Chapter 2
Data Modeling with a Single Subspace

In this chapter, we give a brief review of principal component analysis (PCA), i.e., the method for finding a dominant 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. In this section, we first show that the singular value decomposition (SVD) provides an optimal solution to PCA. Both the geometric and statistical formulation of PCA will be introduced and their equivalence will be established. When the dimension of the subspace is unknown, we introduce some conventional model selection methods to determine the number of principal components. When the samples contain outliers and incomplete data points, we review some robust statistical techniques that help resolve these difficulties. Finally, some nonlinear extensions to PCA such as nonlinear PCA and kernel PCA will also be 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 $X = \{x_1, x_2, \ldots, x_N\}$ 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 Geometric Approach to PCA

We first examine the more intuitive geometric approach to PCA. That is, one tries to find an (affine) subspace that fits the given data points. Let us assume for now that the dimension of the subspace \( d \) is known. Then every point \( x_i \) on a \( d \)-dimensional affine subspace in \( \mathbb{R}^D \) can be represented as

\[
x_i = x_0 + U_d y_i, \quad i = 1, \ldots, N
\]

where \( x_0 \in \mathbb{R}^D \) is an (ny) fixed point in the subspace, \( U_d \) is a \( D \times d \) matrix with \( d \) orthonormal column vectors, and \( y_i \in \mathbb{R}^d \) is simply the vector of new coordinates of \( x_i \) in the subspace. Notice that there is some redundancy in the above representation due to the arbitrariness in the choice of \( x_0 \) in the subspace. More precisely, for any \( y_0 \in \mathbb{R}^d \), we can re-represent \( x_i \) as \( x_i = (x_0 + U_d y_i) + U_d (y_i - y_0) \).

Therefore, we need some additional constraints in order to end up with a unique solution to the problem of finding an affine subspace to fit the data. A common constraint is to impose that the mean of \( y_i \) is zero:

\[
\bar{y} = \frac{1}{N} \sum_{i=1}^N y_i = 0. \quad (2.2)
\]

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 error between \( x_i \) and its projection on the subspace, i.e.,

\[
\min_{x_0, U_d, \{y_i\}} \sum_{i=1}^N \|x_i - x_0 - U_d y_i\|^2, \quad \text{s.t. } U_d^T U_d = I \text{ and } \bar{y} = 0. \quad (2.3)
\]

Differentiating this function with respect to \( x_0 \) and \( y_i \) (assuming \( U_d \) is fixed) and setting the derivatives to be zero, we obtain the relations:

\[
\bar{x}_0 = \bar{x} = \frac{1}{N} \sum_{i=1}^N x_i; \quad \bar{y}_i = U_d^T (x_i - \bar{x}). \quad (2.4)
\]

The vector \( \bar{y}_i \in \mathbb{R}^d \) is simply the coordinates of the projection of \( x_i \in \mathbb{R}^D \) in the subspace \( S \). We may call such \( \bar{y}_i \) the “geometric principal components” of \( x_i \).

Then the original objective becomes one of finding an orthogonal matrix \( U_d \in \mathbb{R}^{D \times d} \) that minimizes

\[
\min_{U_d} \sum_{i=1}^N \| (x_i - \bar{x}) - U_d U_d^T (x_i - \bar{x}) \|^2. \quad (2.5)
\]

---

1. In the statistical setting, \( x_i \) and \( y_i \) will be samples of two random variables \( x \) and \( y \), respectively. Then this constraint is equivalent to setting their means to be zero.

2. Which are the necessary conditions for the minima.

3. As we will soon see in the next section, it coincides with the traditional principal components defined in a statistical sense.
2.1. Principal Component Analysis (PCA) 21

Note that this is a restatement of the original problem with the mean $\bar{x}$ 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 and the solution for $U_d$ remains the same. The following theorem gives a constructive solution to the optimal solution $\hat{U}_d$.

