Chapter 2
Data Modeling with a Single Subspace

"Principal component analysis is probably the oldest and best known of the techniques of multivariate analysis."

– I. T. Jolliffe

In this chapter, we give a brief review of principal component analysis (PCA), i.e., the method for finding an optimal (affine) subspace to fit a set of data points. The solution to PCA has been well established in the literature and it has become one of the most useful tools for data modeling, compression, and visualization. We introduce both the statistical and geometric formulation of PCA and establish their equivalence. Specifically, we show that the singular value decomposition (SVD) provides an optimal solution to PCA. We also establish the similarities and differences between PCA and two generative subspace models, namely Factor Analysis (FA) and Probabilistic PCA (PPCA). When the dimension of the subspace is unknown, we introduce some conventional model selection methods to determine the number of principal components. When the data points are incomplete or contain outliers, we review some robust statistical techniques that help resolve these difficulties. Finally, some nonlinear extensions to PCA such as nonlinear PCA and kernel PCA are also reviewed.

2.1 Principal Component Analysis (PCA)

Principal component analysis (PCA) refers to the problem of fitting a low-dimensional affine subspace S to a set of points $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$ in a

high-dimensional space $\mathbb{R}^D$, the ambient space. Mathematically, this problem can be formulated as either a statistical problem or a geometric one, and they both lead to the same solution, as we will show in this section.

2.1.1 A Statistical View of PCA

Historically, PCA was first formulated in a statistical setting to estimate the principal components of a multivariate random variable $\boldsymbol{x}$ [Pearson, 1901, Hotelling, 1933]. Specifically, given a multivariate random variable $\boldsymbol{x} \in \mathbb{R}^D$ and any integer $d < D$, the d "principal components" of $\boldsymbol{x}$ are defined as the d *uncorrelated* linear components of $\boldsymbol{x}$:

$$y_i = u_i^\top \boldsymbol{x} \in \mathbb{R}, \quad u_i \in \mathbb{R}^D, \quad i = 1, 2, \ldots, d, \tag{2.1}$$

such that the variance of y_i is maximized subject to

$$u_i^\top u_i = 1 \quad \text{and} \quad \text{Var}(y_1) \geq \text{Var}(y_2) \geq \cdots \geq \text{Var}(y_d). \tag{2.2}$$

For example, to find the first principal component, y_1, we seek a vector $u_1^* \in \mathbb{R}^D$ such that

$$u_1^* = \arg\max_{u_1 \in \mathbb{R}^D} \text{Var}(u_1^\top \boldsymbol{x}), \quad \text{s.t.} \quad u_1^\top u_1 = 1. \tag{2.3}$$

Without loss of generality, in what follows, we will assume $\boldsymbol{x}$ has zero-mean.

Theorem 2.1 (Principal Components of a Random Variable)**.** *The first d principal components of a multivariate random variable $\boldsymbol{x}$ are given by $y_i = u_i^\top \boldsymbol{x}$, where $\{u_i\}_{i=1}^d$ are the d leading eigenvectors of its covariance matrix $\Sigma_{\boldsymbol{x}} \doteq \mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top]$.*

Proof. Notice that for any $u \in \mathbb{R}^D$,

$$\text{Var}(u^\top \boldsymbol{x}) = \mathbb{E}[(u^\top \boldsymbol{x})^2] = \mathbb{E}[u^T \boldsymbol{x}\boldsymbol{x}^\top u] = u^\top \Sigma_{\boldsymbol{x}} u. \tag{2.4}$$

Therefore, the optimization in problem in (2.3) for finding the first principal component is equivalent to

$$\max_{u_1 \in \mathbb{R}^D} u_1^\top \Sigma_{\boldsymbol{x}} u_1, \quad \text{s.t.} \quad u_1^\top u_1 = 1. \tag{2.5}$$

In order to solve the above constrained minimization problem, we use the Lagrange multiplier method. The Lagrangian is given by

$$\mathcal{L} = u_1^\top \Sigma_{\boldsymbol{x}} u_1 + \lambda(1 - u_1^\top u_1) \tag{2.6}$$

for some Lagrange multiplier $\lambda \in \mathbb{R}$. The necessary condition for u_1 to be an extrema is

$$\Sigma_{\boldsymbol{x}} u_1 = \lambda u_1, \tag{2.7}$$

and the associated extremum value is $u_1^\top \Sigma_{\boldsymbol{x}} u_1 = \lambda$. It follows that the optimal solution u_1^* is exactly the eigenvector of $\Sigma_{\boldsymbol{x}}$ associated with the largest eigenvalue.

To find the remaining principal components, since $u_1^\top \boldsymbol{x}$ and $u_i^\top \boldsymbol{x}$ $(i > 1)$ need to be uncorrelated, we have

$$\mathbb{E}[(u_1^\top \boldsymbol{x})(u_i^\top \boldsymbol{x})] = \mathbb{E}[u_1^\top \boldsymbol{x}\boldsymbol{x}^\top u_i] = u_1^\top \Sigma_{\boldsymbol{x}} u_i = \lambda_1 u_1^\top u_i = 0. \tag{2.8}$$

That is, $u_2, \ldots, u_d$ are all orthogonal to u_1. More generally, $u_i^\top u_j = 0$ for all $i \neq j = 1, \ldots d$. To find u_2 we define the Lagrangian

$$\mathcal{L} = u_2^\top \Sigma_{\boldsymbol{x}} u_2 + \lambda_2(1 - u_2^\top u_2) + \gamma u_1^\top u_2. \tag{2.9}$$

The necessary condition for u_2 to be an extrema is

$$\Sigma_{\boldsymbol{x}} u_2 + \gamma u_1 = \lambda_2 u_2, \tag{2.10}$$

from which it follows that $u_1^\top \Sigma_{\boldsymbol{x}} u_2 + \gamma u_1^\top u_1 = \lambda_1 u_1^\top u_2 + \gamma = \lambda_2 u_1^\top u_2$, and so $\gamma = 0$. Since the associated extremum value is $u_2^\top \Sigma_{\boldsymbol{x}} u_2 = \lambda_2$, u_2^* is the leading eigenvector of $\Sigma_{\boldsymbol{x}}$ restricted to the orthogonal complement of u_1.[1] Assuming that $\Sigma_{\boldsymbol{x}}$ does not have repeated eigenvalues, u_2^* is the eigenvector of $\Sigma_{\boldsymbol{x}}$ associated with the second largest eigenvalue. Inductively, one can show that $u_3, u_4, \ldots, u_d$ are the top third, fourth, ..., d-th eigenvectors of $\Sigma_{\boldsymbol{x}}$ and that the corresponding eigenvalues give the variance of the principal components, i.e., $\lambda_i = \text{Var}(y_i)$. □

The solution to PCA provided by Theorem 2.1 suggests that we may find the d principal components of $\boldsymbol{x}$ simultaneously, rather than one by one. Specifically, we can define a matrix a random vector $\boldsymbol{y} = [y_1, y_2, \ldots, y_d]^\top \in \mathbb{R}^d$ and a matrix $U_d = [u_1, u_2, \cdots, u_d] \in \mathbb{R}^{D \times d}$. Since $\boldsymbol{y} = U_d^\top \boldsymbol{x}$, we have that

$$\Sigma_{\boldsymbol{y}} = \mathbb{E}(\boldsymbol{y}\boldsymbol{y}^\top) = U_d^\top \mathbb{E}(\boldsymbol{x}\boldsymbol{x}^\top) U_d = U_d^\top \Sigma_{\boldsymbol{x}} U_d. \tag{2.11}$$

Since were are looking for uncorrelated random variables, the matrix Σ_y must be diagonal and the matrix U_d must be orthonormal, i.e., $U_d^\top U_d = I_d$.

Recall that any real, symmetric and positive semi-definite matrix A can be transformed into a diagonal matrix $\Lambda = V^{-1} A V$, where the columns of V are the eigenvectors of A and the diagonal entries of Λ are the corresponding eigenvalues. Recall also that the eigenvalues are real and nonnegative, i.e., $\lambda_i \geq 0$, and that the eigenvectors can be chosen to be orthonormal, so that $V^{-1} = V^\top$. Since the matrix $\Sigma_{\boldsymbol{x}}$ is real, symmetric and positive semi-definite, the equation $\Sigma_{\boldsymbol{y}} = U_d^\top \Sigma_{\boldsymbol{x}} U_d$ suggests that the columns of U_d can be chosen as d eigenvectors of $\Sigma_{\boldsymbol{x}}$ and that the diagonal entries of $\Sigma_{\boldsymbol{y}}$ can be chosen as the corresponding d eigenvalues. Moreover, since our goal is to maximize the variance of each y_i and $\lambda_i = \text{Var}(y_i)$, we conclude that the columns of U_d are the top d eigenvectors of $\Sigma_{\boldsymbol{x}}$ and the entries of $\Sigma_{\boldsymbol{y}}$ are the corresponding top d eigenvalues.

This alternative derivation of PCA allows us to understand what happens when $\Sigma_{\boldsymbol{x}}$ has repeated eigenvalues. When the eigenvalues are different, each eigenvector u_i is unique (up to sign), thus the principal components are unique (up to sign).

[1] The reason for this is that both u_1 and its orthogonal complement $u_1^\perp$ are invariant subspaces of $\Sigma_{\boldsymbol{x}}$.

