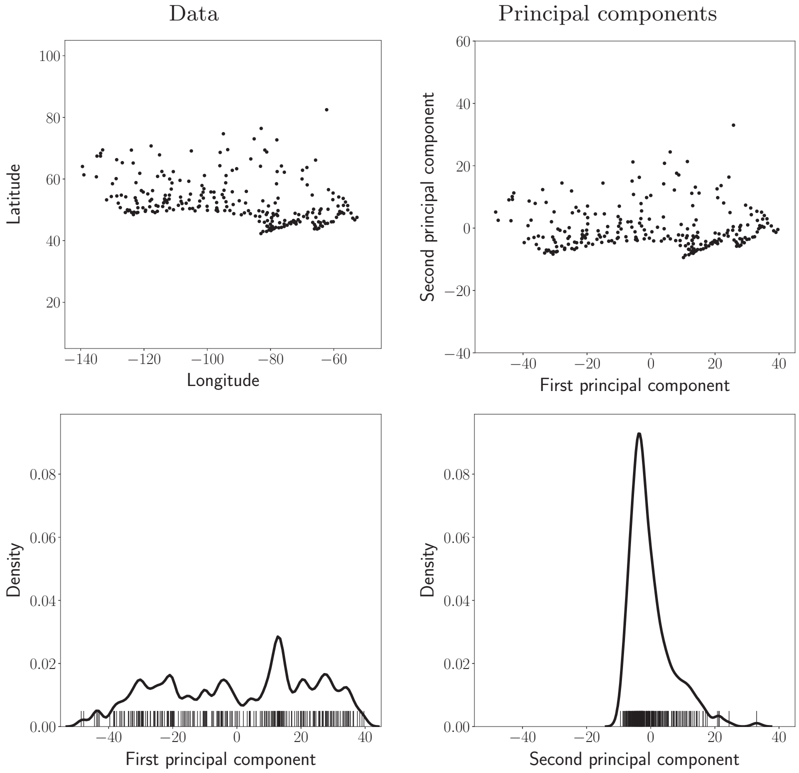


Figure 11.8 Principal components of a dataset. The top left graph shows the data in Example 11.18. The top right graph shows the scatterplot of the first and second principal components, obtained by centering the data and then rotating them so that their axes align with the principal directions. This yields a horizontal component with maximum sample variance and a vertical component with minimum variance. The bottom row shows estimates of the marginal pdfs of the principal components obtained via kernel density estimation.

plot the data points in 4,096 dimensions! To make matters worse, estimating the joint pdf using a nonparametric density estimate is out of the question because of the curse of dimensionality (see Section 4.7). It is in such situations that PCA shines, allowing us to identify the components of the data which account for more variance.

Example 11.26 (Faces: Principal component analysis)**.** We use PCA to analyze Dataset 20, where each 64×64 image is represented by a 4,096-dimensional vector. Figure 11.9 shows several principal directions, together with the variance of the corresponding principal components. The first principal directions capture coarse-level face structure (eyebrows, nose, mouth, face contour) and also the

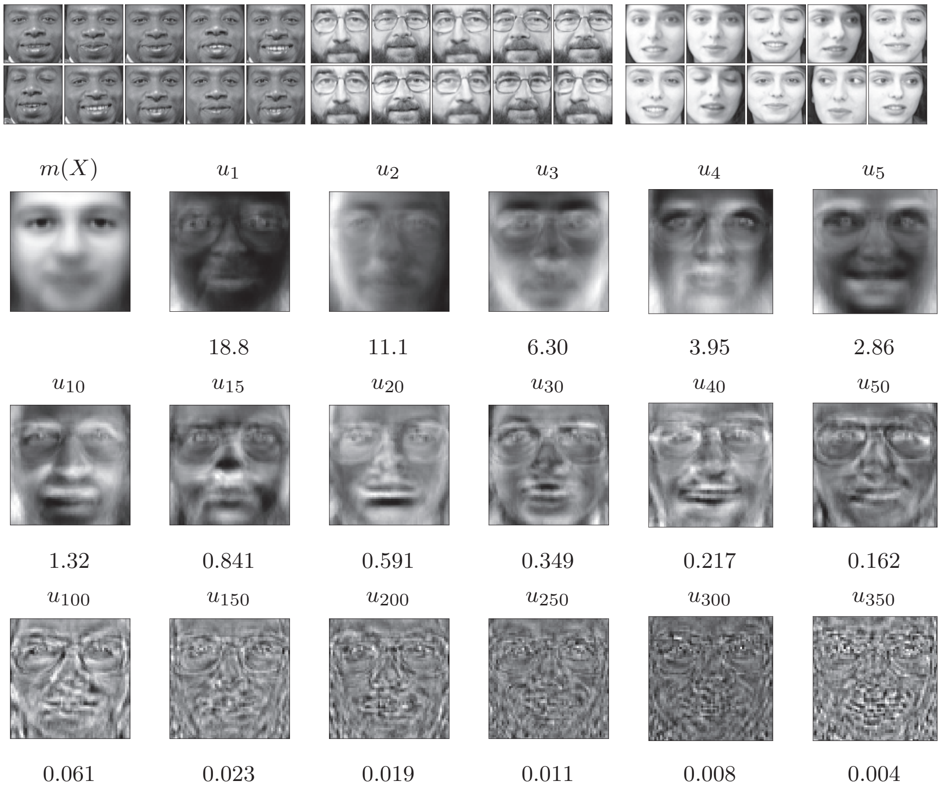


Figure 11.9 Principal directions of face data. The top row shows three different individuals from the dataset in Example 11.26. The sample mean $m(X)$ and the principal directions u_i, $1 \leq d$, obtained by applying PCA to the data are depicted below. The sample variance of each principal component is listed under the corresponding principal direction.

illumination of the photographs (see u_2 for example). These are the characteristics that account for most of the variance in the dataset.

....

11.4.3 Why Eigenvectors? The Mathematics Behind PCA

In this section we explain how to prove the spectral theorem and establish that the eigenvectors of a covariance matrix are the directions of maximum and minimum variance. Consider the function

$$q(a) := a^T M a, \tag{11.92}$$

where M is a $d \times d$ symmetric matrix. Such functions are called quadratic forms, because they are multidimensional extensions of quadratic functions. We are in-

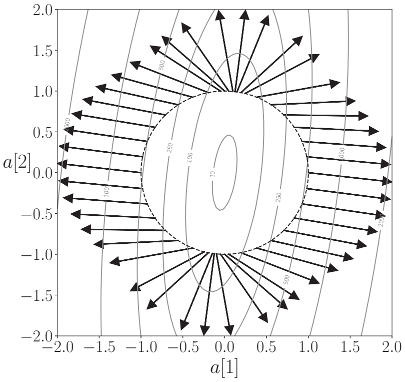


Figure 11.10 Gradients of a quadratic form. Each arrow indicates the direction of the gradient of the quadratic form associated to the sample covariance matrix in Example 11.15 at different points of the unit circle.

terested in the value of $q(a)$ for vectors lying on the unit sphere, i.e. such that $||a||_2 = 1$. If the matrix is a covariance matrix, $q(a)$ represents the variance in the direction of a by Theorems 11.11 and 11.17. In particular, the direction of maximum variance is the value of a (on the unit sphere) where q reaches a maximum value. Fortunately, such a point is guaranteed to exist. The reason is that quadratic forms are continuous and the unit sphere is a closed and bounded set, so its image is also closed and bounded. Consequently, it must contain all its limit points and cannot grow indefinitely (this is known as the extreme value theorem in calculus).

Now let us characterize the direction that attains the maximum. The quadratic function is differentiable because it is a second-order polynomial. The gradient of the quadratic form at a is

$$\nabla q(a) = 2Ma. \tag{11.93}$$

Figure 11.10 shows the direction of the gradient on the unit circle for the quadratic form associated to the sample covariance matrix in Example 11.15. The gradient encodes the rate of change of the quadratic function in different directions. At a point b, the rate of change of q in the direction of a unit vector h is equal to the inner product between $\nabla q(b)$ and h. This rate is the directional derivative of q at b

$$\begin{aligned} q'_h(b) &:= \lim_{\epsilon \to 0} \frac{q(b + \epsilon h) - q(b)}{\epsilon} && (11.94) \\ &= \nabla q(b)^T h. && (11.95) \end{aligned}$$

If the derivative is positive, $h^T \nabla q(b) > 0$, then the function increases in that direction, i.e. for small enough $\epsilon > 0$, $q(b + \epsilon h) > q(b)$. If u_1 is the point at

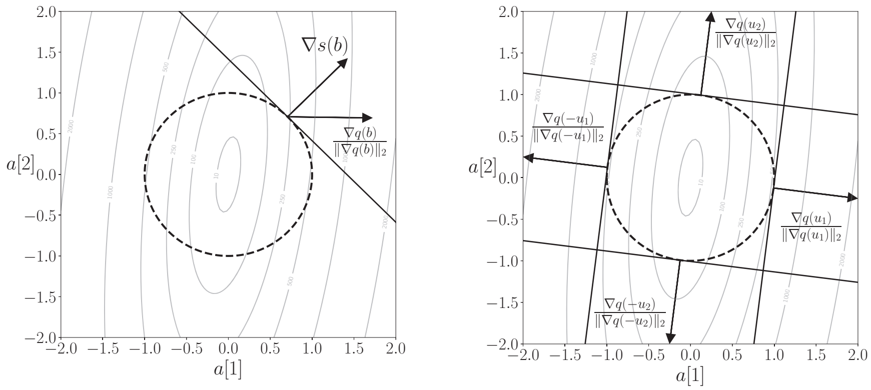


Figure 11.11 Proof of the spectral theorem. Illustration of the proof of the spectral theorem, using the quadratic form associated to the sample covariance matrix in Example 11.15. On the left, the gradient of the quadratic form is not orthogonal to the tangent plane (black solid line) of the unit sphere (dashed circle). Consequently, we can find a point on the sphere that attains a higher value. The right plot shows four points at which the gradient of the quadratic form is orthogonal to the tangent plane. These are the maxima and minima of the quadratic form on the unit circle, and correspond to the principal directions of the data.

which the maximum is attained, the directional derivative *cannot be positive* for directions that stay in our set of interest (i.e. the unit sphere), because this would imply that there is another point in the set that attains a larger value than u_1. Here we hit a minor difficulty: since u_1 is on the sphere, $u_1 + \epsilon h$ can never be on the sphere, because the sphere is a curved surface! Instead, we need to consider directions that *almost* stay on the sphere, in the sense that they belong to its *tangent plane*.

The unit sphere is a level surface of the function $s(a) := a^T a$ (it contains every point a such that $s(a) = 1$). The tangent plane $\mathcal{T}$ of the level surface of a differentiable function s at a point b is the set of vectors orthogonal to the gradient of s. A point y belongs to $\mathcal{T}$ if

$$\nabla s(b)^T (y - b) = 0. \tag{11.96}$$

For such points, if $y - b$ is small, then $s(y) \approx s(b) + \nabla s(b)^T (y - b) = s(b)$, so y is *almost* on the level surface.

If q attains its maximum at u_1, then there cannot be any point b in the tangent plane of the sphere at u_1 such that $\nabla q(u_1)^T (b - u_1) > 0$. If that were the case, then for small enough ϵ, there would exist a point y on the sphere close enough to $u_1 + \epsilon(b - u_1)$, so that $q(y) \approx q(u_1 + \epsilon(b - u_1)) > q(u_1)$, and therefore u_1 would not be the maximum. This implies that $\nabla q(u_1)$ must be orthogonal to the tangent plane, and therefore, collinear with the gradient of s. By the same argument, if q

attains its minimum at a certain point of the sphere, then its gradient must also be collinear with the gradient of s (otherwise, we can find a direction belonging to the tangent plane in which the function is decreasing). Figure 11.11 illustrates this. On the left, the gradients of q and s do not align at a certain point b, so we can find a direction in which the quadratic form increases locally on the unit circle. On the right, the gradients are shown to be collinear at the two maxima and the two minima of the quadratic form.

Collinearity of the gradients implies that there exists a constant $\lambda_1 \in \mathbb{R}$ such that

$$\nabla q(u_1) = \lambda_1 \nabla s(u_1). \tag{11.97}$$

Since $\nabla q(u_1) = 2Mu_1$ and $\nabla s(u_1) = 2u_1$, we conclude that

$$Mu_1 = \lambda_1 u_1. \tag{11.98}$$

In words, the maximum is attained at an eigenvector of the matrix M! The same argument can be used to establish that the minimum is also attained at an eigenvector. To complete the proof of the spectral theorem, we can apply the same ideas on the orthogonal complement of u_1 (and then to the orthogonal complement of the span of u_1 and u_2, and so on).

11.5 Dimensionality Reduction

Data with a large number of features can be difficult to analyze and process. The goal of dimensionality-reduction techniques is to embed such data in a low-dimensional space where they can be described in terms of a small number of variables. This is a crucial preprocessing step in many applications.

A popular choice is to perform *linear* dimensionality reduction, where the lower-dimensional representation is obtained by computing the inner products of each data point with a small number of basis vectors. Let us interpret the data as samples from a d-dimensional random vector $\tilde{x}$. Dimensionality reduction is typically applied after centering the data by subtracting their mean, so we assume that $\tilde{x}$ has zero mean. Consider an orthonormal basis of $\mathbb{R}^d$ formed by the vectors b_1, b_2, …, b_d, and let us express $\tilde{x}$ as a linear combination of the basis vectors,

$$\tilde{x} = \sum_{i=1}^{d} \tilde{a}[i] b_i, \qquad \tilde{a}[i] := b_i^T \tilde{x}. \tag{11.99}$$

The coefficients $\tilde{a}[1]$, …, $\tilde{a}[d]$ are a d-dimensional representation of $\tilde{x}$. In order to compute a representation of lower dimensionality, we can truncate the coefficients and only use the first k. The approximation to the original vector corresponding to this representation is

$$\underset{b_1,\ldots,b_k}{\text{approx}}(\tilde{x}) := \sum_{i=1}^{k} \tilde{a}[i] b_i, \qquad \tilde{a}[i] := b_i^T \tilde{x}. \tag{11.100}$$

A key challenge is how to select the k vectors $b_1, \ldots, b_k$, so that the representation preserves as much information about $\tilde{x}$ as possible.