**Theorem 2.1 (PCA via SVD).** Let $X = [x_1, x_2, \ldots, x_N] \in \mathbb{R}^{D \times N}$ be the matrix formed by stacking the (zero-mean) data points as its column vectors. Let $X = U \Sigma V^T$ be the singular value decomposition (SVD) of the matrix $X$. Then for any given $d < D$, a solution to PCA, $\hat{U}_d$ is exactly the first $d$ columns of $U$; and $\hat{y}_i$ is the $i$th column of the top $d \times N$ submatrix $\Sigma_d V_d^T$ of the matrix $\Sigma V^T$.

**Proof.** Note that the problem

$$
\min_{U_d} \sum_{i=1}^{N} \|x_i - U_d U_d^T x_i\|^2
$$

is equivalent to

$$
\min_{U_d} \sum_{i=1}^{N} \text{trace} \left[ (x_i - U_d U_d^T x_i)(x_i - U_d U_d^T x_i)^T \right]
$$

$$
\Leftrightarrow \min_{U_d} \text{trace} \left[ (I - U_d U_d^T)XX^T \right],
$$

where, for the second equivalence, we use the facts $\text{trace}(AB) = \text{trace}(BA)$, $U_d U_d^T U_d U_d^T = U_d U_d^T$, and $XX^T = \sum_{i=1}^{N} x_i x_i^T$ to simplify the expression. Substitute $X = U \Sigma V^T$ into the above expression, the problem becomes

$$
\min_{U_d} \text{trace} \left[ (I - U^T U_d U_d^T U) \Sigma^2 \right].
$$

Let $\sum_{i=1}^{D} \sigma_i^2 e_i e_i^T$ be the dyadic decomposition of the diagonal matrix $\Sigma^2$. Since $U_d^T U$ is an orthogonal matrix, the above minimization is the same as

$$
\min_{U_d} \sum_{i=1}^{D} \text{trace} \left[ (\sigma_i e_i - U^T U_d U_d^T U \sigma_i e_i)(\sigma_i e_i - U^T U_d U_d^T U \sigma_i e_i)^T \right]
$$

$$
\Leftrightarrow \min_{U_d} \sum_{i=1}^{D} \sigma_i^2 \| (I - U^T U_d U_d^T U) e_i \|^2.
$$

Because $U_d$ is an orthogonal matrix of rank $d$ so is $U_d^T U$ so that $I - U^T U_d U_d^T U$ is an idempotent matrix of rank $D - d$, so that the $D$ terms $\| (I - U^T U_d U_d^T U) e_i \|^2$ always sum up to a constant $D - d$, and $\sigma_1^2 \geq \sigma_2^2 \geq \cdots \geq \sigma_D^2$ are ordered. Therefore, the minimum is achieved when the $d$ terms associated with the higher weights $\sigma_1^2, \ldots, \sigma_D^2$ become zero. This happens only when $\hat{U}_d$ consists of the first $d$ columns of $U$. The rest of the theorem then easily follows.

---

*Here $e_i \in \mathbb{R}^D$ is the standard $i$th base vector of $\mathbb{R}^D$, i.e., its $i$th entry is 1 and others are 0.*
When there are repeated singular values with $\sigma_d = \sigma_{d+1}$, there is a loss of uniqueness of the solution corresponding to the principal components. □

According to the theorem, the SVD gives an optimal solution to the PCA problem. The resulting matrix $\hat{U}_d$ (together with the mean $\bar{x}$ if the data is not zero-mean) provides a geometric description of the dominant subspace structure for all the points; and the columns of the matrix $\Sigma_d V_d^T = [\hat{y}_1, \ldots, \hat{y}_N] \in \mathbb{R}^{d \times N}$, i.e., the principal components, give a more compact representation for the points $X = [x_1, \ldots, x_N] \in \mathbb{R}^{D \times N}$, as $d$ is typically much smaller than $D$.