When an eigenvalue is repeated, $\Sigma_{\boldsymbol{x}}$ still admits a basis of orthonormal eigenvectors. However, the eigenvectors corresponding to the repeated eigenvalue form an eigensubspace and any orthonormal basis for this eigensubspace gives valid principal components. As a consequence, the principal components are not always uniquely defined.

In practice, we may not know the population covariance matrix, $\Sigma_{\boldsymbol{x}}$. Instead, we may be given N i.i.d. samples of $\boldsymbol{x}$, $\{\boldsymbol{x}_i\}_{i=1}^N$. Let $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N]$ be the sample data matrix. It is well known from statistics that an asymptotically unbiased estimate of $\Sigma_{\boldsymbol{x}}$ is given by

$$\widehat{\Sigma}_{\boldsymbol{x}} \doteq \frac{1}{N-1}\sum_{i=1}^{N} \boldsymbol{x}_i \boldsymbol{x}_i^\top = \frac{1}{N-1}\boldsymbol{X}\boldsymbol{X}^\top. \tag{2.12}$$

We define the d "sample principal components" of $\boldsymbol{x}$ as

$$\hat{y}_i = \hat{u}_i^\top \boldsymbol{x}, \quad i = 1, \ldots, d, \tag{2.13}$$

where $\{\hat{u}_i\}_{i=1}^d$ are the top d eigenvectors of $\widehat{\Sigma}_{\boldsymbol{x}}$, or equivalently those of $\boldsymbol{X}\boldsymbol{X}^\top$. Notice also that, even though the principal components of $\boldsymbol{x}$ and the sample principal components of $\boldsymbol{x}$ are different notions, under certain assumptions on the distribution of $\boldsymbol{x}$ they can be related to each other. Specifically, one can show that, if $\boldsymbol{x}$ is Gaussian, then every eigenvector $\hat{u}$ of $\hat{\Sigma}_{\boldsymbol{x}}$ is an asymptotically unbiased estimate for the corresponding eigenvector u of $\Sigma_{\boldsymbol{x}}$ [Jollife, 1986].

2.1.2 A Geometric View of PCA

An alternative geometric view of PCA, which is very much related to the SVD [**?**, **?**], seeks to find an (affine) subspace S that fits the given data points $\{\boldsymbol{x}_i\}_{i=1}^N$.

Let us assume for now that the dimension of the subspace d is known. Then every point $\boldsymbol{x}_i$ on a d-dimensional affine subspace in $\mathbb{R}^D$ can be represented as

$$\boldsymbol{x}_i = \boldsymbol{x}_0 + U_d \boldsymbol{y}_i, \quad i = 1, 2, \ldots, N \tag{2.14}$$

where $\boldsymbol{x}_0 \in \mathbb{R}^D$ is a(ny) fixed point in the subspace, U_d is a $D \times d$ matrix whose columns form a basis for the subspace, and $\boldsymbol{y}_i \in \mathbb{R}^d$ is simply the vector of new coordinates of $\boldsymbol{x}_i$ in the subspace.

Notice that there is some redundancy in the above representation due to the arbitrariness in the choice of $\boldsymbol{x}_0$ and U_d. More precisely, for any $\boldsymbol{y}_0 \in \mathbb{R}^d$, we can re-represent $\boldsymbol{x}_i$ as $\boldsymbol{x}_i = (\boldsymbol{x}_0 + U_d \boldsymbol{y}_0) + U_d(\boldsymbol{y}_i - \boldsymbol{y}_0)$. We call this ambiguity the *translational ambiguity*. Also, for any $A \in \mathbb{R}^{d \times d}$ we can re-represent $\boldsymbol{x}_i$ as $\boldsymbol{x}_i = \boldsymbol{x}_0 + (U_d A)(A^{-1}\boldsymbol{y}_i)$. We call this ambiguity the *change of basis ambiguity*. Therefore, we need some additional constraints in order to end up with a unique solution to the problem of finding an affine subspace to for the data.

A common constraint used to resolve the translational ambiguity is to impose that the mean of $\boldsymbol{y}_i$ is zero:[2]

$$\bar{\boldsymbol{y}} \doteq \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{y}_i = 0, \tag{2.15}$$

while a common constraint used to resolve the change of basis ambiguity is to impose that the columns of U_d be orthonormal. This last constraint eliminates the change of basis ambiguity only up to a rotation, because we can still re-represent $\boldsymbol{x}_i$ as $\boldsymbol{x}_i = \boldsymbol{x}_0 + (U_d R)(R^\top \boldsymbol{y}_i)$ for some rotation R in $\mathbb{R}^d$. However, this *rotational ambiguity* can be easily deal with during optimization, as we shall see.

In general the given points are imperfect and have noise. We define the "optimal" affine subspace to be the one that minimizes the sum of squared distances between $\boldsymbol{x}_i$ and its projection onto the subspace $\boldsymbol{x}_0 + U_d \boldsymbol{y}_i$, i.e.,

$$\min_{\boldsymbol{x}_0, U_d, \{\boldsymbol{y}_i\}} \sum_{i=1}^{N} \left\| \boldsymbol{x}_i - \boldsymbol{x}_0 - U_d \boldsymbol{y}_i \right\|^2, \quad \text{s.t. } U_d^\top U_d = I_d \text{ and } \bar{\boldsymbol{y}} = 0. \tag{2.16}$$

In order to solve this optimization problem, we define the Lagrangian

$$\mathcal{L} = \sum_{i=1}^{N} \left\| \boldsymbol{x}_i - \boldsymbol{x}_0 - U_d \boldsymbol{y}_i \right\|^2 + \gamma^\top \sum_{i=1}^{N} \boldsymbol{y}_i + \mathbf{tr}(\Lambda(I_d - U_d^\top U_d)), \tag{2.17}$$

where $\gamma \in \mathbb{R}^d$ and $\Lambda = \Lambda^\top \in \mathbb{R}^{d \times d}$ are, respectively, a vector and a matrix of Lagrange multipliers.

The necessary condition for $\boldsymbol{x}_0$ to be an extrema is

$$-2\sum_{i=1}^{N} (\boldsymbol{x}_i - \boldsymbol{x}_0 - U_d \boldsymbol{y}_i) = \mathbf{0} \implies \hat{\boldsymbol{x}}_0 = \bar{\boldsymbol{x}} \doteq \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_i. \tag{2.18}$$

The necessary condition for $\boldsymbol{y}_i$ to be an extrema is

$$-2U_d^\top (\boldsymbol{x}_i - \boldsymbol{x}_0 - U_d \boldsymbol{y}_i) + \gamma = \mathbf{0}. \tag{2.19}$$

Summing over i yields $\gamma = 0$, from which we obtain

$$\hat{\boldsymbol{y}}_i = U_d^\top (\boldsymbol{x}_i - \bar{\boldsymbol{x}}). \tag{2.20}$$

The vector $\hat{\boldsymbol{y}}_i \in \mathbb{R}^d$ is simply the coordinates of the projection of $\boldsymbol{x}_i \in \mathbb{R}^D$ onto the subspace S. We may call such $\hat{\boldsymbol{y}}$ the "geometric principal components" of $\boldsymbol{x}$.[3]

[2] In the statistical setting, $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ will be samples of two random variables $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively. Then this constraint is equivalent to setting their means to be zero.

[3] As we will soon see in the next section, the geometric principal components coincide with the sample principal components defined in a statistical sense.

Before optimizing over U_d, we can replace the optimal values for $\boldsymbol{x}_0$ and $\boldsymbol{y}_i$ into the objective function. This leads to the following optimization problem

$$\min_{U_d} \sum_{i=1}^{N} \left\| (\boldsymbol{x}_i - \bar{\boldsymbol{x}}) - U_d U_d^\top (\boldsymbol{x}_i - \bar{\boldsymbol{x}}) \right\|^2 \quad \text{s.t.} \quad U_d^\top U_d = I_d. \tag{2.21}$$

Note that this is a restatement of the original problem with the mean $\bar{\boldsymbol{x}}$ subtracted from each of the sample points. Therefore, from now on, we will consider only the case in which the data points have zero mean. If not, simply subtract the mean from each point before computing U_d.

The following theorem gives a constructive solution for finding an optimal $\hat{U}_d$.

Theorem 2.2 (PCA via SVD)**.** *Let $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N] \in \mathbb{R}^{D \times N}$ be the matrix formed by stacking the (zero-mean) data points as its column vectors. Let $\boldsymbol{X} = U \Sigma V^\top$ be the SVD of the matrix $\boldsymbol{X}$. Then for any given $d < D$, an optimal solution for U_d is given by the first d columns of U, and an optimal solution for $\boldsymbol{y}_i$ is given by the ith column of the top $d \times N$ submatrix $\Sigma_d V_d^\top$ of $\Sigma V^\top$.*

Proof. Recalling that $\boldsymbol{x}^\top A \boldsymbol{x} = \mathbf{tr}(A \boldsymbol{x} \boldsymbol{x}^\top)$, we can rewrite the objective function as $\mathbf{tr}((I_D - U_d U_d^\top) \boldsymbol{X} \boldsymbol{X}^\top)$ and transform the optimization problem to

$$\max_{U_d} \mathbf{tr}(U_d U_d^\top \boldsymbol{X} \boldsymbol{X}^\top) \quad \text{s.t.} \quad U_d^\top U_d = I_d. \tag{2.22}$$

Since $\mathbf{tr}(AB) = \mathbf{tr}(BA)$, the Lagrangian can be written as

$$\mathcal{L} = \mathbf{tr}(U_d^\top \boldsymbol{X} \boldsymbol{X}^\top U_d) + \mathbf{tr}((I_d - U_d^\top U_d)\Lambda). \tag{2.23}$$

The condition for an extrema are given by

$$\boldsymbol{X} \boldsymbol{X}^\top U_d = U_d \Lambda. \tag{2.24}$$

This means that $\Lambda = U_d^\top \boldsymbol{X} \boldsymbol{X}^\top U_d$, hence the objective function is $\mathbf{tr}(\Lambda)$.