The best dimensionality-reduction scheme may vary depending on the specific downstream task of interest, but a reasonable general-purpose criterion is to minimize the error of the corresponding approximation in the original d-dimensional space. To this end, we can minimize the average sum of squared errors, or equivalently, the squared ℓ_2-norm of the approximation error. Consider the representation of $\tilde{x}$ in terms of the basis vectors, and its decomposition in terms of the approximation and the corresponding error:

$$\underbrace{\sum_{i=1}^{d} \tilde{a}[i] b_i}_{\tilde{x}} = \underbrace{\sum_{i=1}^{k} \tilde{a}[i] b_i}_{\underset{b_1,\ldots,b_k}{\operatorname{approx}}(\tilde{x})} + \underbrace{\sum_{i=k+1}^{d} \tilde{a}[i] b_i}_{\text{error}}. \tag{11.101}$$

The approximation and the error are orthogonal, as they are linear combinations of orthogonal vectors, so by the Pythagorean theorem,

$$||\tilde{x}||_2^2 = \left|\left|\sum_{i=1}^{k} \tilde{a}[i] b_i\right|\right|_2^2 + ||\text{error}||_2^2, \tag{11.102}$$

which implies

$$\begin{align} ||\text{error}||_2^2 &= ||\tilde{x}||_2^2 - \left|\left|\sum_{i=1}^{k} \tilde{a}[i] b_i\right|\right|_2^2 \tag{11.103}\\ &= ||\tilde{x}||_2^2 - \sum_{i=1}^{k} \tilde{a}[i]^2 \tag{11.104}\\ &= ||\tilde{x}||_2^2 - \sum_{i=1}^{k} \left(b_i^T \tilde{x}\right)^2, \tag{11.105} \end{align}$$

where (11.104) holds because $b_1, \ldots, b_k$ are orthonormal. By linearity of expectation, the mean of the squared ℓ_2 norm of the error equals

$$\begin{align} \mathrm{E}\left[\left|\left|\tilde{x} - \underset{b_1,\ldots,b_k}{\operatorname{approx}}(\tilde{x})\right|\right|_2^2\right] &= \mathrm{E}\left[||\tilde{x}||_2^2\right] - \sum_{i=1}^{k} \mathrm{E}\left[\left(b_i^T \tilde{x}\right)^2\right] \tag{11.106}\\ &= \mathrm{E}\left[||\tilde{x}||_2^2\right] - \sum_{i=1}^{k} \operatorname{Var}\left[b_i^T \tilde{x}\right]. \tag{11.107} \end{align}$$

The first term $\mathrm{E}\left[||\tilde{x}||_2^2\right]$ does not depend on the chosen basis, so in order to minimize the approximation error, we should choose the k basis vectors so that the sum of the variances in those directions is as large as possible. Consequently, using the first k principal directions of the random vector $\tilde{x}$ seems like the way to go. The corresponding low-dimensional representation of $\tilde{x}$ consists of the first

k principal components. The following theorem establishes that this choice is optimal in terms of mean squared ℓ_2-norm error for any choice of k.

Theorem 11.27 (Linear dimensionality reduction via PCA is optimal)**.** *Let $\tilde{x}$ be a d-dimensional random vector with zero mean and covariance matrix $\Sigma_{\tilde{x}}$ and let $k < d$. We denote the k-dimensional linear approximation of $\tilde{x}$ in a basis of k orthonormal vectors $b_1, \ldots, b_k$ by*

$$\underset{b_1,\ldots,b_k}{\operatorname{approx}}(\tilde{x}) := \sum_{i=1}^{k} b_i^T \tilde{x}\ b_i. \tag{11.108}$$

The first k principal directions associated to the k largest eigenvalues of $\Sigma_{\tilde{x}}$ yield the optimal k-dimensional linear approximation in terms of mean squared ℓ_2-norm error,

$$\{u_1, \ldots, u_k\} = \arg \min_{\substack{\{b_1,\ldots,b_k\} \\ ||b_i||_2=1, 1\leq i\leq k \\ b_i \perp b_j, i\neq j}} \operatorname{E}\left[\left|\left|\tilde{x} - \underset{b_1,\ldots,b_k}{\operatorname{approx}}(\tilde{x})\right|\right|_2^2\right]. \tag{11.109}$$

The optimal linear k-dimensional approximation is

$$\underset{u_1,\ldots,u_k}{\operatorname{approx}}(\tilde{x}) := \sum_{i=1}^{k} \tilde{w}_i u_i, \tag{11.110}$$

where $\tilde{w}_i := u_i^T \tilde{x}$ is the ith principal component of $\tilde{x}$.

Proof By (11.107)

$$\arg \min_{\substack{\{b_1,\ldots,b_k\} \\ ||b_i||_2=1\ 1\leq i\leq k \\ b_i \perp b_j \text{ for } i\neq j}} \operatorname{E}\left[\left|\left|\tilde{x} - \underset{b_1,\ldots,b_k}{\operatorname{approx}}(\tilde{x})\right|\right|_2^2\right] = \arg \max_{\substack{\{b_1,\ldots,b_k\} \\ ||b_i||_2=1\ 1\leq i\leq k \\ b_i \perp b_j \text{ for } i\neq j}} \sum_{i=1}^{k} \operatorname{Var}\left[b_i^T \tilde{x}\right],$$

so to prove the result, we need to show that the first k principal directions maximize the sum of the variances. We prove this by induction on k. The base case $k := 1$ follows immediately from (11.61) in Theorem 11.20, which implies

$$u_1 = \arg \max_{||b||_2=1} \operatorname{Var}[b^T \tilde{x}]. \tag{11.111}$$

To complete the proof, we establish that if the induction hypothesis

$$\{u_1, \ldots, u_{k-1}\} = \arg \max_{\substack{\{b_1,\ldots,b_{k-1}\} \\ ||b_i||_2=1, 1\leq i\leq k-1 \\ b_i \perp b_j, i\neq j}} \sum_{i=1}^{k-1} \operatorname{Var}\left[b_i^T \tilde{x}\right] \tag{11.112}$$

holds, then

$$\{u_1, \ldots, u_k\} = \arg \max_{\substack{\{b_1,\ldots,b_k\} \\ ||b_i||_2=1, 1\leq i\leq k \\ b_i \perp b_j, i\neq j}} \sum_{i=1}^{k} \operatorname{Var}\left[b_i^T \tilde{x}\right]. \tag{11.113}$$

We now set b_1,…, b_k to be an arbitrary fixed set of k orthonormal vectors, and show that they cannot capture more variance than the principal directions u_1, …, u_k if the induction hypothesis holds. Consider the subspace $\mathcal{S} := \operatorname{span}(b_1, \ldots, b_k)$ spanned by b_1, …, b_k. We are interested in the projection of $\tilde{x}$ onto this subspace, because

$$\sum_{i=1}^{k} \operatorname{Var}\left[b_i^T \tilde{x}\right] = \sum_{i=1}^{k} \operatorname{E}\left[\left(b_i^T \tilde{x}\right)^2\right] \tag{11.114}$$

$$= \operatorname{E}\left[\sum_{i=1}^{k} \left(b_i^T \tilde{x}\right)^2\right] \tag{11.115}$$

$$= \operatorname{E}\left[\left|\left|\sum_{i=1}^{k} b_i^T \tilde{x} b_i\right|\right|_2^2\right] \tag{11.116}$$

$$= \operatorname{E}\left[||\mathcal{P}_{\mathcal{S}}\, \tilde{x}||_2^2\right]. \tag{11.117}$$

We can represent this projection using an arbitrary orthonormal basis of $\mathcal{S}$. Indeed, any set of k orthonormal vectors a_1, …, a_k spanning $\mathcal{S}$ satisfy

$$\mathcal{P}_{\mathcal{S}}\, \tilde{x} := \sum_{i=1}^{k} b_i^T \tilde{x} b_i = \sum_{i=1}^{k} a_i^T \tilde{x} a_i. \tag{11.118}$$

The key is to choose the basis wisely. $\mathcal{S}$ has dimension k, so it must contain at least one vector $a_\perp$ that is orthogonal to the first $k-1$ principal directions u_1, u_2,…, u_{k-1}. By (11.63) in Theorem 11.20, the variance in that direction cannot be higher than in the kth principal direction

$$\operatorname{Var}[u_k^T \tilde{x}] \geq \operatorname{Var}[a_\perp^T \tilde{x}]. \tag{11.119}$$

We build our wisely-chosen orthonormal basis $a_1, a_2, \ldots, a_k$ for $\mathcal{S}$ setting $a_k := a_\perp$ (we can construct such a basis via the Gram-Schmidt process, starting with $a_\perp$). By the induction hypothesis,

$$\sum_{i=1}^{k-1} \operatorname{Var}\left[u_i^T \tilde{x}\right] \geq \sum_{i=1}^{k-1} \operatorname{Var}\left[a_i^T \tilde{x}\right]. \tag{11.120}$$

Combining (11.120) and (11.119) yields

$$\sum_{i=1}^{k} \operatorname{Var}\left[u_i^T \tilde{x}\right] \geq \sum_{i=1}^{k} \operatorname{Var}\left[a_i^T \tilde{x}\right] \tag{11.121}$$

$$= \operatorname{E}\left[||\mathcal{P}_{\mathcal{S}}\, \tilde{x}||_2^2\right] \tag{11.122}$$

$$= \sum_{i=1}^{k} \operatorname{Var}\left[b_i^T \tilde{x}\right]. \tag{11.123}$$

Since this holds for any choice of b_1,…, b_k, the proof is complete. ■

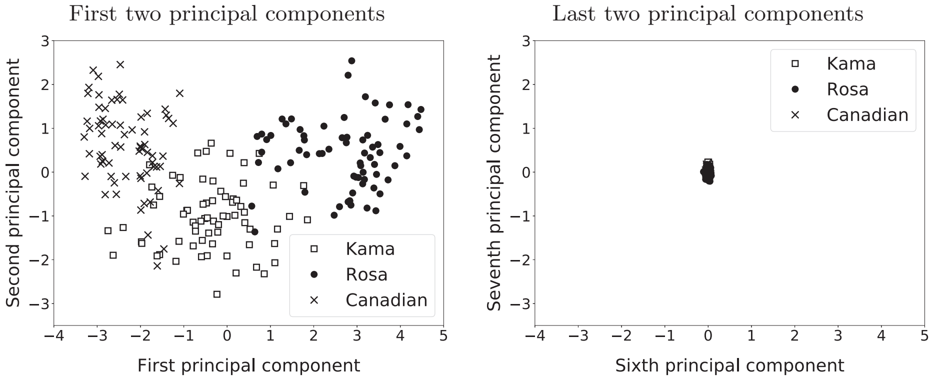


Figure 11.12 Dimensionality reduction for visualization. The left graph shows a scatterplot of the two first principal components of the data in Example 11.28. Each marker indicates a different variety of wheat. The low-dimensional representation makes it possible to visualize meaningful structure; the data points corresponding to each variety cluster together. The right graph shows a scatterplot of the two last principal components, which capture very little variance.

Example 11.28 (Dimensionality reduction for visualization)**.** In order to visualize a dataset, we need to embed the data in two or three dimensions, while preserving their structure as much as possible. In this example, we consider Dataset 21, which describes the geometry of wheat seeds. Each data point consists of seven features: area, perimeter, compactness, length of kernel, width of kernel, asymmetry coefficient and length of kernel groove. The features have different units, so we normalize each of them, dividing by the corresponding sample standard deviation. The seeds belong to three different varieties of wheat: Kama, Rosa and Canadian.

Motivated by Theorem 11.27, we reduce the dimensionality of the data via PCA down to two dimensions in order to visualize them. The left plot of Figure 11.12 shows a scatterplot of the two first principal components of the data. The three varieties of wheat form three distinct clusters. This suggests that the two-dimensional representation preserves meaningful structure, which could be useful for downstream tasks such as clustering or classification. The right plot of the figure shows a scatterplot of the two last principal components. The corresponding principal directions are the directions of least variance, so the points are all on top of each other.

. . . .