### 2.1.2 A Statistical View of PCA

Historically PCA was first formulated in a statistical setting: to estimate the principal components of a multivariate random variable $x$ from given sample points $\{x_i\}$ [Hotelling, 1933]. For a multivariate random variable $x \in \mathbb{R}^D$ and any $d < D$, the $d$ "principal components" are defined to be $d$ uncorrelated linear components of $x$:

$$y_i = u_i^T x \in \mathbb{R}, \quad i = 1, \ldots, d$$

(2.7)

for some $u_i \in \mathbb{R}^D$ such that the variance of $y_i$ is maximized subject to

$$u_i^T u_i = 1, \quad \text{Var}(y_1) \geq \text{Var}(y_2) \geq \cdots \geq \text{Var}(y_d).$$

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

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

(2.8)

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

**Theorem 2.2** (Principal Components of a Random Variable). The first $d$ principal components of a multivariate random variable $x$ are given by the $d$ leading eigenvectors of its covariance matrix $\Sigma_x = E[x x^T]$.

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

$$\text{Var}(u^T x) = E[(u^T x)^2] = E[u^T x x^T u] = u^T \Sigma_x u.$$

Then to find the first principal component, the above minimization (2.8) is equivalent to

$$\max_{u_1 \in \mathbb{R}^D} u_1^T \Sigma_x u_1, \quad \text{s.t.} \quad u_1^T u_1 = 1.$$  

(2.9)

Solving the above constrained minimization problem using Lagrange multiplier method, we obtain the necessary condition for $u_1$ to be an extrema:

$$\Sigma_x u_1 = \lambda u_1$$  

(2.10)

---

3From 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."
for some Lagrange multiplier $\lambda \in \mathbb{R}$, and the the associated extremum value is $u_1^T \Sigma \hat{u}_1 = \lambda$. Obviously, the optimal solution $u_1^T$ is exactly the eigenvector associated with the largest eigenvalue of $\Sigma_x$.

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

$$ E[(u_1^T \mathbf{x})(u_i^T \mathbf{x})] = E[u_1^T \mathbf{x} \mathbf{x}^T u_i] = u_1^T \Sigma \mathbf{x} u_i = \lambda_1 u_1^T u_i = 0. $$

That is, $u_2, \ldots, u_d$ are all orthogonal to $u_1$. Following the proof for the optimality of $u_1$, $u_2$ is then the leading eigenvector of $\Sigma_x$ restricted to the orthogonal complement of $u_1$. Overall, $u_2$ is the second leading eigenvector of $\Sigma_x$. Inductively, one can show for the rest of the principal components.

Normally, we do not know $\Sigma_x$ and can only estimate it from the given $N$ samples $\mathbf{x}_i$. It is known from statistics that

$$ \hat{\Sigma}_x = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_i \mathbf{x}_i^T = \frac{1}{N} \mathbf{X} \mathbf{X}^T $$

(2.11)

is an asymptotically unbiased estimate of the covariance matrix $\Sigma_x$. The eigenvectors of $\hat{\Sigma}_x$, or equivalently those of $\mathbf{X} \mathbf{X}^T$, lead to the "sample principal components":

$$ \hat{\mathbf{y}}_i = \hat{\mathbf{u}}_i^T \mathbf{x}, \quad \text{s.t.} \quad \hat{\Sigma}_x \hat{\mathbf{u}}_i = \lambda \hat{\mathbf{u}}_i \text{ and } \hat{\mathbf{u}}_i^T \hat{\mathbf{u}}_i = 1. $$

(2.12)

One can show that, if $\mathbf{x}$ is Gaussian, then every eigenvector $u$ of $\hat{\Sigma}_x$ is an asymptotically unbiased estimate for the corresponding eigenvector of $\Sigma_x$ [Jolliffe, 1986].