Now, recall that U_d is defined only up to a rotation, i.e., $U_d' = U_d R$. Therefore, $\Lambda' = R \Lambda R^\top$ is also a valid solution. Since Λ is symmetric, it has an orthogonal matrix of eigenvectors. Thus, we can choose R to be such a matrix, so that Λ' is diagonal. As a consequence, we can choose Λ to be diagonal without loss of generality. It follows from (2.24) that the columns of U_d must be eigenvectors of $\boldsymbol{X} \boldsymbol{X}^\top$ with the corresponding eigenvalues in the diagonal entries of Λ. Since the goal is to maximize $\mathbf{tr}(\Lambda)$, an optimal solution is given by the top d eigenvectors of $\boldsymbol{X} \boldsymbol{X}^\top$ or the top d singular vectors of $\boldsymbol{X} = U \Sigma V^\top$, i.e., the first d columns of U. Finally, it follows from (2.20) that $\boldsymbol{Y} = [\boldsymbol{y}_1, \cdots, \boldsymbol{y}_N] = U_d^\top \boldsymbol{X} = U_d^\top U \Sigma V^\top = \Sigma_d V_d^\top$. □

According to the theorem, the SVD gives an optimal solution to the PCA problem. The resulting matrix $\hat{U}_d$ (together with the mean $\bar{\boldsymbol{x}}$ if the data is not zero-mean) provides a geometric description of the dominant subspace structure for all

the points[4]; and the columns of the matrix $\Sigma_d V_d^\top = [\hat{\boldsymbol{y}}_1, \hat{\boldsymbol{y}}_2, \ldots, \hat{\boldsymbol{y}}_N] \in \mathbb{R}^{d \times N}$, i.e., the principal components, give a more compact representation for the points $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N] \in \mathbb{R}^{D \times N}$, as d is typically much smaller than D.

Theorem 2.3 (Equivalence of Geometric and Sample Principal Components)**.** *Let $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N] \in \mathbb{R}^{D \times N}$ be the data matrix (with $\bar{\boldsymbol{x}} = 0$). The vectors $\hat{u}_1, \hat{u}_2, \ldots, \hat{u}_d \in \mathbb{R}^D$ associated with the d sample principal components for $\boldsymbol{X}$ are exactly the columns of the matrix $\hat{U}_d \in \mathbb{R}^{D \times d}$ that minimizes the least-squares error (**??**).*

Proof. The proof is simple. Notice that if $\boldsymbol{X}$ has the singular value decomposition $\boldsymbol{X} = U \Sigma V^T$, then $\boldsymbol{X}\boldsymbol{X}^T = U \Sigma^2 U^T$ is the eigenvalue decomposition of $\boldsymbol{X}\boldsymbol{X}^T$. If Σ is ordered, then the first d columns of U are exactly the leading d eigenvectors of $\boldsymbol{X}\boldsymbol{X}^T$, which give the d sample principal components. □

Therefore, both the geometric and statistical formulation of PCA lead to exactly the same solutions/estimates of the principal components. The geometric formulation allows us to apply PCA to data even if the statistical nature of the data is unclear; the statistical formulation allows to quantitatively evaluate the quality of the estimates. For instance, for Gaussian random variables, one can derive explicit formulae for the mean and covariance of the estimated principal components. For a more thorough analysis of the statistical properties of PCA, we refer the reader to the classical book [Jollife, 1986].

2.2 Factor Analysis and Probabilistic PCA

The PCA model described so far allows us to find a low-dimensional representation $\{\boldsymbol{y}_i \in \mathbb{R}^d\}$ of a set of points $\{\boldsymbol{x}_i \in \mathbb{R}^D\}$, with $d \ll D$. However, the PCA model is not a proper generative model, because the low-dimensional representation $\boldsymbol{y}$ and the error ε are treated as parameters, rather than as random variables. As a consequence, the PCA model cannot be used to generate new samples $\boldsymbol{x}$.

To address this issue, assume that the low-dimensional representation $\boldsymbol{y}$ and the error ε are independent random variables with pdfs $p(\boldsymbol{y})$ and $p(\varepsilon)$, respectively. This allows us to generate a new sample of $\boldsymbol{x}$ from samples of $\boldsymbol{y}$ and ε as

$$\boldsymbol{x} = \boldsymbol{x}_0 + U_d \boldsymbol{y} + \varepsilon. \tag{2.25}$$

In this model, the entries of $\boldsymbol{y}$ are called *factors*, while the entries of U_d are called *factor loadings*. Assume that mean and covariance of $\boldsymbol{y}$ are denoted as $\mu_{\boldsymbol{y}}$ and $\Sigma_{\boldsymbol{y}}$, respectively. Assume also that ε is zero mean with covariance Σ_ε. The mean and covariance of the observations are then given by

$$\mu_{\boldsymbol{x}} = \boldsymbol{x}_0 + U_d \mu_{\boldsymbol{y}} \quad \text{and} \quad \Sigma_{\boldsymbol{x}} = U_d \Sigma_{\boldsymbol{y}} U_d^\top + \Sigma_\varepsilon. \tag{2.26}$$

[4] From a statistical standpoint, the column vectors of U_d give the directions in which the data X has the largest variance, hence the name "principal components." See the next section for detail.

The remainder of the section discusses three different methods for estimating the parameters of this model, $\boldsymbol{x}_0$, U_d, μ_y, $\Sigma_{\boldsymbol{y}}$ and Σ_ε, from the mean and covariance of the population, $\mu_{\boldsymbol{x}}$ and $\Sigma_{\boldsymbol{x}}$, or from i.i.d. samples $\{\boldsymbol{x}_i\}_{i=1}^N$.

2.2.1 Estimation by Linear Algebra

Observe that, in general, we cannot recover model parameters from $\mu_{\boldsymbol{x}}$ and $\Sigma_{\boldsymbol{x}}$. For instance, notice that $\boldsymbol{x}_0$ and $\mu_{\boldsymbol{y}}$ cannot be uniquely recovered from $\mu_{\boldsymbol{x}}$. As in the case of PCA, this issue can be easily resolved by assuming that $\mu_{\boldsymbol{y}} = \boldsymbol{0}$. This leads to the following estimate of $\boldsymbol{x}_0$

$$\widehat{\boldsymbol{x}}_0 = \mu_{\boldsymbol{x}}, \tag{2.27}$$

which is the same estimate as that of PCA.

Another ambiguity, which cannot be resolved in a straightforward manner, is that $\Sigma_{\boldsymbol{y}}$ and Σ_ε cannot be uniquely recovered from $\Sigma_{\boldsymbol{x}}$. For instance, $\Sigma_{\boldsymbol{y}} = 0$ and $\Sigma_\varepsilon = \Sigma_{\boldsymbol{x}}$ is a valid solution. However, this solution is not meaningful, because it assigns all the information in $\Sigma_{\boldsymbol{x}}$ to the error, rather than to the low-dimensional representation. Intuitively we would like $\Sigma_{\boldsymbol{y}}$ to capture as much information about $\Sigma_{\boldsymbol{x}}$ as possible. Thus it makes sense for $\Sigma_{\boldsymbol{y}}$ to be full rank and for Σ_ε to be as close to zero as possible. In what follows, we discuss two ways of achieving this, which lead to two well known methods: Factor Analysis and Probabilistic PCA.

Factor Analysis

Factor Analysis (FA) resolves the aforementioned ambiguity by assuming that

1. the low-dimensional representation has unit covariance $\Sigma_{\boldsymbol{y}} = I_d \in \mathbb{R}^{d\times d}$ and
2. the noise covariance matrix $\Sigma_\varepsilon \in \mathbb{R}^{D\times D}$ is diagonal.

These assumptions lead to the following relationship

$$\Sigma_{\boldsymbol{x}} = U_d U_d^\top + \Sigma_\varepsilon, \tag{2.28}$$

from which it follows that the off-diagonal entries of $\Sigma_{\boldsymbol{x}}$ are equal to the off-diagonal entries of $U_d U_d^\top$. As a consequence, even though both FA and PCA try to capture as much information from $\Sigma_{\boldsymbol{x}}$ into $\Sigma_{\boldsymbol{y}}$, the information they attempt to capture is not the same. On the one hand, FA tries to find a matrix U_d such that the covariances are preserved, i.e., the off-diagonal entries of $\Sigma_{\boldsymbol{x}}$. On the other hand, PCA tries to preserve the variances, i.e., the diagonal entries of $\Sigma_{\boldsymbol{x}}$.