Example 11.29 (Faces: Dimensionality reduction via PCA)**.** Figure 11.13 shows the result of representing one of the faces from Dataset 20 using its first five principal components $w_1, \ldots, w_5$. To visualize the representation, we project it onto the image space using the corresponding principal directions $u_1, \ldots, u_5$ and

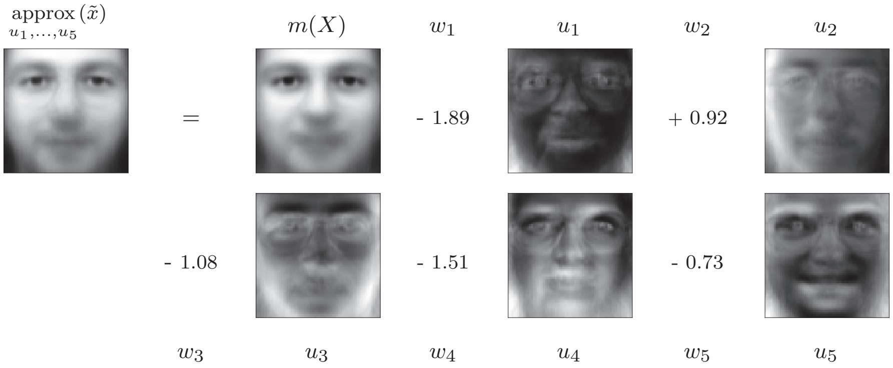


Figure 11.13 Dimensionality reduction of faces based on PCA. Visualization of a five-dimensional representation of a face from Dataset 20 obtained via PCA. To visualize the representation, we project it onto image space by summing the sample mean and the first five principal directions weighted by the corresponding principal component, as described by (11.124). This allows us to see what characteristics of the original image are preserved by the first five principal components (and the mean).

add back the sample mean $m(X)$,

$$\underset{u_1,\ldots,u_5}{\text{approx}}(x_i) := m(X) + \sum_{j=1}^{5} w_j[i]u_j, \quad w_j[i] := u_j^T \operatorname{ct}(x_i), \tag{11.124}$$

where x_i is the chosen face, and X is the whole dataset. Figure 11.14 shows visualizations of the representation for increasing dimensionalities. As suggested by the visualization of the principal directions in Figure 11.9, the first principal directions capture coarse-level characteristics, so the lower-dimensional approximations are quite blurry. In this case, dimensionality reduction via PCA is optimal in terms of preserving the variance in the dataset, but it loses fine-scale details which may be crucial for downstream tasks such as classification.

Example 11.30 (Nearest neighbors in principal-component space)**.** PCA-based dimensionality reduction is often used as a preprocessing step in data science. In this example, we combine it with the nearest-neighbor method to perform face classification. Assume that we have access to a training set of n pairs of data encoded as vectors in $\mathbb{R}^d$ along with their corresponding labels: $\{x_1, y_1\}$, ..., $\{x_n, y_n\}$. To classify a new data point x_{test} using the nearest-neighbor method, we find the closest element of the training set,

$$i^* := \arg \min_{1 \leq i \leq n} ||x_{\text{test}} - x_i||_2 \,, \tag{11.125}$$

and assign the corresponding label y_{i^*} to x_{test}.

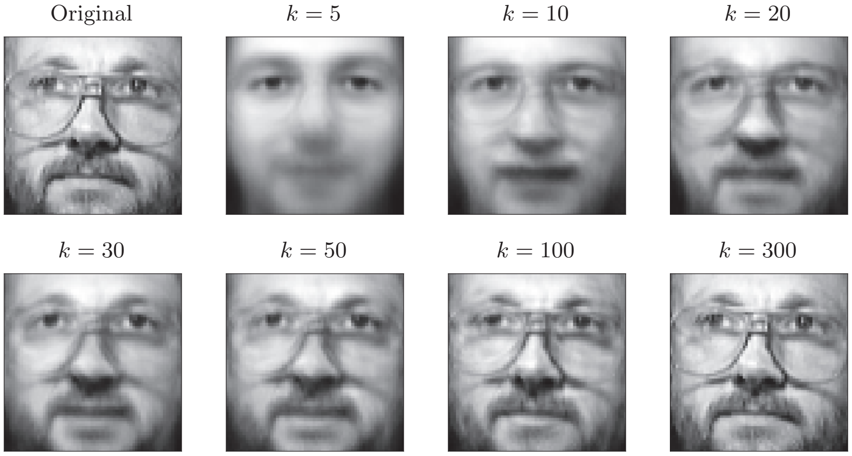


Figure 11.14 Low-dimensional approximations of face data. The images show k-dimensional approximations of a face from Dataset 20 obtained via PCA (as described in (11.124) and Figure 11.13), for different values of k. The approximation is very blurry for small k, because the principal directions mainly capture coarse-level structure. As k increases, the approximation improves.

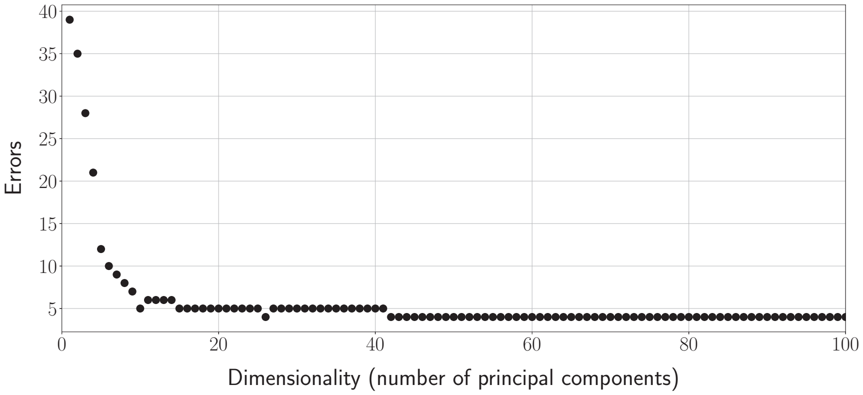


Figure 11.15 Classification in a space of reduced dimensionality. The plot shows the test error of the reduced-dimensionality nearest-neighbor classification approach in Example 11.30, for different values of the dimensionality k. The performance for $k \geq 9$ is very close to the performance achieved in the original space with full dimensionality.

Every time we classify a new point, we need to compute n distances in a d-dimensional space, so the overall computational cost is $\mathcal{O}(nd)$. This cost can be alleviated via dimensionality reduction, representing each point in the training data by its first k principal components. To classify a test data point, we:

1 Center the test data point by subtracting the sample mean of the training data.
2 Compute the inner product of the centered test data point with the first k principal directions of the training data in order to obtain a k-dimensional representation w_{test}.
3 Find the nearest neighbor in the reduced space:

$$i^*_{[k]} := \arg \min_{1 \leq i \leq n} \left|\left| w_{\text{test}} - w_{[1:k]}[i] \right|\right|_2 , \tag{11.126}$$

where $w_{[1:k]}[i]$ denotes the k first principal components of the ith training data point.

The cost of the searching for the closest data point decreases to $\mathcal{O}(nk)$ due to the reduction in dimensionality. Computing the eigendecomposition of the sample covariance matrix is costly, but this only needs to be done once, before processing the test data points.

We apply this approach to Dataset 20. The training set consists of 360 64×64 images taken from 40 different subjects (9 per subject). The test set consists of an image of each subject, which is different from the ones in the training set. Without dimensionality reduction, the nearest-neighbor method classifies 36 of the 40 subjects correctly. Figure 11.15 shows the results when we apply the method after performing dimensionality reduction. At very low dimensions, the number of errors is high, but as we increase the value of the dimensionality k, the accuracy improves; for $k \geq 9$, it is very close to the performance without dimensionality reduction (it matches it at $k := 42$). Figure 11.16 shows some test examples along with their nearest neighbors in the k-dimensional space.

11.6 Low-Rank Models

Up to now, we have mostly considered datasets where each data point is associated to a single entity. For instance, in Example 11.26 each data point corresponds to a person, and in Example 11.28 each data point corresponds to a seed. In this section, we explain how to model datasets where the data are associated to *two* different entities. We represent such data as entries in a matrix D. For each entry $D[i, j]$, the column i and the row j indicate the entities associated to the corresponding data point. In recommender systems, $D[i, j]$ could be the rating given to product j by user i. In computational genomics, $D[i, j]$ could be the expression level of gene i in cell j. In climatology, $D[i, j]$ could be the temperature measured in location i at time j.

In order to analyze and manipulate matrix-valued data, it is often very useful to represent each data point in terms of a small number of factors. This can be achieved by fitting the data using a low-rank approximation. Let us consider a matrix of movie ratings $D \in \mathbb{R}^{n_1 \times n_2}$, where the entry $D[i, j]$ is the rating given by

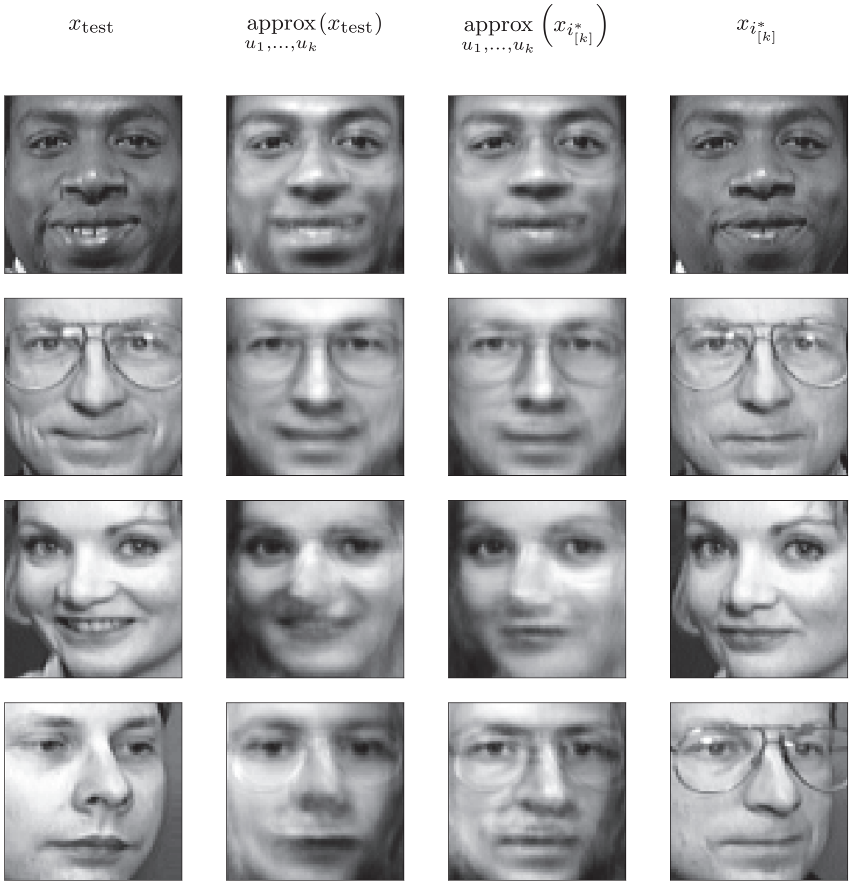

x_{test}
$\underset{u_1,\ldots,u_k}{\text{approx}}(x_{\text{test}})$
$\underset{u_1,\ldots,u_k}{\text{approx}}\left(x_{i^*_{[k]}}\right)$
$x_{i^*_{[k]}}$

Figure 11.16 Face classification from low-dimensional representations. Results of nearest-neighbor classification performed after PCA-based dimensionality reduction with $k := 42$ for four of the people in Example 11.30. The first column shows the test image x_{test}. The second column shows the corresponding k-dimensional approximation $\text{approx}_{u_1,\ldots,u_k}(x_{\text{test}})$, computed as described in (11.124) and Figure 11.13. The third column shows the k-dimensional approximation of the image in the training set that is closest to $\text{approx}_{u_1,\ldots,u_k}(x_{\text{test}})$ denoted by $\text{approx}_{u_1,\ldots,u_k}(x_{i^*_{[k]}})$, where $i^*_{[k]}$ is defined in (11.126). The fourth column shows the corresponding training image $x_{i^*_{[k]}}$. The assignments of the first three examples are correct, but the fourth is wrong.

user j to movie i. To model D, we approximate $D[i,j]$ as a sum of r components, where $r < \min\{n_1, n_2\}$. The lth component is the product of a coefficient $a_l[i]$ associated to movie i and a coefficient $b_l[j]$ associated to user j. The resulting

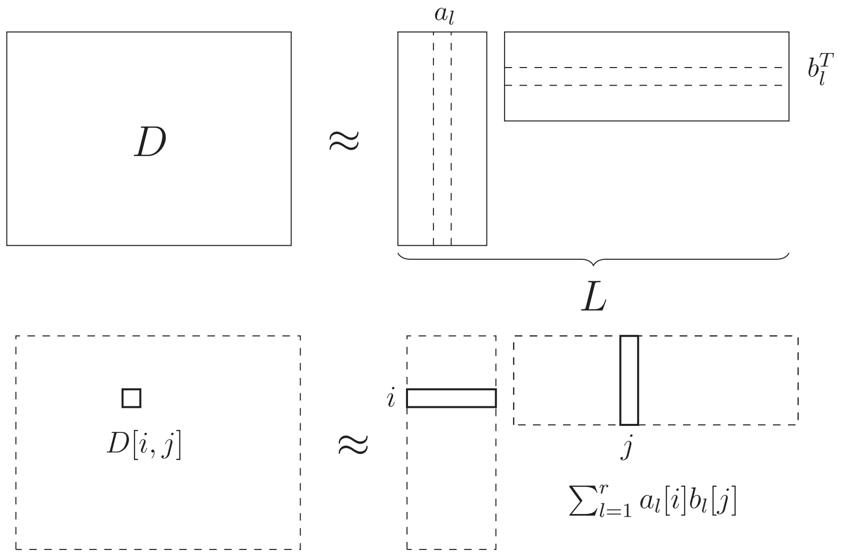


Figure 11.17 Low-rank model. Illustration of a rank-r approximation L to a data matrix $D \in \mathbb{R}^{n_1 \times n_2}$. The parameters of the rank-r model are r n_1-dimensional vectors $a_1, \ldots, a_r$ spanning the columns of L, and r n_2-dimensional vectors $b_1, \ldots, b_r$ spanning the rows of L. To fit the low-rank model, we approximate each entry $D[i,j]$ of the data matrix by the corresponding entry of the low-rank matrix, which is the sum of the products of the corresponding entries of a_l and b_l for $1 \leq l \leq r$.

approximation is the matrix

$$L[i,j] := \sum_{l=1}^{r} a_l[i]b_l[j], \qquad 1 \leq i \leq n_1, 1 \leq j \leq n_2. \tag{11.127}$$

Figure 11.17 shows a diagram of the matrix and the coefficient vectors.

We can interpret each term in (11.127) as a factor that influences the rating. The coefficient $a_l[i]$ determines whether the association of movie i to factor l is positive (> 0), negative (< 0) or negligible (≈ 0). Similarly, the coefficient $b_l[j]$ determines whether the association of user j to factor l is positive (> 0), negative (< 0) or negligible (≈ 0). The model is *bilinear*, because the approximation is a bilinear function of the coefficients: if the user coefficients are fixed, the model is linear in the movie coefficients; if the movie coefficients are fixed, the model is linear in the user coefficients.

The rank of the matrix L in (11.127) is equal to r. Its columns are spanned by the r n_1-dimensional vectors $a_1, a_2, \ldots, a_r$ and its rows are spanned by the r n_2-dimensional vectors $b_1, b_2, \ldots, b_r$ (see Figure 11.17). Since r is typically chosen to be much smaller than the rank of the data matrix D, the resulting model is known as a *low-rank model*. In Section 11.6.1 we explore the connection between low-rank models and principal-component analysis. Section 11.6.2 introduces the singular-value decomposition (SVD), a fundamental tool from linear algebra that decomposes a matrix into rank-1 components. In Section 11.6.3 we explain how to fit low-rank models using the SVD, illustrating the approach with applications to movie ratings and weather data. Finally, Section 11.6.4 shows that this SVD-based approach yields an optimal low-rank approximation.

11.6.1 Low-Rank Models And Principal Component Analysis

In the low-rank model (11.127), each column of L is a linear combination of r basis vectors. In order to fit the model from data, we need to obtain these basis vectors and the corresponding coefficients of the linear combination. This can be achieved via principal component analysis (PCA). We interpret the columns of the data matrix $D \in \mathbb{R}^{n_1 \times n_2}$ as a set of n_1-dimensional data points, and reduce their dimensionality from n_1 to r via PCA, as explained in Section 11.5.

Let us denote each of the n_2 columns by $D[:, j] \in \mathbb{R}^{n_1}$, $1 \leq j \leq n_2$, following Python notation. In order to apply PCA, we compute the eigendecomposition of the sample covariance matrix of the columns,

$$\Sigma_{\text{cols}} := \frac{1}{n_2} \sum_{j=1}^{n_2} D[:, j] D[:, j]^T \tag{11.128}$$

$$= \frac{1}{n_2} D D^T. \tag{11.129}$$

Let u_1, u_2, $\ldots$, u_r be the eigenvectors of Σ_{cols} corresponding to the r largest eigenvalues. The corresponding principal components are

$$w_l[j] := u_l^T D[:, j], \quad 1 \leq j \leq n_2, \tag{11.130}$$

assuming the data are centered. This yields a rank-r model, which we call $L_{\text{PCA-cols}}$, because it is obtained by applying PCA to the columns of the data matrix:

$$L_{\text{PCA-cols}}[i, j] := \sum_{l=1}^{r} u_l[i] w_l[j], \qquad 1 \leq i \leq n_1, 1 \leq j \leq n_2. \tag{11.131}$$

The low-rank approximation is depicted in the top diagram of Figure 11.18. By Theorem 11.27, the model is optimal from the point of view of preserving the mean squared ℓ_2 norm of the columns of D.

You might be wondering why we interpret the *columns* as data points, and not the *rows*. Indeed, we can apply the same reasoning to the rows. Let us denote each of the n_1 rows by $D[i, :]$, $1 \leq i \leq n_1$, again following Python notation. We apply PCA to the rows by computing the eigendecomposition of their sample covariance matrix,

$$\Sigma_{\text{rows}} := \frac{1}{n_1} \sum_{i=1}^{n_1} D[i, :]^T D[i, :] \tag{11.132}$$

$$= \frac{1}{n_1} D^T D. \tag{11.133}$$

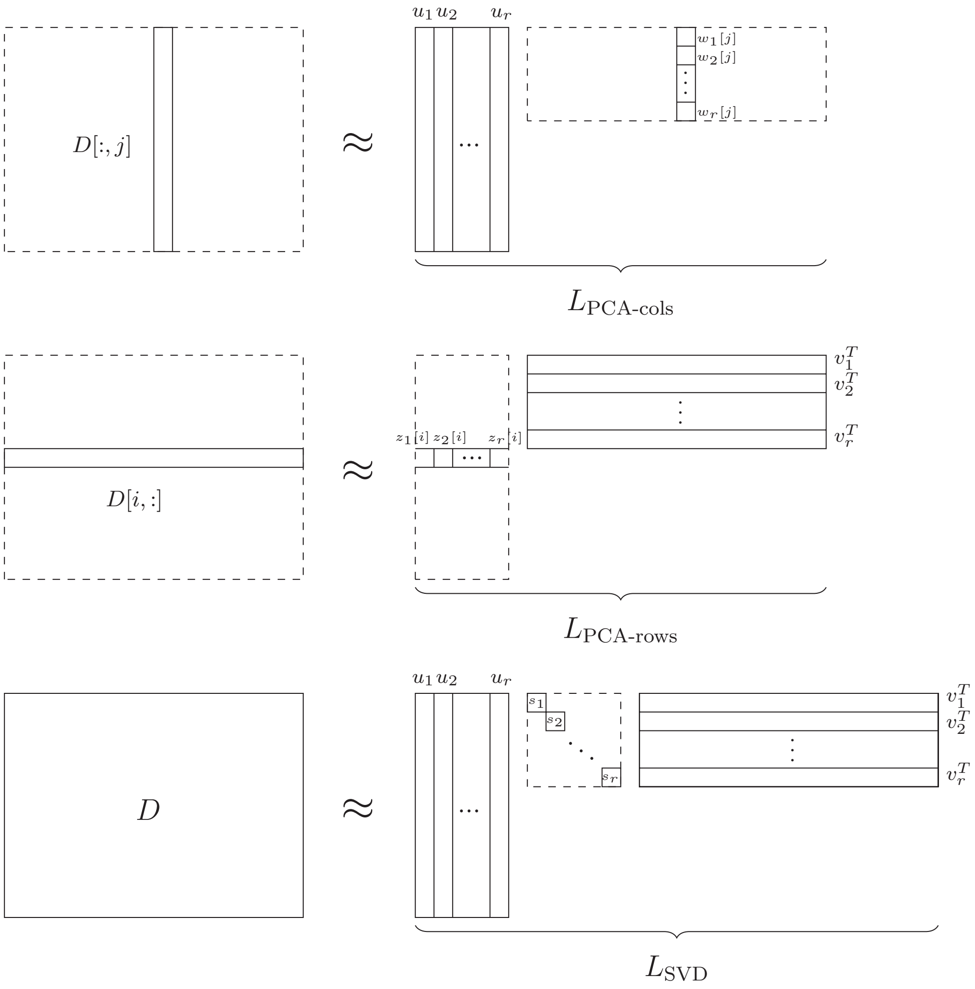


Figure 11.18 Low-rank models computed via PCA and truncated SVD. Illustration of the PCA-based low-rank approximations described in Section 11.6.1. $L_{\text{PCA-cols}}$ (top) is obtained by performing dimensionality reduction of the columns of the data matrix D. $L_{\text{PCA-rows}}$ (center) is obtained by performing dimensionality reduction of the rows of D. Theorem 11.35 shows that these approximations are the same, and also equal to the low-rank approximation L_{SVD} (bottom), obtained by truncating the singular-value-decomposition of D, as explained in Section 11.6.3.

Let v_1, v_2, ..., v_r be the eigenvectors of Σ_{rows} associated to the r largest eigenvalues. The corresponding principal components for each movie are

$$z_l[i] := D[i,:]v_l, \tag{11.134}$$

assuming the data are centered. This yields a rank-r model, which we call $L_{\text{PCA-rows}}$

because it is obtained by applying PCA to the rows of the data matrix:

$$L_{\text{PCA-rows}}[i,j] := \sum_{l=1}^{r} z_l[i] v_l[j], \qquad 1 \leq i \leq n_1, 1 \leq j \leq n_2. \tag{11.135}$$

The model is depicted in the center diagram of Figure 11.18. By Theorem 11.27, this model is optimal from the point of view of preserving the mean squared ℓ_2 norm of the rows.

In summary, there are two alternative ways of fitting the low-rank model: applying PCA to the columns or to the rows. *Which one is better?* It turns out that they are completely equivalent! Section 11.6.3 shows that this is the case using the singular value decomposition, which is the subject of the following section.

11.6.2 The Singular-Value Decomposition

The singular-value decomposition (SVD) of a matrix is a fundamental tool in linear algebra and applied mathematics. It decomposes a matrix into the product of a matrix with orthonormal columns, a diagonal matrix, and a matrix with orthonormal rows. The following theorem shows that every matrix has an SVD.

Theorem 11.31 (Singular-value decomposition)**.** *Any matrix $A \in R^{n_1 \times n_2}$, $n_1 \leq n_2$, with rank t has a singular-value decomposition (SVD) of the form*

$$A = \underbrace{\begin{bmatrix} u_1 & u_2 & \cdots & u_{n_1} \end{bmatrix}}_{U} \underbrace{\begin{bmatrix} s_1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & s_2 & \cdots & 0 & 0 & \cdots & 0 \\ \cdots & \cdots & \ddots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & s_t & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \ddots & \cdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \end{bmatrix}}_{S} \underbrace{\begin{bmatrix} v_1 & v_2 & \cdots & v_{n_1} \end{bmatrix}^T}_{V^T},$$

where the singular values $s_1 \geq s_2 \geq \cdots \geq s_t$ are positive real numbers, the left singular vectors u_1, u_2, $\ldots u_{n_1} \in \mathbb{R}^{n_1}$ are orthonormal, and the right singular vectors v_1, v_2, $\ldots v_{n_1} \in \mathbb{R}^{n_2}$ are also orthonormal.

Proof By the spectral theorem, the symmetric matrix $M := AA^T \in \mathbb{R}^{n_1 \times n_1}$ has n_1 orthonormal eigenvectors $u_1, \ldots, u_{n_1}$. The corresponding eigenvalues $\lambda_1, \ldots, \lambda_{n_1}$ are nonnegative, since for $1 \leq j \leq n_1$,

$$\begin{aligned} \left|\left|A^T u_j\right|\right|_2^2 &= u_j^T A A^T u_j && (11.136) \\ &= \lambda_j u_j^T u_j && (11.137) \\ &= \lambda_j. && (11.138) \end{aligned}$$

The number of nonzero eigenvalues is equal to the rank t of A, because A and AA^T have the same rank. For $1 \leq l \leq t$, we define $s_l := \sqrt{\lambda_l}$ and

$$v_l := \frac{1}{s_l} A^T u_l. \tag{11.139}$$

These vectors have unit ℓ_2 norm,

$$||v_l||_2^2 = \frac{1}{s_l^2} u_l^T A A^T u_l \tag{11.140}$$

$$= \frac{\lambda_l}{\lambda_l} u_l^T u_l = 1, \tag{11.141}$$

and are orthogonal to each other if $l \neq k$ because $u_l^T u_k = 0$, which implies

$$v_l^T v_k = \frac{u_l^T A A^T u_k}{s_l s_k} \tag{11.142}$$

$$= \frac{\lambda_k u_l^T u_k}{s_l s_k} = 0. \tag{11.143}$$

We now define the matrices

$$U := \begin{bmatrix} u_1 & u_2 & \cdots & u_{n_1} \end{bmatrix}, \tag{11.144}$$

$$V := \begin{bmatrix} v_1 & v_2 & \cdots & v_{n_1} \end{bmatrix}, \tag{11.145}$$

where we choose v_{t+1}, v_{t+2}, ..., v_{n_1} to be an orthonormal set of vectors, which are also orthogonal to v_1, ..., v_t. By (11.139),

$$U^T A = S V^T. \tag{11.146}$$

Notice that U is an orthogonal or unitary matrix, because it is square and has orthonormal columns. As a result, UU^T is equal to the identity, so

$$A = USV^T. \tag{11.147}$$

■

The rank of a matrix is equal to the number of nonzero singular values. If a matrix only has t nonzero singular values, then its columns are all linear combinations of the corresponding t left singular vectors. In fact, the SVD can be interpreted as a decomposition of the matrix into rank-1 matrices. We can write the SVD of a matrix $D \in \mathbb{R}^{n_1 \times n_2}$, $n_1 \leq n_2$, in the following form:

$$D = \sum_{l=1}^{n_1} s_l K_l, \qquad K_l := u_l v_l^T. \tag{11.148}$$

The rank of the matrix K_l is one because it is the outer product of the left singular vector u_l and the right singular vector v_l, so its columns are scaled copies of u_l and its rows are scaled copies of v_l. It turns out that these rank-1 matrices are orthogonal and have unit norm. To make this precise, we define an inner product between matrices.

Definition 11.32 (Frobenius inner product and norm)**.** *The Frobenius inner product between two matrices $A \in \mathbb{R}^{n_1 \times n_2}$ and $B \in \mathbb{R}^{n_1 \times n_2}$ equals*

$$\langle A, B \rangle := \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} A[i,j] B[i,j] \tag{11.149}$$

$$= \operatorname{Trace}\left(A^T B\right), \tag{11.150}$$

where the trace of a matrix is the sum of its diagonal entries. The norm associated to the Frobenius inner product is known as the Frobenius norm,

$$||A||_{\mathrm{F}} := \sqrt{\langle A, A \rangle} \tag{11.151}$$

$$= \sqrt{\sum_{i=1}^{n_1} \sum_{j=1}^{n_2} A[i,j]^2} \tag{11.152}$$

$$= \sqrt{\operatorname{Trace}\left(A^T A\right)}. \tag{11.153}$$

If we vectorize the matrices (by concatenating the columns into a vector of length $n_1 n_2$), then the Frobenius inner product and norm are equivalent to the standard Euclidean inner product and to the ℓ_2 norm, respectively.