The analysis above implies that, in general, the solutions to PCA and FA need not be the same. To see this, simply multiply (2.28) on the right by U_d to obtain

$$(\Sigma_{\boldsymbol{x}} - \Sigma_\varepsilon)U_d = U_d\Lambda, \tag{2.29}$$

where $\Lambda = U_d^\top U_d \succ 0$. Notice that Λ is, without loss of generality, a diagonal matrix. This is because U_d is defined only up to a rotation R_d, thus if Λ is not diagonal, we can replace it by $R_d^\top U_d^\top U_d R_d$, which is diagonal if R_d is chosen as

the matrix of eigenvectors of $U_d^\top U_d$. As a consequence the columns of U_d must be eigenvectors of $\Sigma_{\boldsymbol{x}} - \Sigma_\varepsilon$. Such eigenvectors do not generally coincide with the top d eigenvectors of $\Sigma_{\boldsymbol{x}}$, which give the solution to PCA. Moreover, the eigenvectors of $\Sigma_{\boldsymbol{x}} - \Sigma_\varepsilon$ cannot be directly computed without knowing Σ_ε. As a consequence, the solution to FA is often found via the following iterative procedure:

1. Initialize $\Sigma_\varepsilon = 0$.
2. Given Σ_ε, set $U_d = U_1 \Sigma_1^{1/2}$, where the columns of U_1 are the top d eigenvectors of $\Sigma_{\boldsymbol{x}} - \Sigma_\varepsilon$ and Σ_1 is a diagonal matrix whose diagonal entries are the corresponding eigenvalues.
3. Given U_d, set $\Sigma_\varepsilon = \text{diag}(\Sigma_{\boldsymbol{x}} - U_d U_d^\top)$.
4. Go to 2. until convergence.

Notice that the solutions to PCA and FA are initially the same, except for the linear transformation $\Sigma_1^{1/2}$. However, as the iterations proceed, the solutions are generally different.

Probabilistic Principal Component Analysis

Probabilistic Principal Component Analysis (PPCA) provides a non iterative solution to FA by further assuming that the errors are isotropic, i.e., $\Sigma_\varepsilon = \sigma^2 I_D$ for some $\sigma > 0$. In this case, we have that

$$(\Sigma_{\boldsymbol{x}} - \sigma^2 I_D) U_d = U_d \Lambda. \tag{2.30}$$

Therefore, the columns of U_d must be eigenvectors of $\Sigma_{\boldsymbol{x}} - \sigma^2 I_D$, which are the same as the eigenvectors of $\Sigma_{\boldsymbol{x}}$. Since we want σ to be as small as possible, it makes sense to choose the top d eigenvectors of $\Sigma_{\boldsymbol{x}}$. So see this, let $U_d = U_1 \Gamma$, where the columns of $U_1 \in \mathbb{R}^{D \times d}$ are *any* d orthonormal eigenvectors of $\Sigma_{\boldsymbol{x}}$ and $\Gamma \in \mathbb{R}^{d \times d}$ is a diagonal matrix, which scales these eigenvectors so that they satisfy $U_d^\top U_d = \Lambda$. Since $U_1^\top U_1 = I_d$, we obtain $\Gamma^2 = \Lambda = \Sigma_1 - \sigma^2 I_d$, where Σ_1 is a diagonal matrix with the d eigenvalues of $\Sigma_{\boldsymbol{x}}$ corresponding to the d eigenvectors in U_1. Now, recalling that $\Sigma_{\boldsymbol{x}} = U_d U_d^\top + \sigma^2 I_D$ we have that

$$\begin{aligned} \mathbf{tr}(\Sigma_{\boldsymbol{x}}) &= \mathbf{tr}(U_d U_d^\top) + \mathbf{tr}(\sigma^2 I_D) = \mathbf{tr}(U_d^\top U_d) + D\sigma^2 && (2.31) \\ &= \mathbf{tr}(\Lambda) + D\sigma^2 = \mathbf{tr}(\Sigma_1) + (D-d)\sigma^2. && (2.32) \end{aligned}$$

Therefore, the smallest possible σ is obtained when $\mathbf{tr}(\Sigma_1)$ is maximized, which happens if we choose the diagonal entries of Σ_1 to be the top d eigenvalues.

In summary, we have shown that the optimal solution to PPCA is given by

$$\widehat{U}_d = U_1 (\Sigma_1 - \widehat{\sigma}^2 I)^{1/2} \quad \text{and} \quad \widehat{\sigma}^2 = \frac{1}{D-d} \sum_{i=d+1}^{D} \lambda_i, \tag{2.33}$$

where U_1 is the matrix with the top d eigenvectors of $\Sigma_{\boldsymbol{x}}$, Σ_1 is the matrix with the corresponding d top eigenvalues, and λ_i is the i-th eigenvalue of $\Sigma_{\boldsymbol{x}}$.

2.2.2 Estimation by Maximum Likelihood

In general, we may not know the true covariance matrix $\Sigma_{\boldsymbol{x}}$. Instead, we are given samples $\{\boldsymbol{x}_i\}_{i=1}^N$ from which we can estimate the sample covariance matrix $\widehat{\Sigma}_{\boldsymbol{x}}$. The question is whether the model parameters can be estimated as in the previous section after replacing $\Sigma_{\boldsymbol{x}}$ by $\widehat{\Sigma}_{\boldsymbol{x}}$. As it turns out, the maximum likelihood estimates of the model parameters can be computed as before when $\boldsymbol{x}$, $\boldsymbol{y}$ and ε are assumed to be Gaussian random variables.

More specifically, assume that both $\boldsymbol{y}$ and ε are Gaussian random variables $\boldsymbol{y} \sim G(\mu_{\boldsymbol{y}}, \Sigma_{\boldsymbol{y}})$ and $\varepsilon \sim G(\mathbf{0}, \Sigma_\varepsilon)$. This implies that $\boldsymbol{x}$ is also Gaussian, because it is a linear combination of Gaussians. Specifically, $\boldsymbol{x} \sim G(\mu_{\boldsymbol{x}}, \Sigma_{\boldsymbol{x}})$, where $\mu_{\boldsymbol{x}}$ and $\Sigma_{\boldsymbol{x}}$ are given in (2.26). Assume also that $\Sigma_{\boldsymbol{y}} = I_d$ and that $\Sigma_\varepsilon = \sigma^2 I$. The maximum likelihood estimate for $\mu_{\boldsymbol{x}}$ is $\frac{1}{N}\sum_{i=1}^N \boldsymbol{x}_i$. The maximum likelihood estimates for U_d and Σ_ε are obtained by maximizing

$$\mathcal{L}(U_d, \Sigma_\varepsilon) = -\frac{ND}{2}\log(2\pi) - \frac{N}{2}\log\det(\Sigma_{\boldsymbol{x}}) - \frac{N}{2}\,\mathbf{tr}(\Sigma_{\boldsymbol{x}}^{-1}\widehat{\Sigma}_{\boldsymbol{x}}) \tag{2.34}$$

subject to $\Sigma_{\boldsymbol{x}} = U_d U_d^\top + \Sigma_\varepsilon$.

After taking derivatives with respect to U_d, we obtain

$$\frac{\partial \mathcal{L}}{\partial U_d} = -N\Sigma_{\boldsymbol{x}}^{-1}U_d + N\Sigma_{\boldsymbol{x}}^{-1}\widehat{\Sigma}_{\boldsymbol{x}}\Sigma_{\boldsymbol{x}}^{-1}U_d = 0 \implies \widehat{\Sigma}_{\boldsymbol{x}}\Sigma_{\boldsymbol{x}}^{-1}U_d = U_d. \tag{2.35}$$

One possible solution is $U_d = 0$, which leads to a minimum of the log-likelihood and violates our assumption that U_d should be full rank. Another possible solution is $\Sigma_{\boldsymbol{x}} = \widehat{\Sigma}_{\boldsymbol{x}}$, where the covariance model is exact. This corresponds to the case discussed in the previous section, after replacing $\Sigma_{\boldsymbol{x}}$ by $\widehat{\Sigma}_{\boldsymbol{x}}$. Thus, the model parameters can be computed as before. A third solution is obtained when $U_d \neq 0$ and $\Sigma_{\boldsymbol{x}} \neq \widehat{\Sigma}_{\boldsymbol{x}}$. In this case, we have,

$$\Sigma_{\boldsymbol{x}}U_d = U_d(\Lambda + \sigma^2 U_d) \implies U_d = \Sigma_{\boldsymbol{x}}^{-1}U_d(\Lambda + \sigma^2 I_d) \tag{2.36}$$

$$\implies \widehat{\Sigma}_{\boldsymbol{x}}U_d = U_d(\Lambda + \sigma^2 I_d) \tag{2.37}$$

Notice that the last equation is the same as that in (2.30) with $\Sigma_{\boldsymbol{x}}$ replaced by $\widehat{\Sigma}_{\boldsymbol{x}}$. Therefore, the optimal solution is of the form $U_d = U_1(\Sigma_1 - \sigma^2 I)^{1/2}$, where U_1 is a matrix with d eigenvectors of $\widehat{\Sigma}_{\boldsymbol{x}}$ with the corresponding eigenvalues in Σ_1.