The rank-1 matrices K_1, …, K_{n_1} in (11.148) form an orthonormal set with respect to the Frobenius inner product and norm.

Theorem 11.33 (Orthonormality of rank-1 matrices)**.** *For $n_1 \leq n_2$, let u_1, …, u_{n_1} be an orthonormal set of vectors in $\mathbb{R}^{n_1}$ and let v_1, …, v_{n_1} be an orthonormal set of vectors in $\mathbb{R}^{n_2}$. The rank-1 matrices K_1, …, K_{n_1}, where*

$$K_l := u_l v_l^T, \qquad 1 \leq l \leq n_1, \tag{11.154}$$

form an orthonormal set with respect to the Frobenius inner product and norm.

Proof For any matrices $A \in \mathbb{R}^{n_1 \times n_2}$ and $B \in \mathbb{R}^{n_1 \times n_2}$ we have $\operatorname{Trace}(A^T B) = \operatorname{Trace}(B A^T)$ (you can check by writing out both expressions). As a result, for $1 \leq l, k \leq n$,

$$||K_l||_{\mathrm{F}}^2 = \left|\left|u_l v_l^T\right|\right|_{\mathrm{F}}^2 = \operatorname{Trace}\left(v_l u_l^T u_l v_l^T\right) \tag{11.155}$$

$$= \operatorname{Trace}\left(v_l^T v_l u_l^T u_l\right) = 1 \tag{11.156}$$

and if $l \neq k$,

$$\langle K_l, K_k \rangle = \left\langle u_l v_l^T, u_k v_k^T \right\rangle = \operatorname{Trace}\left(v_l u_l^T u_k v_k^T\right) \tag{11.157}$$

$$= \operatorname{Trace}\left(v_k^T v_l u_l^T u_k\right) = 0. \tag{11.158}$$

■

In the SVD decomposition (11.148), each rank-1 matrix K_l with unit Frobenius norm is weighted by the corresponding singular value s_l. The following lemma shows that the sum of squared singular values is equal to the squared Frobenius norm of the whole matrix.

Theorem 11.34 (Frobenius norm and singular values)**.** *For any matrix $A \in \mathbb{R}^{n_1 \times n_2}$, with singular values $s_1, \ldots, s_{\min\{n_1,n_2\}}$*

$$||A||_{\mathrm{F}} = \sqrt{\sum_{l=1}^{\min\{n_1,n_2\}} s_l^2}. \tag{11.159}$$

Proof Assume $n_1 \leq n_2$ (for $n_1 > n_2$, the same argument applies to the transpose of A). We denote the SVD of A by $\sum_{i=1}^{n_1} s_l K_l$, where $K_l := u_l v_l^T$. Theorem 11.33 establishes that the rank-1 matrices K_l are orthogonal, so by the Pythagorean theorem,

$$\begin{aligned} ||A||_{\mathrm{F}}^2 &= \left|\left|\sum_{i=1}^{n_1} s_l K_l\right|\right|_{\mathrm{F}}^2 && (11.160)\\ &= \sum_{i=1}^{n_1} \left|\left|s_l K_l^T\right|\right|_{\mathrm{F}}^2 && (11.161)\\ &= \sum_{i=1}^{n_1} s_l^2 \left|\left|K_l\right|\right|_{\mathrm{F}}^2 && (11.162)\\ &= \sum_{i=1}^{n_1} s_l^2. && (11.163) \end{aligned}$$

■

In summary, the SVD provides a decomposition in terms of orthogonal rank-1 matrices, where the squared norm of each component is equal to the corresponding squared singular value. This suggests an informal definition of when a matrix is *approximately* rank r: r of its singular values are much larger than the rest. For such matrices, most of the energy (measured in Frobenius norm) is concentrated in the corresponding r rank-1 components.

11.6.3 Truncating The Singular-Value Decomposition

In this section, we show how to leverage the SVD to obtain a low-rank approximation to a matrix. According to our analysis in Section 11.6.2, we can decompose any matrix in rank-1 components weighted by the corresponding singular values. The singular values allow us to quantify the contribution of each component, because the total squared Frobenius norm of the matrix equals the sum of the squared singular values by Theorem 11.34. This motivates the following scheme to obtain a rank-r approximation: compute the SVD of the matrix and only keep the rank-1 components corresponding to the r largest singular values. The

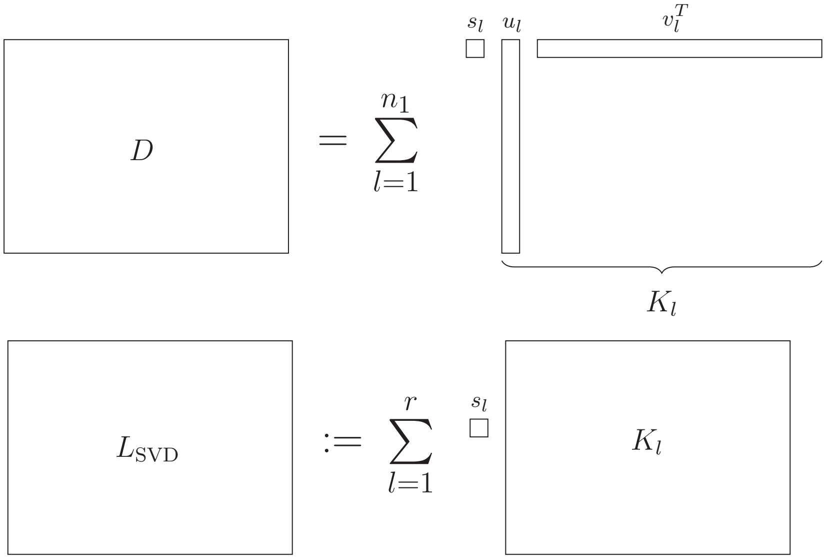


Figure 11.19 Low-rank approximation via SVD truncation. As explained in Section 11.6.3, the SVD of a matrix can be interpreted as a sum of the rank-1 matrices K_l depicted at the top, weighted by the corresponding singular values. In order to obtain a rank-r approximation, we truncate the sum, retaining the components corresponding to the top r singulaxr values.

resulting truncated-SVD approximation equals

$$L_{\text{SVD}} := \sum_{l=1}^{r} s_l K_l \tag{11.164}$$

$$= \sum_{l=1}^{r} s_l u_l v_l^T, \tag{11.165}$$

as depicted in Figures 11.18 and 11.19. Theorem 11.35 shows that L_{SVD} is equivalent to the PCA-based low-rank approximations in Section 11.6.1 (also shown in Figure 11.18). By Theorem 11.34, the approximation error is

$$||D - L_{\text{SVD}}||_{\text{F}}^2 = \left|\left| \sum_{l=r+1}^{n_1} s_l u_l v_l^T \right|\right|_{\text{F}}^2 \tag{11.166}$$

$$= \sum_{l=r+1}^{n_1} s_l^2, \tag{11.167}$$

because $s_{r+1}, \ldots, s_{r+n_1}$ are the singular values of $D - L_{\text{SVD}}$ (we assume $n_1 \leq n_2$ without loss of generality). Section 11.6.4 establishes that this is optimal. No other rank-r approximation can have a smaller Frobenius-norm error.

Theorem 11.35 (PCA and SVD yield the same low-rank approximation)**.** *Let*

D be any real valued $n_1 \times n_2$ matrix, which has been centered by subtracting its mean entry, and let r be a positive integer smaller than n_1 and n_2. We denote the column-wise PCA rank-r approximation of D by

$$L_{\text{PCA-cols}} := \sum_{l=1}^{r} u_l w_l^T, \tag{11.168}$$

where $w_l[j] := u_l^T D[:,j]$, $1 \leq j \leq n_2$, and u_l, $1 \leq l \leq r$, is the eigenvector of the sample covariance matrix $\Sigma_{\text{cols}} := \frac{1}{n_2} DD^T$ of the columns with the lth largest eigenvalue. Similarly, we denote the row-wise PCA rank-r approximation of D by

$$L_{\text{PCA-rows}} := \sum_{l=1}^{r} z_l v_l^T, \tag{11.169}$$

where $z_l[i] := D[i,:]v_l$, $1 \leq i \leq n_1$, and v_l, $1 \leq l \leq r$, is the eigenvector of the sample covariance matrix $\Sigma_{\text{rows}} := \frac{1}{n_1} D^T D$ of the rows with the lth largest eigenvalue. The rank-r SVD-based approximation L_{SVD} defined in (11.165) is equal to both $L_{\text{PCA-cols}}$ and $L_{\text{PCA-rows}}$.

Proof Let $D = USV^T$ denote the SVD of D. The column-wise sample covariance matrix equals

$$\begin{aligned} \Sigma_{\text{cols}} &:= \frac{1}{n_2} DD^T && (11.170)\\ &= \frac{1}{n_2} USV^T VSU^T && (11.171)\\ &= U\left(\frac{1}{n_2} S^2\right) U^T, && (11.172) \end{aligned}$$

where S^2 is a diagonal matrix containing the squared singular values of D. Therefore, the r eigenvectors of Σ_{cols} with the largest eigenvalues are equal to the first r left singular vectors of D. As a result,

$$\begin{aligned} w_l[j] &:= u_l^T D[:,j] && (11.173)\\ &= u_l^T \sum_{k=1}^{r} s_k u_k v_k[j] && (11.174)\\ &= s_l v_l[j]. && (11.175) \end{aligned}$$

Consequently,

$$\begin{aligned} L_{\text{PCA-cols}}[i,j] &:= \sum_{l=1}^{r} u_l[i] w_l[j] && (11.176)\\ &= \sum_{l=1}^{r} s_l u_l[i] v_l[j] = L_{\text{SVD}}[i,j]. && (11.177) \end{aligned}$$

The row-wise sample covariance matrix equals

$$\Sigma_{\text{rows}} := \frac{1}{n_1} D^T D \tag{11.178}$$

$$= \frac{1}{n_1} V S U^T U S V^T \tag{11.179}$$

$$= V \left(\frac{1}{n_1} S^2 \right) V^T, \tag{11.180}$$

so the r eigenvectors with the largest eigenvalues are equal to the first r right singular vectors of D. As a result,

$$z_l[i] := D[i,:] v_l \tag{11.181}$$

$$= \left(\sum_{k=1}^{r} s_k u_k[i] v_k^T \right) v_l \tag{11.182}$$

$$= s_l u_l[i], \tag{11.183}$$

which implies

$$L_{\text{PCA-rows}}[i,j] := \sum_{l=1}^{r} z_l[i] v_l[j] \tag{11.184}$$

$$= \sum_{l=1}^{r} s_l u_l[i] v_l[j] = L_{\text{SVD}}[i,j]. \tag{11.185}$$

■

Example 11.36 (Rank-1 model for movie ratings)**.** Bob, Molly, Mary and Larry rate the following six movies from 1 to 5,

$$D := \begin{array}{c} \begin{array}{cccc} \text{Bob} & \text{Molly} & \text{Mary} & \text{Larry} \end{array} \\ \begin{pmatrix} 1 & 1 & 5 & 4 \\ 2 & 1 & 4 & 5 \\ 4 & 5 & 2 & 1 \\ 5 & 4 & 2 & 1 \\ 4 & 5 & 1 & 2 \\ 1 & 2 & 5 & 5 \end{pmatrix} \end{array} \begin{array}{l} \\ \text{The Dark Knight} \\ \text{Spiderman 3} \\ \text{Love Actually} \\ \text{Bridget Jones's Diary} \\ \text{Pretty Woman} \\ \text{Superman 2} \end{array} \tag{11.186}$$

Our goal is to fit a low-rank model to these data using the SVD. A common preprocessing step before fitting a low-rank model is to subtract the sample mean,

$$m(D) := \frac{1}{24} \sum_{i=1}^{6} \sum_{j=1}^{4} D[i,j], \tag{11.187}$$

from each entry to obtain a centered matrix $\text{ct}(D) := D - m(D)$. The SVD of ct (D) equals

$$\operatorname{ct}(D) = USV^T = U \begin{bmatrix} 7.79 & 0 & 0 & 0 \\ 0 & 1.62 & 0 & 0 \\ 0 & 0 & 1.55 & 0 \\ 0 & 0 & 0 & 0.62 \end{bmatrix} V^T. \tag{11.188}$$

The first singular value is much larger than the rest, which suggests that the data can be well approximated by a rank-1 matrix. To compute the approximation, we select the first term of the rank-1 decomposition provided by the SVD and add back the sample mean,

$$L_{\text{SVD}} = m(D)11^T + s_1 u_1 v_1^T \tag{11.189}$$

$$= \begin{array}{c} \\ \left(\begin{array}{cccc} \text{Bob} & \text{Molly} & \text{Mary} & \text{Larry} \\ 1.34\,(1) & 1.19\,(1) & 4.66\,(5) & 4.81\,(4) \\ 1.55\,(2) & 1.42\,(1) & 4.45\,(4) & 4.58\,(5) \\ 4.45\,(4) & 4.58\,(5) & 1.55\,(2) & 1.42\,(1) \\ 4.43\,(5) & 4.56\,(4) & 1.57\,(2) & 1.44\,(1) \\ 4.43\,(4) & 4.56\,(5) & 1.57\,(1) & 1.44\,(2) \\ 1.34\,(1) & 1.19\,(2) & 4.66\,(5) & 4.81\,(5) \end{array}\right) \end{array} \begin{array}{l} \\ \text{The Dark Knight} \\ \text{Spiderman 3} \\ \text{Love Actually} \\ \text{Bridget Jones's Diary} \\ \text{Pretty Woman} \\ \text{Superman 2} \end{array},$$

where 11^T denotes a 6×4 matrix with entries equal to one (this is just to add back the sample mean to each entry). For ease of comparison, the values of D are shown in parenthesis.