Before replacing this solution into (2.34), recall two well known identities, the matrix determinant lemma $\det(A + UV^\top) = \det(I + V^\top A^{-1}U)\det(A)$ and the matrix inversion lemma $(A + UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1}$. Applying the matrix determinant lemma to $\det(\Sigma_{\boldsymbol{x}})$ leads to

$$|U_d U_d^\top + \sigma^2 I_D| = |I_d + \sigma^{-2}U_d^\top U_d)||\sigma^2 I_D| = |(\Sigma_1/\sigma^2)|\sigma^{2D} = |\Sigma_1|\sigma^{2(D-d)}, \tag{2.38}$$

while applying the matrix inversion lemma to $\Sigma_{\boldsymbol{x}}$ leads to

$$(U_d U_d^\top + \sigma^2 I_D)^{-1} = \frac{I_D}{\sigma^2} - \frac{U_d}{\sigma^2}(I_d + \frac{1}{\sigma^2} U_d^\top U_d)^{-1} \frac{U_d^\top}{\sigma^2} \tag{2.39}$$

$$= \frac{1}{\sigma^2}(I_D - U_d \Lambda^{-1} U_d^\top) = \frac{1}{\sigma^2}(I_D - U_1 U_1^\top) \tag{2.40}$$

Therefore, the log-likelihood can be rewritten as

$$\mathcal{L} = -\frac{ND}{2}\log(2\pi) - \frac{N}{2}\big((D-d)\log\sigma^2 + \log\det(\Sigma_1)\big) \tag{2.41}$$

$$- \frac{N}{2\sigma^2}\,\mathbf{tr}(\widehat{\Sigma}_{\boldsymbol{x}} - U_1 U_1^\top \widehat{\Sigma}_{\boldsymbol{x}}) \tag{2.42}$$

The condition for an extrema in σ^2 is given by

$$\frac{\partial \mathcal{L}}{\partial \sigma^2} = -\frac{N}{2}\frac{D-d}{\sigma^2} + \frac{N}{2\sigma^4}\big(\,\mathbf{tr}(\widehat{\Sigma}_{\boldsymbol{x}}) - \,\mathbf{tr}(U_1^\top \widehat{\Sigma}_{\boldsymbol{x}} U_1)\big) = 0. \tag{2.43}$$

Since $\mathbf{tr}(U_1^\top \widehat{\Sigma}_{\boldsymbol{x}} U_1) = \mathbf{tr}(\Sigma_1)$, we conclude that

$$\sigma^2 = \frac{1}{D-d}\big(\,\mathbf{tr}(\widehat{\Sigma}_{\boldsymbol{x}}) - \,\mathbf{tr}(\Sigma_1)\big). \tag{2.44}$$

This expression is minimized when $\mathbf{tr}(\Sigma_1)$ is maximized, which happens when Σ_1 is chosen as the matrix with the top d eigenvalues of $\widehat{\Sigma}_{\boldsymbol{x}}$.

In summary, we have shown that the optimal solution to PPCA is given by

$$\widehat{U}_d = U_1(\Sigma_1 - \widehat{\sigma}^2 I)^{1/2} \quad \text{and} \quad \widehat{\sigma}^2 = \frac{1}{D-d}\sum_{i=d+1}^{D} \lambda_i, \tag{2.45}$$

where U_1 is the matrix with the top d eigenvectors of $\widehat{\Sigma}_{\boldsymbol{x}}$, Σ_1 is the matrix with the corresponding d top eigenvalues, and λ_i is the i-th eigenvalue of $\widehat{\Sigma}_{\boldsymbol{x}}$.

2.3 Model Selection and Robustness Issues for PCA

In the above discussions, we have assumed that all the sample points can be fit with the same geometric or statistical model. In this section, we discuss various robustness issues for PCA. More specifically, we study how to resolve the difficulties with outliers and incomplete data points.

2.3.1 Determining the Number of Principal Components

Notice that SVD of the noisy data matrix $\boldsymbol{X}$ does not only give a solution to PCA for a particular d, but also the solutions to all $d = 1, 2, \ldots, D$. This has an important side-benefit: If the dimension d of the subspace S, or equivalently the rank of the matrix $\boldsymbol{X}$, is *not* known or specified a priori, one may have to look at the entire spectrum of solutions to decide on the "best" estimate $\hat{d}$ for the dimension and hence the subspace S for the given data.

The problem of determining the optimal dimension d is in fact a "model selection" problem. As we have discussed in the introduction of the book, the conventional wisdom is to strike a good balance between the complexity of the chosen model and the data fidelity (to the model). In Appendix A, we have given a brief review of some general model-selection criteria. One can certainly directly employ any of those for PCA (see Appendix A.4.2 for detail). We here discuss a few heuristic criteria that are especially designed for PCA and are easy to use in practice.

In PCA, the dimension d of the subspace S can be viewed as a natural measure of model complexity; and the sum of squares of the remaining singular values $\sum_{i=d+1}^{D} \sigma_i^2$ is exactly the modeling error $\sum_{i=1}^{N} \|\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i\|^2$ (see the proof of Theorem 2.2). Normally, the leading term σ_{d+1}^2 of $\sum_{i=d+1}^{D} \sigma_i^2$ is already a good index of the magnitude of the remaining ones. Thus, one can simply seek for a balance between d and σ_{d+1}^2 by minimizing an objective function of the form:

$$J_{PCA}(d) \doteq \alpha \cdot \sigma_{d+1}^2 + \beta \cdot d \tag{2.46}$$

for some proper weights $\alpha, \beta > 0$. Another somewhat similar criterion that people often use to determine the rank d of a noisy matrix $\boldsymbol{X}$ is:

$$J_{rank}(d) \doteq \frac{\sigma_{d+1}^2}{\sum_{i=1}^{d} \sigma_i^2} + \kappa d, \tag{2.47}$$

where $\kappa > 0$ is a proper weight (see [Kanatani and Matsunaga, 2002a]). In this book, unless stated otherwise, this will be the criterion of choice when we try to determine the rank of a (data) matrix corrupted by noise.

In general, the ordered singular values of the data matrix $\boldsymbol{X}$ versus the dimension d of the subspace resemble a plot as in Figure 2.1. In the statistics literature, this is known as the "Scree graph." We will see a significant drop in the singular

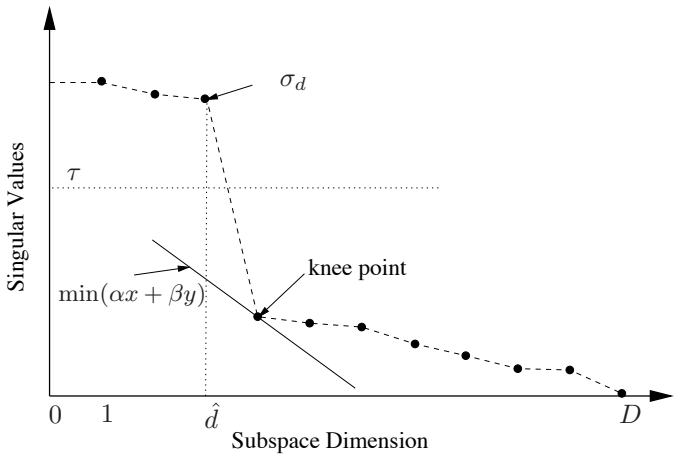

Figure 2.1. Singular value as a function of the dimension of the subspace.

value right after the "correct" dimension $\hat{d}$, which is sometimes called the "knee" or "elbow" point of the plot. Obviously, such a point is a stable minimum as it optimizes the above objective function (2.46) for a range of values for α and β, or (2.47) for a range of κ.

A model can also be selected from the Scree graph in another way. If, instead of the dimension d, a tolerance τ for the modeling error is specified, one can easily use the plot to identify the model that has the lowest dimension and satisfies the given tolerance, as indicated in the figure.

There are many other methods for determining the dimension for PCA. Interested readers may find more references in [Jollife, 1986].

2.3.2 Outliers

In practice, it is often the case that a small portion of the data points do not fit well the same model as the rest of the data. Such points are called *outliers*. The true nature of outliers can be very elusive. There is really no unanimous definition for what an outlier is.[5] Outliers can be perfectly valid samples from the same distribution as the rest of the data and it just happens so that they are *small-probability* instances; or they are not samples drawn from the same model at all and therefore they will likely *not be consistent* with the model derived from the rest of the data; or they are atypical samples that have an unusually *large influence* on the estimated model parameters. In principle however, there is no way that one can tell which case is really true for a particular "outlying" sample point. In fact, for many common noise models, all these cases lead to more or less equivalent criteria for detecting or accommodating outliers. However, these different interpretations may lead to different approaches to detect (and subsequently eliminate or accommodate) outliers. We here discuss a few approaches that are particularly related to PCA. In Chapter **??**, we will further explore the possibility of generalizing these approaches to GPCA.

Probability-Based Outlier Detection

The first approach is to first fit a model to *all* the sample points, including potential outliers, and then detect the outliers as the ones that, with respect to the identified model, correspond to small-probability events or have large modeling errors. In PCA, if we assume the samples are all drawn from a (zero-mean) Gaussian distribution, the covariance of the distribution can be estimated as $\hat{\Sigma} = \frac{1}{N-1}\sum_{i=1}^{N} \boldsymbol{x}_i \boldsymbol{x}_i^T$. The probability distribution is approximately $p(\boldsymbol{x}) \propto \exp(-\frac{1}{2}\boldsymbol{x}^T \hat{\Sigma}^{-1} \boldsymbol{x})$. If the probability of a sample $\boldsymbol{x}_i$ is small if and only if the following quantity

$$d_i = \boldsymbol{x}_i^T \hat{\Sigma}^{-1} \boldsymbol{x}_i \tag{2.48}$$

[5] For a more thorough exposition of outliers in statistics, we recommend the books of [Barnett and Lewis, 1983, Huber, 1981].

is large. The quantity d_i is also known as the *Mahalanobis distance*. In terms of the principal components $\boldsymbol{y} = U^T \boldsymbol{x}$, the Mahalanobis distance can also be written as

$$d_i = \sum_{i=1}^{D} \frac{y_i^2}{\sigma_i^2}, \tag{2.49}$$

where σ_i are the singular values of $\boldsymbol{X}$ (or equivalently, σ_i^2 are the eigenvalues of $\hat{\Sigma}$).