The fact that the rank of the matrix is approximately one suggests that there is a single latent factor, which accounts for most of the structure in the data. To interpret this factor, we examine the left and right singular vectors. The first left singular vector is equal to

$$u_1 = \left(\begin{array}{cccccc} \text{D. Knight} & \text{Spiderman 3} & \text{Love Act.} & \text{B.J.'s Diary} & \text{P. Woman} & \text{Superman 2} \\ -0.45 & -0.39 & 0.39 & 0.38 & 0.38 & -0.45 \end{array}\right).$$

The latent factor separates the movies into action movies (negative entries of u_1) and romantic movies (positive entries of u_1). Consequently, the users can be clustered according to which type of movie they prefer using the first right singular vector:

$$v_1 = \left(\begin{array}{cccc} \text{Bob} & \text{Molly} & \text{Mary} & \text{Larry} \\ 0.48 & 0.52 & -0.48 & -0.52 \end{array}\right). \tag{11.190}$$

Negative entries indicate users that like action movies but hate romantic movies (Mary and Larry), whereas positive entries indicate the contrary (Bob and Molly).
....

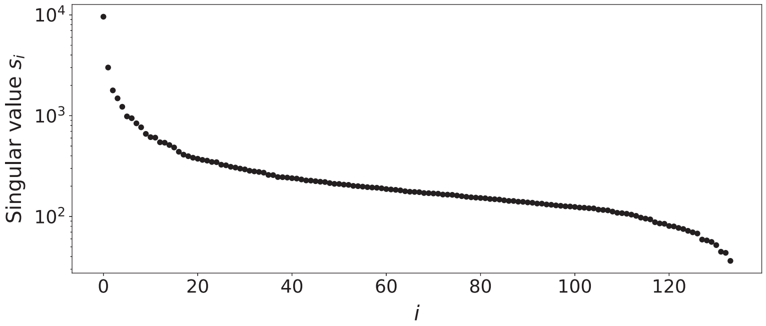
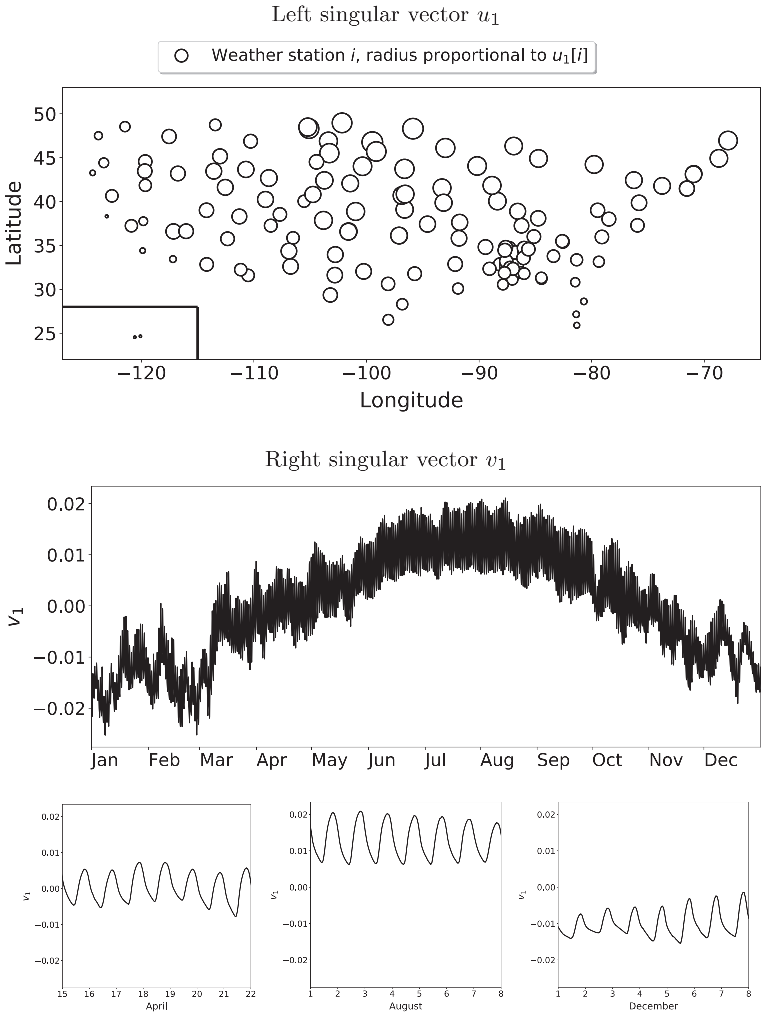


Figure 11.20 Singular values of temperature data. Singular values of the data in Example 11.37. The two largest singular values are much larger than the rest, suggesting that the matrix can be approximated by a rank-2 model.

Example 11.37 (SVD analysis of temperature data)**.** In this example we build a low-rank model to analyze hourly temperatures measured at 134 weather stations in the United States in 2015, extracted from Dataset 9. We represent the data as a matrix with 134 rows, each corresponding to a weather station, and $24 \cdot 365 = 8{,}760$ columns, one for each hour in the year. We compute the SVD after subtracting the mean temperature averaged over all the data. The singular values are shown in Figure 11.20. The two largest singular values are much larger than the rest, suggesting that the data can be approximated by a rank-2 matrix.

The first component of the low-rank model clearly captures temperature seasonality, as depicted in Figure 11.21. The first right singular vector v_1 follows a yearly pattern where the summer is warmer than the winter, and a daily pattern where the day is warmer than the night (see the zoomed-in graphs at the bottom). Each entry of the first left singular vector u_1 represents the contribution of the first component to a weather station. In the top graph of Figure 11.21, we depict the entries as markers situated at the geographic location of each station. The radius of each marker is proportional to the magnitude of the entry. The contribution is larger in the interior of the United States, where the weather is very seasonal, and smaller for locations with less marked annual seasonality, such as Hawaii, the West Coast, or Florida.

The second rank-1 component, depicted in Figure 11.22, seems more mysterious at first glance. However, looking closely at the right singular vector v_2 (see the graphs at the bottom of Figure 11.22) reveals that this component is a *correction* to the daily pattern. Adding v_2 (scaled by the singular value s_2 and the corresponding entry of u_2) shifts the daily temperature pattern to the left. This shift accounts for the fact that the sun rises earlier in the East, and later in the West. For each station, the corresponding entry of the second left singular vector u_2 determines to what extent the pattern is shifted. We expect a larger shift for stations with Eastern locations. The top graph in Figure 11.22 confirms that this



Left singular vector u_1

Right singular vector v_1

Figure 11.21 SVD analysis automatically reveals weather seasonality. The graph at the top shows the first left singular vector u_1 of the temperature dataset from Example 11.37, after centering by subtracting the mean temperature. The entries are plotted at the location of the corresponding weather station; the radius of each marker is proportional to the magnitude of the entry (all entries are positive). The right singular vector v_1, depicted below, reveals that the first rank-1 component of the SVD captures annual seasonality (see the graph at the center) and daily periodicity (see the zoomed-in graphs below).

is the case: the entries of u_2 increase from West to East, and are in fact negative for stations in the West.

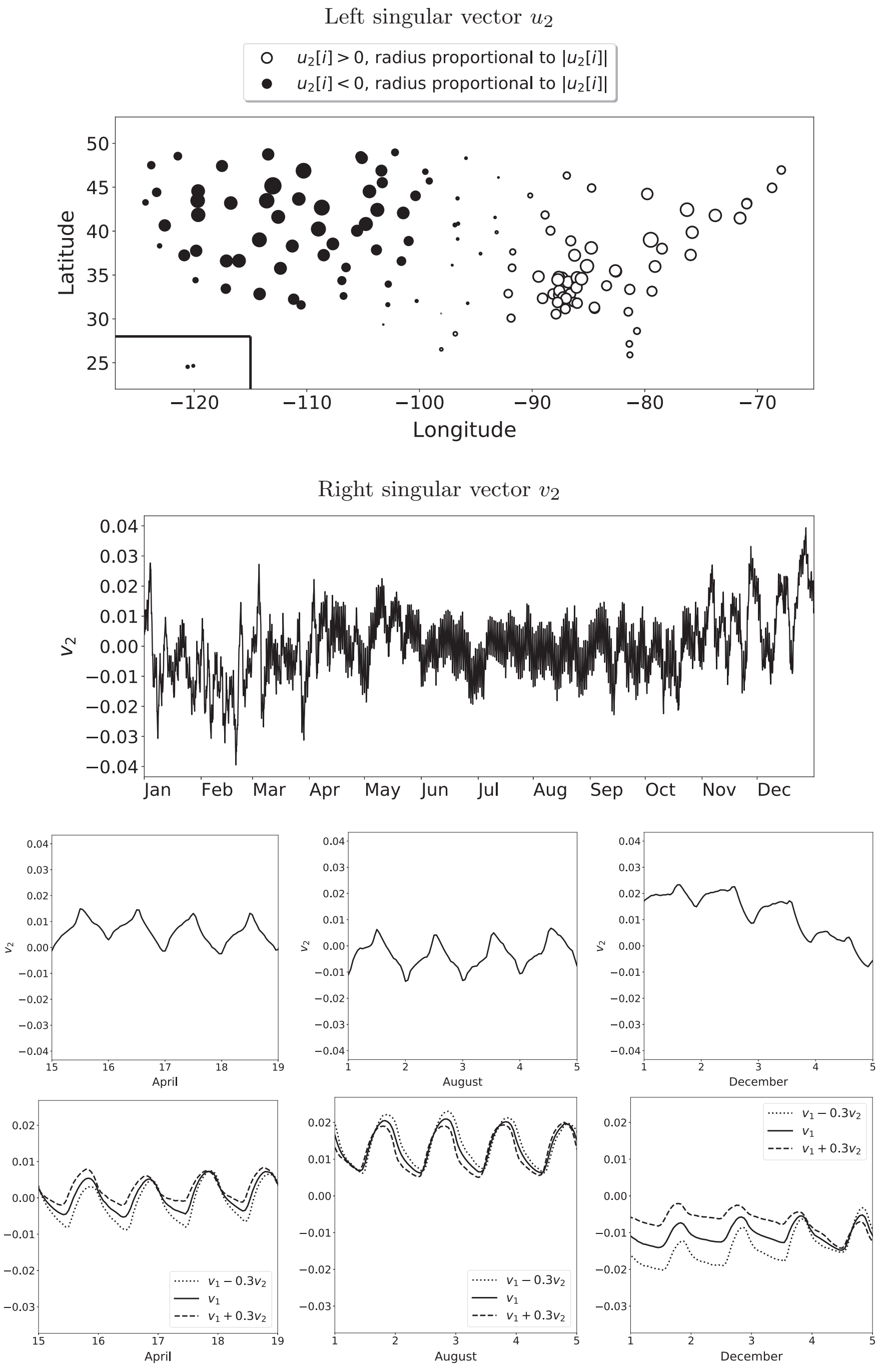


Figure 11.22 SVD analysis reveals longitude-dependent shift. The top graph shows the second left singular vector u_2 of the temperature dataset from Example 11.37. The entries are plotted at the location of the corresponding weather station. Black represents negative entries and white positive entries, and the radius of each marker is proportional to the magnitude. The right singular vector v_2, depicted below, reveals that the second rank-1 component of the SVD shifts the daily pattern (see Figure 11.21) to the left or right to account for the sun rising earlier in the East than in the West (see the zoomed-in graphs at the bottom). This is why the coefficients in the top graph change from negative to positive as we go from West to East.

11.6.4 Optimal Low-Rank Matrix Estimation

In this section, we prove that the rank-r approximation obtained by truncating the SVD in (11.165) is optimal with respect to approximation error in Frobenius norm. The following lemma is key to the proof.

Lemma 11.38. *If the column spaces of any pair of matrices $A, B \in \mathbb{R}^{n_1 \times n_2}$ are orthogonal*

$$||A+B||_{\mathrm{F}}^2 = ||A||_{\mathrm{F}}^2 + ||B||_{\mathrm{F}}^2 . \tag{11.191}$$

Proof The matrices are orthogonal with respect to the Frobenius product, which can be expressed as a sum of products between the columns of A and B that all equal zero:

$$\langle A, B\rangle := \operatorname{Trace}\left(A^T B\right) \tag{11.192}$$

$$= \sum_{i=1}^{n} \langle A[:,i], B[:,i]\rangle = 0, \tag{11.193}$$

where $A[:,i]$ and $B[:,i]$ denote the ith column of A and B respectively. The result then follows directly from the Pythagorean theorem. ■

Theorem 11.39 (Optimal rank-r approximation)**.** *Let USV^T be the SVD of a matrix $D \in \mathbb{R}^{n_1 \times n_2}$. For any positive integer $r \leq \min\{n_1, n_2\}$, truncating the SVD achieves the best rank-r approximation of D in terms of Frobenius-norm error:*

$$L_{\mathrm{SVD}} := \sum_{l=1}^{r} s_l u_l v_l^T = \arg \min_{\operatorname{rank}(L)=r} ||D-L||_{\mathrm{F}} . \tag{11.194}$$

Proof Let L be an arbitrary matrix in $\mathbb{R}^{n_1 \times n_2}$ with rank r. Let U_L be a matrix with r orthonormal columns that span the column space of L, such that $U_L U_L^T L = L$. The low-rank approximation $U_L U_L^T D$ obtained by projecting the columns of D onto the column space $\operatorname{col}(L)$ of L can only improve the approximation error. By Lemma 11.38,

$$||D-L||_{\mathrm{F}}^2 = \left|\left|D - U_L U_L^T D\right|\right|_{\mathrm{F}}^2 + \left|\left|L - U_L U_L^T D\right|\right|_{\mathrm{F}}^2 \tag{11.195}$$

$$\geq \left|\left|D - U_L U_L^T D\right|\right|_{\mathrm{F}}^2 , \tag{11.196}$$

because the column space of $D_\perp := D - U_L U_L^T D$ is orthogonal to $\operatorname{col}(L)$. Indeed, each column of $D_\perp$ is the projection of the corresponding column of D onto the orthogonal complement of $\operatorname{col}(L)$. We conclude that it suffices to show that the approximation error of L_{SVD} is not larger than that of $U_L U_L^T D$.