Thus, one can remove a certain percentage (say 10 percent) of samples that have relatively large Mahalanobis distance, as outliers. Once the outliers are trimmed out, one can use the remaining samples to re-estimate the covariance matrix $\hat{\Sigma}$ as well as their principal components. One can repeat the above trimming process until the estimate of the covariance matrix stabilizes. The resulting estimate will in general be more robust. This is essentially the basic idea of a very popular robust covariance estimator, known as *multivariate trimming* (MVT). The reader may refer to Appendix A.5 for more details. As we will see in Chapter **??**, this scheme will also be very useful in the context of GPCA.

Consensus-Based Outlier Detection

The second approach assumes that the outliers are not drawn from the same model as the rest of the data. Hence it makes sense to try to avoid the outliers when we infer the model in the first place. However, without knowing which points are outliers beforehand, how can we avoid them? One idea is to fit a model, instead of to all the data points at the same time, only to a *subset* of the data. This is possible when the number of data points required for a unique solution for the estimate is *much* smaller than that of the given data set. Of course, one should *not* expect that a randomly chosen subset will have no outliers and always lead to a good estimate of the model. Thus, one should try on *many different subsets*:

$$\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n \subset \boldsymbol{X}. \tag{2.50}$$

The rationale is that if the number of subsets are large enough,[6] one of the trial subsets, say $\boldsymbol{X}_i$, likely contains few or no outliers and hence the resulting model would be the most consistent with the rest of the data points. For instance, for PCA we may claim a subset $\boldsymbol{X}_i$ gives a consistent estimate $\hat{U}_d(\boldsymbol{X}_i)$ of the subspace if the following criterion is maximized (among all the chosen subsets):

$$\max_i \#\big\{\boldsymbol{x} \in \boldsymbol{X} : \big\|\boldsymbol{x} - \hat{U}_d(\boldsymbol{X}_i)\big\| \leq \tau\big\}, \tag{2.51}$$

where $\tau > 0$ is a chosen error threshold. This scheme is typically called *Random Sample Consensus* (RANSAC) [Fischler and Bolles, 1981], and it normally improves the robustness of the estimate. As a word of caution, in practice, in order to design a successful RANSAC algorithm, one needs to carefully choose a few

[6] See Appendix A.5 for details on how large this number needs to be.

key parameters: the size of every subset, the number of subsets, and the consensus criterion.[7] There is a vast amount of literature on RANSAC-type algorithms, especially in computer vision. For more details on RANSAC and other related random sampling techniques, the reader is referred to Appendix A.5. In Chapter **??**, we will discuss some limitations of RANSAC in the context of estimating multiple subspaces simultaneously.

Influence-Based Outlier Detection

The third approach relies on the assumption that an outlier is an *atypical* sample which has an unusually large influence on the estimated model parameters. This leads to an outlier detection scheme which to some extent combines the characteristics of the previous two approaches: it determines the influence of a sample by comparing the difference between the model estimated with and without this sample. For instance, for PCA one may use a *sample influence function* to measure the difference:

$$I(\boldsymbol{x}_i, U_d) \doteq \langle \hat{U}_d, \hat{U}_{d(i)} \rangle, \tag{2.52}$$

where $\langle \cdot, \cdot \rangle$ is the largest subspace angle (see Exercise 2.2) between the subspace $\text{span}(\hat{U}_d)$ estimated with the ith sample and the subspace $\text{span}(\hat{U}_{d(i)})$ without the ith sample. The larger the difference, the larger the influence of $\boldsymbol{x}_i$ on the estimate, and the more likely that $\boldsymbol{x}_i$ is an outlier. Thus, we may eliminate a sample $\boldsymbol{x}_i$ as an outlier if

$$I(\boldsymbol{x}_i, U_d) \geq \tau \tag{2.53}$$

for some threshold $\tau > 0$ or $I(\boldsymbol{x}_i, U_d)$ is relatively large among all the samples. However, this method does not come without an extra cost. We need to compute the principal components (and hence perform SVD) N times: one time with all the samples together and another $N-1$ times with one sample eliminated from each time. There have been many studies that aim to give a formula that can accurately approximate the sample influence without performing SVD N times. Such a formula is called a *theoretical influence* function. For more detailed discussion of the sample influence (as well the other robust statistical techniques) for PCA, we refer the interested readers to [Jollife, 2002].

2.3.3 Incomplete Data Points

Another issue that we often encounter in practice is that some of the given data points are "incomplete." For an incomplete data point $\boldsymbol{x} = [x_1, x_2, \ldots, x_D]^T$, we mean that some of its entries are missing or unspecified. For instance, if the x_i-entry is missing from $\boldsymbol{x}$, it means that we know $\boldsymbol{x}$ only up to a line in $\mathbb{R}^D$:

$$\boldsymbol{x} \in L \doteq \big\{ [x_1, \ldots, x_{i-1}, t, x_{i+1}, \ldots, x_D]^T, t \in \mathbb{R} \big\}. \tag{2.54}$$

[7] That is, the criterion that verifies whether each sample is consistent with the model derived from the subset.

One should be aware that an incomplete data point is in nature rather different from a noisy data point or an outlier.[8] In general, such incomplete data points can contain useful information about the model, and in the case of PCA, the principal subspace. For instance, if the principal subspace happens to contain the line L, the principal subspace can be determined from a sufficiently large number of such lines. In general, the line L may or may not lie in the principal subspace. We therefore should handle incomplete data points with more care.

A useful observation here is that an incomplete data point $\boldsymbol{x}$ is just as good as any point on the line L. Hence it is natural to choose a representative $\hat{\boldsymbol{x}} \in L$ that is the closest to the principal subspace. If we denote $B_d \doteq I - U_d U_d^T$, then the closest point $\boldsymbol{x}^* = [x_1, \ldots, x_{i-1}, t^*, x_{i+1}, \ldots, x_D]^T$ on L to the principal subspace can be found by minimizing the following quadratic function in t:

$$t^* = \arg\min_t \left(\boldsymbol{x}^T B_d^T B_d \boldsymbol{x}\right). \tag{2.55}$$

This problem has a unique solution as long as the line L is not parallel to the principal subspace, i.e., $e_i \notin \text{span}(U_d)$.

In essence, the above process of finding $\boldsymbol{x}^*$ on the principal subspace is to give a rank-d approximation of the entire data set containing both complete and incomplete data points. Mathematically, PCA with incomplete data is equivalent to finding a rank-d approximation/factorization of the data matrix $\boldsymbol{X}$ with incomplete data entries (in a least-squares sense). In numerical linear algebra, *power factorization* is especially designed to solve this problem. We refer the interested readers to [Vidal and Hartley, 2004] and references therein.

2.4 Extensions to PCA

Although PCA offers a rather useful tool to model the linear structure of a given data set, it however becomes less effective when the data actually has some significant nonlinearity, e.g., belonging to some nonlinear manifold. In this section, we introduce some basic extensions to PCA which can, to some extent, handle the difficulty with nonlinearity.

2.4.1 Nonlinear PCA

For nonlinear data, the basic rationale is not to apply PCA directly to the given data, but rather to a transformed version of the data. More precisely, we seek a nonlinear transformation (more precisely, usually an embedding):

$$\begin{aligned} \phi(\cdot): \mathbb{R}^D &\rightarrow \mathbb{R}^M, \\ \boldsymbol{x} &\mapsto \phi(\boldsymbol{x}), \end{aligned}$$

[8] One can view incomplete data points as a very special type of noisy data points which have infinite uncertainty only in certain directions.

such that the structure of the resulting data $\{\phi(\boldsymbol{x}_i)\}$ becomes (significantly more) linear. In machine learning, $\phi(\boldsymbol{x})$ is called the "feature" of the data point $\boldsymbol{x}$, and $\mathbb{R}^M$ is called the "feature space."

Define the matrix $\Phi \doteq [\phi(\boldsymbol{x}_1), \phi(\boldsymbol{x}_2), \ldots, \phi(\boldsymbol{x}_N)] \in \mathbb{R}^{M \times N}$. The principal components in the feature space are given by the eigenvectors of the sample covariance matrix[9]

$$\Sigma_{\phi(\boldsymbol{x})} \doteq \frac{1}{N-1} \sum_{i=1}^{N} \phi(\boldsymbol{x}_i)\phi(\boldsymbol{x}_i)^T = \frac{1}{N-1} \Phi\Phi^T \quad \in \mathbb{R}^{M \times M}.$$

Let $v_i \in \mathbb{R}^M$ be the eigenvectors:

$$\Sigma_{\phi(\boldsymbol{x})} v_i = \lambda_i v_i, \quad i = 1, 2, \ldots, M. \tag{2.56}$$

Then the d "nonlinear principal components" of every data point $\boldsymbol{x}$ are given by

$$y_i \doteq v_i^T \phi(\boldsymbol{x}) \ \in \mathbb{R}, \quad i = 1, 2, \ldots, d. \tag{2.57}$$

In general, we do not expect that the map $\phi(\cdot)$ is given together with the data. In many cases, searching for the proper map is a difficult task, and the use of nonlinear PCA is therefore limited. However, in some practical applications, good candidates for the map $\phi(\cdot)$ can be found from the nature of the problem. In such cases, the map, together with PCA, can be very effective in extracting the overall geometric structure of the data.