Let U_* be a matrix with r columns equal to the first r left singular vectors of D. The SVD-based low-rank approximation can be obtained by applying $U_* U_*^T$ to each column of D. To see why, let S_* be a diagonal matrix with the first r singular values of D and V_* a matrix with the first r right singular vectors as its rows, so $L_{\mathrm{SVD}} = U_* S_* V_*^T$. We denote by $U_\perp$, $S_\perp$ and $V_\perp$ the matrices containing the rest of singular vectors and singular values,

$$D = \begin{bmatrix} U_* & U_\perp \end{bmatrix} \begin{bmatrix} S_* & 0 \\ 0 & S_\perp \end{bmatrix} \begin{bmatrix} V_* & V_\perp \end{bmatrix}^T, \tag{11.197}$$

where 0 denotes a matrix of zeros with the right dimensions. By orthogonality of the singular vectors, the columns of U_* are orthogonal to those of $U_\perp$, so

$$\begin{aligned} U_* U_*^T D &= U_* U_*^T \begin{bmatrix} U_* & U_\perp \end{bmatrix} \begin{bmatrix} S_* & 0 \\ 0 & S_\perp \end{bmatrix} \begin{bmatrix} V_* & V_\perp \end{bmatrix}^T & (11.198) \\ &= \begin{bmatrix} U_* & 0 \end{bmatrix} \begin{bmatrix} S_* & 0 \\ 0 & S_\perp \end{bmatrix} \begin{bmatrix} V_* & V_\perp \end{bmatrix}^T & (11.199) \\ &= U_* S_* V_*^T & (11.200) \\ &= L_{\text{SVD}}. & (11.201) \end{aligned}$$

Consequently, $D = U_* U_*^T D + U_\perp S_\perp V_\perp^T$, and the two components have orthogonal column spaces. By Lemma 11.38,

$$||D||_{\text{F}}^2 = \left|\left|U_* U_*^T D\right|\right|_{\text{F}}^2 + \left|\left|U_\perp S_\perp V_\perp^T\right|\right|_{\text{F}}^2, \tag{11.202}$$

so the approximation error incurred by L_{SVD} equals

$$\begin{aligned} ||D - L_{\text{SVD}}||_{\text{F}}^2 &= \left|\left|U_\perp S_\perp V_\perp^T\right|\right|_{\text{F}}^2 & (11.203) \\ &= ||D||_{\text{F}}^2 - \left|\left|U_* U_*^T D\right|\right|_{\text{F}}^2. & (11.204) \end{aligned}$$

Also, by Lemma 11.38 and the Pythagorean theorem, the error incurred by $U_L U_L^T D$ equals

$$\left|\left|D - U_L U_L^T D\right|\right|_{\text{F}}^2 = ||D||_{\text{F}}^2 - \left|\left|U_L U_L^T D\right|\right|_{\text{F}}^2. \tag{11.205}$$

To complete the proof we need to show

$$\left|\left|U_* U_*^T D\right|\right|_{\text{F}}^2 \geq \left|\left|U_L U_L^T D\right|\right|_{\text{F}}^2 \tag{11.206}$$

for any possible choice of U_L. We can express these quantities in terms of the sample covariance matrix Σ_{cols} of the columns of D. Since the Frobenius norm is just the sum of the squared entries of a matrix,

$$\left|\left|U_* U_*^T D\right|\right|_{\mathrm{F}}^2 = \sum_{j=1}^{n_2} \left|\left|U_* U_*^T D[:,j]\right|\right|_2^2 \tag{11.207}$$

$$= \sum_{j=1}^{n_2} D[:,j]^T U_* U_*^T U_* U_*^T D[:,j] \tag{11.208}$$

$$= \sum_{j=1}^{n_2} D[:,j]^T U_* U_*^T D[:,j] \tag{11.209}$$

$$= \sum_{l=1}^{r} \sum_{j=1}^{n_2} u_l^T D[:,j] D[:,j]^T u_l \tag{11.210}$$

$$= n_2 \sum_{l=1}^{r} u_l^T \left(\frac{1}{n_2} \sum_{j=1}^{n_2} D[:,j] D[:,j]^T \right) u_l \tag{11.211}$$

$$= n_2 \sum_{l=1}^{r} u_l^T \Sigma_{\text{cols}} u_l, \tag{11.212}$$

where $u_1, \ldots, u_r$ are the first left singular vectors of D and $D[:,j]$ denotes the jth column of D. This is equal to the sum of the sample variances of the data in the direction of the orthonormal vectors $u_1, \ldots, u_r$. Let $u_1^{[L]}, \ldots, u_r^{[L]}$ denote the r columns of U_L. By the same argument,

$$\left|\left|U_L U_L^T D\right|\right|_{\mathrm{F}}^2 = n_2 \sum_{l=1}^{r} (u_l^{[L]})^T \Sigma_{\text{cols}} u_l^{[L]}. \tag{11.213}$$

The term equals the sum of the sample variances of the data in the direction of $u_1^{[L]}, \ldots, u_r^{[L]}$.

In Theorem 11.27 we establish that the first r eigenvectors of a covariance matrix are the set of r orthonormal vectors that capture the most variance. The same holds for sample covariance matrices (see Exercise 11.7): the eigenvectors corresponding to the r largest eigenvalues capture the most sample variance. These eigenvectors are exactly equal to the first r left singular vectors $u_1, \ldots, u_r$ (see the proof of Theorem 11.35). Therefore (11.212) is larger than (11.213) for any value of $r \leq \min\{n_1, n_2\}$. This completes the proof. ■

11.7 Matrix Completion For Collaborative Filtering

A key challenge in recommender systems is to estimate user preferences from data, a task known as *collaborative filtering*. A famous example is the 2007 Netflix Prize contest, where the goal was to predict movie ratings of Netflix users. Collaborative filtering can often be cast as a *matrix completion* problem. To illustrate this, let us consider movie-rating prediction. We arrange ratings given by users to different movies into a matrix where each user is assigned a column, and each movie is assigned a row. The missing entries correspond to movies that have not been rated

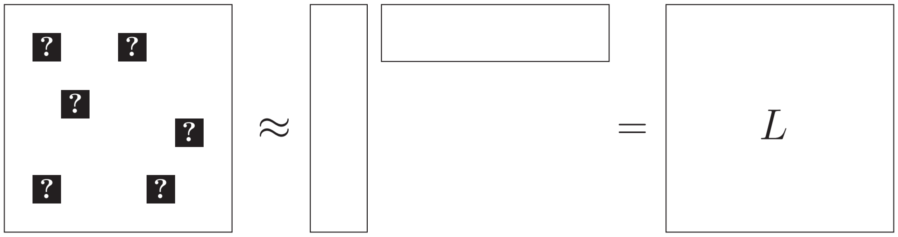


Figure 11.23 Low-rank matrix completion. In order to estimate the missing entries in a matrix, we can approximate it using a low-rank model. The parameters of the model are estimated using the observed data, and then used to form the low-rank estimate L, which provides an estimate of the unobserved entries.

by those users (see Example 11.43). Predicting the missing ratings is equivalent to *completing* the matrix.

At first glance, completing a matrix such as this one

$$\begin{bmatrix} 1 & ? & 1 \\ ? & 6 & 3 \\ ? & 4 & 2 \end{bmatrix} \tag{11.214}$$

looks like an ill-posed problem. The missing entries can be filled in arbitrarily! However, that is no longer the case if we require that the completed matrix be low rank. The low-rank assumption ties the missing entries to the observed data. Let $D \in \mathbb{R}^{n_1 \times n_2}$ denote the data matrix. The low-rank approximation to each observed entry $D[i,j]$ is

$$L[i,j] := \sum_{l=1}^{r} a_l[i] b_l[j], \qquad 1 \leq i \leq n_1, 1 \leq j \leq n_2. \tag{11.215}$$

If we observe enough entries, we can estimate the latent variables $a_1, \ldots, a_r \in \mathbb{R}^{n_1}$ and $b_1, \ldots, b_r \in \mathbb{R}^{n_1}$ from the available data. Then, we can use the parameters to compute the whole low-rank matrix, including the missing entries! Figure 11.23 illustrates the approach.

Example 11.40 (Completing a rank-1 matrix)**.** Let us fit a rank-1 model to the matrix in (11.214):

$$\begin{bmatrix} 1 & ? & 1 \\ ? & 6 & 3 \\ ? & 4 & 2 \end{bmatrix} = a_1 b_1^T = \begin{bmatrix} a_1[1]\, b_1[1] & a_1[1]\, b_1[2] & a_1[1]\, b_1[3] \\ a_1[2]\, b_1[1] & a_1[2]\, b_1[2] & a_1[2]\, b_1[3] \\ a_1[3]\, b_1[1] & a_1[3]\, b_1[2] & a_1[3]\, b_1[3] \end{bmatrix}. \tag{11.216}$$

The rank-1 model has 6 parameters, corresponding to the entries of a_1 and b_1. Our goal is to estimate these parameters from the 6 observed entries. For any choice of a_1 and b_1, we can obtain an equivalent model by multiplying all entries of a_1 by a constant α, and all entries of b_1 by $1/\alpha$. To avoid this ambiguity, we set $a_1[1] = 1$ (another option could have been to impose $||a_1||_2 = 1$). From (11.216)

$$b_1[1] = a_1[1] = 1, \qquad b_1[3] = a_1[1] = 1, \qquad a_1[2] = \frac{3}{b_1[2]} = 3, \tag{11.217}$$

$$b_1[2] = \frac{6}{a_1[2]} = 2, \qquad a_1[3] = \frac{4}{b_1[2]} = 2. \tag{11.218}$$

The resulting completed matrix is

$$L = a_1 b_1^T = \begin{bmatrix} 1 \\ 3 \\ 2 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 & \mathbf{2} & 1 \\ \mathbf{3} & 6 & 3 \\ \mathbf{2} & 4 & 2 \end{bmatrix}, \tag{11.219}$$

where the estimated missing entries are highlighted in bold.

Even under a low-rank assumption, matrix completion can be an ill-posed problem, as illustrated by the following example.

Example 11.41 (Matrix completion can be ill posed)**.** If a whole row or a whole column is missing from a data matrix, then the matrix-completion problem is ill posed. The corresponding entry of the low-rank parameters cannot possibly be determined from the observed entries. For instance, consider the problem of fitting a rank-1 model to the following data

$$\begin{bmatrix} 1 & 1 & 1 \\ ? & ? & ? \\ 1 & 1 & 1 \end{bmatrix} = a_1 b_1^T = \begin{bmatrix} 1 \\ ? \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}. \tag{11.220}$$

The second entry of a_1 cannot be recovered because it is only encoded in the missing entries. In order to avoid such situations, we need to observe some entries in every column and every row. In the context of movie-rating prediction this makes sense: how can we predict the preferences of a user if we don't observe any of their ratings?

Even if we observe every row and column, some low-rank matrices cannot be recovered from a subset of their entries. Consider the rank-1 matrix

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 23 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{11.221}$$

If we do not observe the nonzero entry 23, there isn't a unique rank-1 model that fits the data:

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & ? & 0 \\ 0 & 0 & 0 \end{bmatrix} = a_1 b_1^T = \begin{bmatrix} 0 \\ ? \\ 0 \end{bmatrix} \begin{bmatrix} 0 & ? & 0 \end{bmatrix}. \tag{11.222}$$

The second entries of a_1 and b_1 are only encoded in the missing entry, so we cannot recover them from the observed entries. In movie-rating prediction, we cannot expect to predict highly idiosyncratic preferences that are unique to a specific movie and user. For example, imagine that a user loves Star Wars movies, except for the first one, because their girlfriend broke up with them after watching it. Then, we won't be able to predict the unusual rating if it is missing, because it is completely unrelated to the available ratings.

. . . .

In Example 11.40 we complete the matrix by assuming that its rank is equal to one. In practice, we cannot expect the data to exactly follow a low-rank model. Instead, we can try to compute a rank-r model that is as close to the data as possible. A reasonable criterion to evaluate the model is the squared error between the observed entries and the low-rank estimate,

$$\sum_{(i,j)\in\text{observed}} \left(D[i,j] - \sum_{l=1}^{r} a_l[i] b_l[j] \right)^2 . \tag{11.223}$$

Unfortunately, this cost function is highly non-convex, which makes it challenging to optimize. There exist several approaches for finding a local minimum, which include variants of gradient descent (Keshavan et al., 2009) and alternating least squares (Jain et al., 2013) (see also (Candès and Recht, 2009) for an approach that uses a convex regularization term to promote low-rank structure).

Here, we present a simple heuristic procedure to minimize (11.223). Section 11.6.4 shows that truncating the SVD of a matrix yields an optimal rank-r approximation. Unfortunately, this assumes that all the entries are available, which is not the case in matrix completion. Without the missing entries, we cannot compute the SVD. A pragmatic solution is to impute those entries, for example using the average of the observed data (or the average of the corresponding row or column). Then we can compute a truncated SVD to obtain a low-rank approximation of the imputed matrix. The problem is that the resulting low-rank model *also approximates the imputed entries*. The key insight is that these estimated entries tend to be better estimates of the missing entries than our naive initial imputation. Consequently, we can use them to re-impute the missing entries and re-compute the SVD in order to obtain an even better low-rank estimate. These two steps (imputation and SVD truncation) can be repeated over and over until the estimate converges. This method is known as *hard-impute* in the literature (Mazumder et al., 2010), because we alternately impute the missing entries and hard-threshold the singular values. As illustrated by the following examples, it is often effective in practice.

Definition 11.42 (SVD-based low-rank matrix completion)**.** *Let $D \in \mathbb{R}^{n_1 \times n_2}$ denote a data matrix with incomplete entries, which have been centered by subtracting their average. We initialize the low-rank estimate by setting all its entries equal to zero, or to the mean of either the columns or rows. We fix a rank r and repeat the following steps until convergence.*

1 For (i, j) belonging to the set of observed entries, set

$$M[i,j] = \begin{cases} D[i,j] & \text{if } (i,j) \text{ is observed,} \\ L[i,j] & \text{if } (i,j) \text{ is missing.} \end{cases} \tag{11.224}$$

2 Compute the SVD of M and set

$$L = \sum_{l=1}^{r} s_l u_l v_l^T, \tag{11.225}$$

where $s_1, \ldots, s_r$ are the first r singular values of M, $u_1, \ldots, u_r$ are the corresponding left singular vectors, and $v_1, \ldots, v_r$ are the corresponding right singular vectors.

After convergence, L is the final rank-r estimate.

As illustrated in Example 11.44 below, the rank r can be chosen by evaluating the model on held-out validation data.

Example 11.43 (Rank-1 model for matrix completion)**.** The following data correspond to a subset of the movie ratings in Example 11.36:

$$D := \begin{array}{cccccc} & \text{Bob} & \text{Molly} & \text{Mary} & \text{Larry} & \\ \left(\vphantom{\begin{matrix}1\\1\\1\\1\\1\\1\end{matrix}}\right. & \begin{matrix} ? \\ ? \\ 4 \\ ? \\ 4 \\ 1 \end{matrix} & \begin{matrix} ? \\ 1 \\ 5 \\ 4 \\ ? \\ 2 \end{matrix} & \begin{matrix} 5 \\ 4 \\ 2 \\ 2 \\ 1 \\ ? \end{matrix} & \begin{matrix} 4 \\ ? \\ ? \\ 1 \\ 2 \\ 5 \end{matrix} & \left.\vphantom{\begin{matrix}1\\1\\1\\1\\1\\1\end{matrix}}\right) \begin{matrix} \text{The Dark Knight} \\ \text{Spiderman 3} \\ \text{Love Actually} \\ \text{Bridget Jones's Diary} \\ \text{Pretty Woman} \\ \text{Superman 2} \end{matrix} \end{array} \tag{11.226}$$

Our goal is to perform collaborative filtering, i.e. guess the missing ratings from the observed ones. We center the data by subtracting the mean observed rating ($m(D) = 2.94$) and impute zeros to obtain the matrix

$$\begin{bmatrix} 0 & 0 & 2.06 & 1.06 \\ 0 & -1.94 & 1.06 & 0 \\ 1.06 & 2.06 & -0.94 & 0 \\ 0 & 1.06 & -0.94 & -1.94 \\ 1.06 & 0 & -1.94 & -0.94 \\ -1.94 & -0.94 & 0 & 2.06 \end{bmatrix}. \tag{11.227}$$

We compute the SVD of this matrix. The singular values equal

$$s_1 = 4.8, \; s_2 = 2.48, \; s_3 = 2.36, \; s_4 = 1.46. \tag{11.228}$$

The first singular value is larger than the rest, suggesting that a rank-1 model might fit the data well, although this is much less obvious now that we are missing entries (compare to (11.188)). To compute the rank-1 approximation, we combine the first singular value s_1 and the singular vectors u_1 and v_1, and add back the mean rating. The resulting estimate is

$$m(D)11^T + s_1 u_1 v_1^T$$

$$= \begin{array}{ccccl} \text{Bob} & \text{Molly} & \text{Mary} & \text{Larry} & \\ \mathbf{2.28}\ (1) & \mathbf{2.08}\ (1) & 3.91\ (5) & 3.88\ (4) & \text{The Dark Knight} \\ \mathbf{2.35}\ (2) & 2.16\ (1) & 3.81\ (4) & \mathbf{3.79}\ (5) & \text{Spiderman 3} \\ 3.67\ (4) & 3.91\ (5) & 1.85\ (2) & \mathbf{1.87}\ (1) & \text{Love Actually} \\ \mathbf{3.73}\ (5) & 3.99\ (4) & 1.76\ (2) & 1.79\ (1) & \text{Bridget Jones's Diary} \\ 3.69\ (4) & \mathbf{3.93}\ (5) & 1.82\ (1) & 1.85\ (2) & \text{Pretty Woman} \\ 2.06\ (1) & 1.78\ (2) & \mathbf{4.24}\ (5) & 4.21\ (5) & \text{Superman 2} \end{array} \tag{11.229}$$

where 11^T denotes a 6×4 matrix with entries equal to one (this is just to add back the sample mean to each entry). The original ratings are shown in parenthesis and the missing ratings are in bold.

The low-rank approximation is clearly influenced by the imputed values for the missing entries: the corresponding estimates are biased towards the mean rating $m(D) = 2.94$. We address this by applying the iterative algorithm from Definition 11.42. Figure 11.24 shows the values of the missing entries as the iterations proceed. They move away from the mean and eventually converge. Let L_{iter} be the centered rank-1 estimate produced by the algorithm. Adding back the mean rating yields the final estimate

$$m(D)11^T + L_{\text{iter}}$$

$$= \begin{array}{ccccl} \text{Bob} & \text{Molly} & \text{Mary} & \text{Larry} & \\ \mathbf{1.48}\ (1) & \mathbf{1.38}\ (1) & 4.45\ (5) & 4.52\ (4) & \text{The Dark Knight} \\ \mathbf{1.50}\ (2) & 1.41\ (1) & 4.42\ (4) & \mathbf{4.50}\ (5) & \text{Spiderman 3} \\ 4.26\ (4) & 4.34\ (5) & 1.57\ (2) & \mathbf{1.51}\ (1) & \text{Love Actually} \\ \mathbf{4.18}\ (5) & 4.26\ (4) & 1.65\ (2) & 1.59\ (1) & \text{Bridget Jones's Diary} \\ 4.2\ (4) & \mathbf{4.28}\ (5) & 1.64\ (1) & 1.57\ (2) & \text{Pretty Woman} \\ 1.37\ (1) & 1.27\ (2) & \mathbf{4.55}\ (5) & 4.63\ (5) & \text{Superman 2} \end{array} \tag{11.230}$$

The original ratings are again shown in parenthesis and the missing ratings in bold. The low-rank model has clearly improved after the iterative procedure, yielding estimates for the missing values that are closer to the ground-truth entries than to the mean rating.

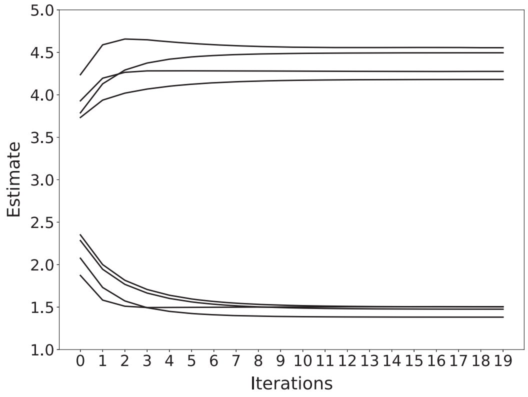


Figure 11.24 SVD-based matrix completion. Evolution of the estimates for the missing entries in Example 11.43 during the iterations of the SVD-based matrix completion method from Definition 11.42. The estimates are initially close to the mean rating, because this is the value used to impute the missing entries. As the iterations proceed, the estimates become more accurate, moving away from the mean rating.

Example 11.44 (Real movie ratings)**.** The Movielens dataset (Dataset 8) contains ratings given by a group of users to popular movies. The ratings are integers between 1 and 5. For this example we select the 1,000 users and 100 movies with the most ratings. The number of observed ratings is 30,055. We separate these data into disjoint training, validation and test sets at random. The validation and test sets both contain 1,000 ratings. Our goal is to estimate the ratings in the test set from a model trained on the training set. We evaluate the result using the root mean square error (RMSE), which is the square root of the average squared difference between the estimated and the ground-truth entries.

In order to estimate the ratings, we fit a low-rank model via the SVD-based matrix-completion method of Definition 11.42. We initialize the missing entries with the mean rating of each movie computed from the training data (this yields better results than imputing using the mean rating, or the mean rating per user). We select the rank r by evaluating the RMSE on the validation set. Figure 11.25 shows the training and validation error for different values of r. The parameters of the rank-r model are the r singular values, the 100-dimensional r left singular vectors (the movies correspond to the rows), and the 1,000-dimensional r right singular vectors. Consequently, the number of parameters is proportional to r. If r is too large, the model is able to *overfit* the training data, which hurts its generalization to the validation data. If r is too small, the model is not complex enough and *underfits* the training data. The best trade-off is achieved by $r := 3$, which yields a model with 3,303 parameters (roughly 9 times less than the training data).

Our rank-3 model achieves an RMSE of 0.89 on the test set. For comparison, a

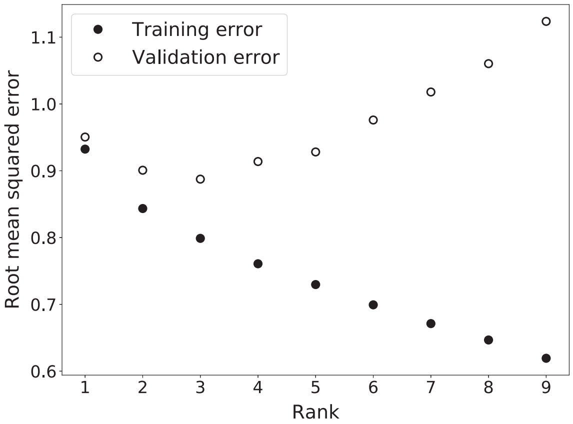


Figure 11.25 Training and validation error of movie-rating estimates for different model ranks. The graph shows the RMSE of the rank-r model in Example 11.44 for different values of r on the training set and on the validation set. On the training set, the error decreases with r, because higher-rank models are able to fit the training data better. On the validation set, the error initially decreases with r, and then increases. For small r (1 and 2) the model underfits the training data. For large r (above 3) it overfits, since its performance no longer generalizes to the validation set.

baseline that estimates each rating using the mean rating for the corresponding movie (computed from the training data) yields a test RMSE of 0.97. In order to interpret our low-rank model, we examine the singular vectors. The estimated rating for movie i and user j is equal to

$$L(i,j) := m(D) + \sum_{l=1}^{3} s_l u_l[i] v_l[j]. \tag{11.231}$$

We can interpret each of the three terms in the sum as a *factor*. The value $u_l[i]$ quantifies how movie i is associated to the lth factor. Table 11.2 shows the movies corresponding to the highest and lowest values of u_1, u_2, u_3. Even though they are learned in a completely data-driven fashion, these singular vectors capture intuitive properties:

1 The highest entries of u_1 correspond to three terrible movies: Dante's Peak (FilmAffinity score: 4.6/10), Volcano (FilmAffinity score: 4.2/10), and The Saint (FilmAffinity score: 5.5/10). The lowest entries correspond to classics that received critical acclaim: One Flew Over The Cuckoo's Nest (5 Oscars), The Godfather (3 Oscars), Casablanca (3 Oscars).
2 The highest entries of u_2 are again terrible movies, which were box-office flops: Dante's Peak, Evita (FilmAffinity score: 5.3/10) and Volcano. The lowest entries are some of the highest-grossing movies of all time: Titanic, Star Wars, Raiders of the Lost Ark.

Table 11.2 ***Interpretation of collaborative-filtering model.*** *The entries of u_1, u_2 and u_3 reveal the factors learned by the collaborative-filtering model in Example 11.44. The first factor seems to capture critical acclaim. The second seems to capture popularity and box-office success. The third differentiates between adventure blockbusters and darker, edgier films.*

l	1	2	3
Highest entries of u_l	Dante's Peak (0.29)	Volcano (0.07)	Return of the Jedi (0.14)
	Volcano (0.28)	Evita (0.05)	Star Wars (0.13)
	The Saint (0.24)	Dante's Peak (0.04)	Raiders of the Lost Ark (0.13)
Lowest entries of u_l	One Flew Over The Cuckoo's Nest (-0.10)	Titanic (-0.16)	Leaving Las Vegas (-0.28)
	The Godfather (-0.12)	Star Wars (-0.16)	Trainspotting (-0.29)
	Casablanca (-0.13)	Raiders of the Lost Ark (-0.17)	A Clockwork Orange (-0.30)

3 The highest entries of u_3 are popular adventure movies: Return of the Jedi, Star Wars, Raiders of the Lost Ark. The lowest entries are also very popular, but they are edgier and darker: Leaving Las Vegas, Trainspotting, A Clockwork Orange.

We can analyze the tastes of different users by examining the entries of the right singular vectors corresponding to these factors. Since v_1 is multiplied with u_1 in the estimated rating, large negative entries of v_1 indicate that the user enjoys classical movies such as The Godfather or Casablanca. By the same argument, large negative entries of v_2 indicate that the user likes box-office superhits like Titanic or Star Wars. Conversely, large positive entries of v_1 and v_2 indicate that the user has questionable taste, and enjoys terrible movies. Finally, large positive entries of v_3 suggest a preference for adventure blockbusters such as Star Wars, over alternative, darker movies like Trainspotting.