Example 2.4 (Veronese Map for an Arrangement of Subspaces). As we will see later in this book, if the data points belong to a union of multiple subspaces, then a natural choice of the transformation $\phi(\cdot)$ is the Veronese map:

$$\begin{aligned} \nu_n(\cdot): \quad \boldsymbol{x} \quad &\mapsto \quad \nu_n(\boldsymbol{x}), \\ (x_1, \ldots, x_D) \quad &\mapsto \quad (x_1^n, x_1^{n-1}x_2, \ldots, x_D^n), \end{aligned}$$

where the monomials are ordered in the degree-lexicographic order. Under such a mapping, the multiple low-dimensional subspaces are mapped into a single subspace in the feature space, which can then be identified via PCA for the features. ■

NLPCA in a High-dimensional Feature Space.

There is a potential difficulty associated with nonlinear PCA. The dimension of the feature space, depending on the map $\phi(\cdot)$, can be very high and it may be computationally prohibitive to compute the principal components in the feature space. For instance, if we try to search for a Veronese map of the proper degree n, the dimension of the feature space M grows exponentially with the degree. When M exceeds N, the eigenvalue decomposition of $\Phi\Phi^T \in \mathbb{R}^{M \times M}$ becomes more costly than that of $\Phi^T\Phi \in \mathbb{R}^{N \times N}$, although the two matrices have the same eigenvalues.

[9] In principle, we should use the notation $\hat{\Sigma}_{\phi(\boldsymbol{x})}$ to indicate that it is the estimate of the actual covariance matrix. But for simplicity, we will drop the hat in the sequel and simply use $\Sigma_{\phi(\boldsymbol{x})}$. The same goes for the eigenvectors and the principal components.

This motivates us to examine whether computation of PCA in the feature space can be reduced to computation with the lower-dimensional matrix $\Phi^T\Phi$. The answer is actually yes. The key is to notice that, despite the dimension of the feature space, every eigenvector $v \in \mathbb{R}^M$ of $\Phi\Phi^T$ associated with a non-zero eigenvalue is always in the span of the matrix Φ:[10]

$$\Phi\Phi^T v = \lambda v \quad \Leftrightarrow \quad v = \Phi(\lambda^{-1}\Phi^T v) \ \in \text{range}(\Phi). \tag{2.58}$$

We define the vector $w \doteq \lambda^{-1}\Phi^T v \in \mathbb{R}^N$. Obviously $\|w\|^2 = \lambda^{-1}$. It is straightforward to check that w is an eigenvector of $\Phi^T\Phi$ for the same eigenvalue λ. Once such a w is computed from $\Phi^T\Phi$, we can recover the corresponding v in the feature space as:

$$v = \Phi w. \tag{2.59}$$

Therefore the ith nonlinear principal component of $\boldsymbol{x}$ under the map $\phi(\cdot)$ can be computed as:

$$y_i \doteq v_i^T\phi(\boldsymbol{x}) = w_i^T\Phi^T\phi(\boldsymbol{x}) \ \in \mathbb{R}, \tag{2.60}$$

where $w_i \in \mathbb{R}^M$ is the ith leading eigenvector of $\Phi^T\Phi$.

2.4.2 Kernel PCA

One should notice a very interesting feature about the above NLPCA method. Entries of both the matrix $\Phi^T\Phi$ and the vector $\Phi^T\phi(\boldsymbol{x})$ (in the expression for y_i) are all inner products of two features, i.e., of the form $\phi(\boldsymbol{x})^T\phi(\boldsymbol{y})$. In other words, computation of the principal components involves only inner products of the features. In the machine learning literature, one defines the "kernel function" of two vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^D$ to be the inner product of their features

$$k(\boldsymbol{x}, \boldsymbol{y}) \doteq \phi(\boldsymbol{x})^T\phi(\boldsymbol{y}) \ \in \mathbb{R}. \tag{2.61}$$

The so-defined function $k(\cdot,\cdot)$ is a symmetric semi-positive definite function in $\boldsymbol{x}$ and $\boldsymbol{y}$.[11] The entries of the matrix $\Phi^T\Phi$ are nothing but $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$.

As a consequence of our discussion above, one can perform nonlinear principal component analysis as long as a (semi-positive definite) kernel function is given. One does not have to explicitly define and evaluate the map $\phi(\cdot)$. In fact, given any (positive-definite) kernel function, according to a fundamental result in functional analysis, one can in principle decompose the kernel and recover the associated map $\phi(\cdot)$ if one wishes to.

[10] The remaining $M - N$ eigenvectors of $\Phi\Phi^T$ are associated with the eigenvalue zero.

[11] A function $k(\boldsymbol{x}, \boldsymbol{y})$ is semi-positive definite if $\int\int_{\mathbb{R}^D} f(\boldsymbol{x})k(\boldsymbol{x}, \boldsymbol{y})f(\boldsymbol{y})\, d\boldsymbol{x}d\boldsymbol{y} \geq 0$ for all smooth functions $f(\cdot)$.

Theorem 2.5 (Mercer's Theorem)**.** *Given a symmetric function $k(\boldsymbol{x}, \boldsymbol{y})$ with $|k(\cdot, \cdot)| \leq K$ for some K, if the linear operator $\mathcal{L} : L^2(\mathbb{R}^D) \to L^2(\mathbb{R}^D)$:*

$$\mathcal{L}(f)(\boldsymbol{x}) \doteq \int_{\mathbb{R}^D} k(\boldsymbol{x}, \boldsymbol{y}) f(\boldsymbol{y}) d\boldsymbol{y} \tag{2.62}$$

is semi-positive definite, then:

- *The operator $\mathcal{L}$ has an eigenvalue-eigenfunction decomposition $\{(\lambda_i, \phi_i(\cdot)\}$ such that $\sum_i |\lambda_i| < \infty$ and $|\phi_i(\cdot)| < K_i$ for some K_i.*
- *The kernel $k(\boldsymbol{x}, \boldsymbol{y}) = \sum_i \lambda_i \phi_i(\boldsymbol{x}) \phi_i(\boldsymbol{y})$ for almost all $(\boldsymbol{x}, \boldsymbol{y})$.*[12]

The interested readers may refer to [Mercer, 1909] for a proof of the theorem. One important reason for computing with the kernel function is because when the dimension of the feature space is high (sometimes even infinite), the computation of features and their inner products is expensive. But for many popular choices of embedding, the evaluation of the kernel function can be much simpler.

Example 2.6 (Examples of Kernels). There are several popular choices for the nonlinear kernel function:

$$k_1(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x}^T \boldsymbol{y})^n, \quad k_2(\boldsymbol{x}, \boldsymbol{y}) = \exp\big(-\frac{\|\boldsymbol{x} - \boldsymbol{y}\|^2}{2}\big). \tag{2.63}$$

Evaluation of such functions only involves the inner product or the difference between two vectors in the original space $\mathbb{R}^D$. This is much more efficient than evaluating the inner product in the associated feature space, whose dimension for the first kernel grows exponentially with the degree n and for the second kernel is infinite. ■

We summarize our discussion in this section as Algorithm 2.1.

Algorithm 2.1 (Nonlinear Kernel PCA).

For a given set of data points $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N] \in \mathbb{R}^{D \times N}$, and a given map $\phi(\boldsymbol{x})$ or a kernel function $k(\boldsymbol{x}, \boldsymbol{y})$:

1. Compute the inner product matrix

$$\Phi^T \Phi \;=\; \big(\phi(\boldsymbol{x}_i)^T \phi(\boldsymbol{x}_j)\big) \;\text{ or }\; \big(k(\boldsymbol{x}_i, \boldsymbol{x}_j)\big) \;\in \mathbb{R}^{N \times N}; \tag{2.64}$$

2. Compute the eigenvectors $w_i \in \mathbb{R}^N$ of $\Phi^T \Phi$:

$$\Phi^T \Phi w_i = \lambda_i w_i, \tag{2.65}$$

   and normalize $\|w_i\|^2 = \lambda_i^{-1}$;

3. For any data point $\boldsymbol{x}$, its ith nonlinear principal component is given by

$$y_i \;=\; w_i^T \Phi^T \phi(\boldsymbol{x}) \;\text{ or }\; w_i^T [k(\boldsymbol{x}_1, \boldsymbol{x}), \ldots, k(\boldsymbol{x}_N, \boldsymbol{x})]^T, \tag{2.66}$$

   for $i = 1, 2, \ldots, d$.

[12] "Almost all" means except for a zero-measure set.

2.5 Bibliographic Notes

As a matrix decomposition tool, SVD was initially developed independently from PCA in the numerical linear algebra literature, also known as the Erkart and Young decomposition [Eckart and Young, 1936, Hubert et al., 2000]. The result regarding the least-squares optimality of SVD given in Theorem 2.2 can be traced back to [Householder and Young, 1938, Gabriel, 1978]. While principal components were initially defined exclusively in a statistical sense [Pearson, 1901, Hotelling, 1933], one can show that the algebraic solution given by SVD gives asymptotically unbiased estimates of the true parameters in the case of Gaussian distributions. A more detailed analysis of the statistical properties of PCA can be found in [Jollife, 2002].

Note that PCA only infers the principal subspace (or components), but not a probabilistic distribution of the data in the subspace. Probabilistic PCA was developed to infer an explicit probabilistic distribution from the data [Tipping and Bishop, 1999b]. The data is assumed to be independent samples drawn from an unknown distribution, and the problem becomes one of identifying the subspace and the parameters of the distribution in a maximum-likelihood or a maximum-a-posteriori sense. When the underlying noise distribution is Gaussian, the geometric and probabilistic interpretations of PCA coincide [Collins et al., 2001]. However, when the underlying distribution is non Gaussian, the optimal solution to PPCA may no longer be linear. For example, in [Collins et al., 2001] PCA is generalized to arbitrary distributions in the exponential family.

PCA is obviously not applicable to data whose underlying structure is nonlinear. PCA was generalized to principal curves and surfaces by [Hastie, 1984] and [Hastie and Stuetzle, 1989]. A more general approach however is to find a nonlinear embedding map, or equivalently a kernel function, such that the embedded data would lie on a linear subspace. Such methods are referred to as nonlinear kernel PCA [Scholkopf et al., 1998]. Finding such nonlinear maps or kernels is by no means a simple problem. Learning kernels is still an active research topic in the statistical learning community.

2.6 Exercises

Exercise 2.1 (Some Properties of PCA). Let $\boldsymbol{x}$ be a random vector with covariance matrix $\Sigma_{\boldsymbol{x}}$. Consider a linear transformation of $\boldsymbol{x}$:

$$\boldsymbol{y} = W^T \boldsymbol{x}, \tag{2.67}$$

where $\boldsymbol{y} \in \mathbb{R}^d$ and W is a $D \times d$ orthogonal matrix. Let $\Sigma_{\boldsymbol{y}} = W^T \Sigma_{\boldsymbol{x}} W$ be the covariance matrix for $\boldsymbol{y}$. Show that

1. The trace of $\Sigma_{\boldsymbol{y}}$ is maximized by $W = U_d$, where U_d consists of the first d (normalized) eigenvectors of $\Sigma_{\boldsymbol{x}}$.
2. The trace of $\Sigma_{\boldsymbol{y}}$ is minimized by $W = \tilde{U}_d$, where $\tilde{U}_d$ consists of the last d (normalized) eigenvectors of $\Sigma_{\boldsymbol{x}}$.

Exercise 2.2 (Subspace Angles). Given two d-dimensional subspaces S_1 and S_2 in $\mathbb{R}^D$, define the largest subspace angle θ_1 between S_1 and S_2 to be the largest possible sharp angle ($< 90^\circ$) formed by any two vectors $u_1, u_2 \in (S_1 \cap S_2)^\perp$ with $u_1 \in \boldsymbol{S}_1$ and $u_2 \in S_2$ respectively. Let $U_1 \in \mathbb{R}^{D\times d}$ be an orthogonal matrix whose columns form a basis for S_1 and similarly U_2 for S_2. Then show that if σ_1 is the smallest non-zero singular value of the matrix $W = U_1^T U_2$, then we have

$$\cos(\theta_1) = \sigma_1. \tag{2.68}$$

Similarly, one can define the rest of the subspace angles as $\cos(\theta_i) = \sigma_i, i = 2, \ldots, d$ from the rest of the singular values of W.

Exercise 2.3 (Fixed-Rank Approximation of a Matrix). Given an arbitrary full-rank matrix $A \in \mathbb{R}^{m\times n}$, find the matrix $B \in \mathbb{R}^{m\times n}$ with a fixed rank $r < \min\{m, n\}$ such that the Frobenius norm $\|A-B\|_F$ is minimized. The Frobenius norm of a matrix M is defined to be $\|M\|_F^2 = \text{trace}(M^T M)$. (Hint: Use the SVD of A to guess the matrix B and then prove its optimality.)

Exercise 2.4 (Identification of Auto-Regressive Exogeneous (ARX) Systems). A popular model that people use to analyze a time series $\{y_t\}_{t\in\mathbb{Z}}$ is the linear auto-regressive model:

$$y_t = a_1 y_{t-1} + a_2 y_{t-2} + \cdots + a_n y_{t-n} + \varepsilon_t, \quad \forall t, y_t \in \mathbb{R}, \tag{2.69}$$

where $\varepsilon_t \in \mathbb{R}$ models the modeling error or noise and it is often assumed to be a white-noise random process. Now suppose that you are given the values of y_t for a sufficiently long period of time.

1. Show that in the noise free case, i.e. $\varepsilon_t \equiv 0$, regardless of the initial conditions, the vectors $\boldsymbol{x}_t = [y_t, y_{t-1}, \ldots, y_{t-n}]^T$ for all t lie on an n-dimensional hyperplane in $\mathbb{R}^{n+1}$. What is the normal vector to this hyperplane?
2. Now consider the case with noise. Describe how you may use PCA to identify the unknown model parameters $(a_1, a_2, \ldots, a_n)$?

Exercise 2.5 (Basis for an Image). Given a gray-level image $\boldsymbol{I}$, consider all of its $b \times b$ blocks, denoted as $\{B_i \in \mathbb{R}^{b\times b}\}$. We would like to approximate each block as a superposition of d base blocks, say $\{\hat{B}_j \in \mathbb{R}^{b\times b}\}_{j=1}^d$. That is,

$$B_i = \sum_{j=1}^{d} a_{ij}\hat{B}_j + E_i, \tag{2.70}$$

where $E_i \in \mathbb{R}^{b\times b}$ is the possible residual from the approximation. Describe how you can use PCA to identify an optimal set of d base blocks so that the residual is minimized?

In Section 1.2.1, we have seen an example in which a similar process can be applied to an ensemble of face images, where the first $d = 3$ principal components are computed for further classification. In the computer vision literature, the corresponding base images are called "eigen faces."

Exercise 2.6 (Probability of Selecting a Subset of Inliers). Imagine we have 80 samples from a four-dimensional subspace in $\mathbb{R}^5$. However, the samples are contaminated with another 20 samples that are far from the subspace. We want to estimate the subspace from randomly drawn subsets of four samples. In order to be of probability 0.95 that one of the subsets contains only inliers, what is the smallest number of subsets that we need to draw?

Exercise 2.7 (Ranking of Webpages). PCA is actually used to rank webpages on the Internet by many popular search engines. One way to see this is to view the Internet as a directed graph $G = (V, E)$, where every webpage, denoted as p_i, is a node in V, and every hyperlink from p_i to p_j, denoted as e_{ij}, are directed edges in E. We can assign each webpage p_i an "authority" score x_i that indicates how many other webpages point to it and a "hub" score y_i that indicates how many other webpages it points out to. Then, the authority score x_i depends on how many hubs point to p_i and the hub score y_i depends on how many authorities p_i points to. Let L be the adjacent matrix of the graph G (i.e. $L_{ij} = 1$ if $e_{ij} = E$), $\boldsymbol{x}$ the vector of the authority scores and $\boldsymbol{y}$ of the hub scores.

1. Justify that the following relationships hold:
$$\boldsymbol{y}' = L\boldsymbol{x}, \quad \boldsymbol{x}' = L^T\boldsymbol{y}; \quad \boldsymbol{x} = \boldsymbol{x}'/\|\boldsymbol{x}'\|, \quad \boldsymbol{y} = \boldsymbol{y}'/\|\boldsymbol{y}'\|. \tag{2.71}$$
2. Show that $\boldsymbol{x}$ is the eigenvector of L^TL and $\boldsymbol{y}$ is the eigenvector of LL^T associated with the largest eigenvalue (why not the others). Explain how $\boldsymbol{x}$ and $\boldsymbol{y}$ can be computed from the singular value decomposition of L.

In the literature, this is known as the *Hybertext Induced Topic Selection* (HITS) algorithm [Kleinberg, 1999, Ding et al., 2004]. In fact, the same algorithm can also be used to rank any competitive sports such as football teams and chess players.

Exercise 2.8 (Karhunen-Loève Transform). The Karhunen-Loève transform (KLT) can be thought as a generalization of PCA from a (finite-dimensional) random vector $\boldsymbol{x} \in \mathbb{R}^D$ to an (infinite-dimensional) random process $x(t), t \in \mathbb{R}$. When $x(t)$ is a (zero-mean) second-order stationary random process, its auto correlation function is defined to be $K(t, \tau) \doteq E[x(t)x)(\tau)]$ for all $t, \tau \in \mathbb{R}$.

1. Show that $K(t, \tau)$ has a family of orthonormal eigen-functions $\{\phi_i(t)\}_{i=1}^{\infty}$ that are defined as
$$\int K(t, \tau)\phi_i(\tau)\, d\tau = \lambda_i\phi_i(t), \quad i = 1, 2, \ldots. \tag{2.72}$$
(Hint: First show that $K(t, \tau)$ is a positive definite function and then use Mercer's Theorem.)
2. Show that with respect to the eigen-functions, we original random process can be decomposed as
$$x(t) = \sum_{i=1}^{n} x_i\phi_i(t), \tag{2.73}$$
where $\{x_i\}_{i=1}^{\infty}$ are a set of uncorrelated random variables.