**Theorem 2.3** (Equivalence of Geometric and Sample Principal Components). Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N] \in \mathbb{R}^{D \times N}$ be the data matrix (with $\bar{\mathbf{x}} = 0$). The vectors $\hat{\mathbf{u}}_1, \hat{\mathbf{u}}_2, \ldots, \hat{\mathbf{u}}_d \in \mathbb{R}^D$ associated with the $d$ sample principal components for $\mathbf{X}$ are exactly the columns of the matrix $U_d \in \mathbb{R}^{D \times d}$ that minimizes the least-squares error (2.6).

**Proof.** The proof is simple. Notice that if $\mathbf{X}$ has the singular value decomposition $\mathbf{X} = U \Sigma V^T$, then $\mathbf{X} \mathbf{X}^T = U \Sigma^2 U^T$ is the eigenvalue decomposition of $\mathbf{X}$. If $\Sigma$ is ordered, then the first $d$ columns of $U$ are exactly the leading $d$ eigenvectors of $\mathbf{X} \mathbf{X}^T$, 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

---

6The reason for this is that both $u_1$ and its orthogonal complement $u_1^\perp$ are invariant subspaces of $\Sigma_x$. 
a more thorough analysis of the statistical properties of PCA, we refer the reader to the classical book [Jollife, 1986].

2.1.3 Determining the Number of Principal Components

Notice that SVD of the noisy data matrix $X$ 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 $X$, 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.

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). The dimension $d$ of the subspace $S$ can be viewed as a natural measure of the complexity of the model; and the sum of squares of the remaining singular values $\sum_{i=d+1}^{D} \sigma_i^2$ is exactly the modeling error $\sum_{i=1}^{N} \|x_i - \bar{x}_i\|^2$. The leading singular value $\sigma_{d+1}^2$ of the remaining ones is a good index of the modeling error. Therefore, one can seek for a model that balances between $d$ and $\sigma_{d+1}^2$ by minimizing an objective function of the form:

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

for some proper positive weights $\alpha, \beta > 0$. Another somewhat similar and popular objective function that people often use to determine the rank $d$ of the noise matrix $X$ from its singular values is (e.g., [Kanatani and Matsunaga, 2002a]):

$$J_{rank}(d) \doteq \frac{\sigma_{d+1}^2}{\sum_{i=1}^{d} \sigma_i^2} + \kappa d.$$  \hfill (2.14)

In this book, unless stated otherwise, this will be the criterion of choice when we try to determine the rank of a matrix corrupted by noise.

In general, the ordered singular values of the data matrix $X$ 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 value right after the "correct" dimension $d$, 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 (2.13) for a range of values for $\alpha$ and $\beta$, or (2.14) for a range of $\kappa$.

A model can also be selected from the Scree graph in another way. If, instead of the dimension $d$, a tolerance $\tau$ 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].
2.2 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.

2.2.1 Outliers

In practice, it is often the case that a small subset 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 is an outlier. 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 really tell which case is really true for a certain “outlier” sample point. But these different interpretations of outliers lead to different approaches to “detect” (and subsequently eliminate) outliers.

Error-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 result in too large modeling errors. In PCA, if we assume the samples are drawn from a Gaussian distribution, the probability of a sample is approximately inversely proportional to its (squared)
geometric distance to the identified subspace. Hence using either the probability or the geometric error ends up similar criteria for outlier detection. For instance, from the SVD $X = U\Sigma V^T$, we know

$$y_i = u_i^T x, \quad i = 1, \ldots, D$$

(2.15)

are the new coordinates of each point $x$ with respect to a new orthonormal coordinate frame specified by $U$. The first $d$ terms $y_1, \ldots, y_d$ are the principal components (the new coordinates of $\hat{x}$ in the principal subspace); and the last $D - d$ terms $y_{d+1}, \ldots, y_D$ are the new coordinates of $x - \hat{x}$ in the orthogonal complement of the principal subspace. Clearly the geometric modeling error is

$$\|x - \hat{x}\|^2 = y_{d+1}^2 + y_{d+2}^2 + \cdots + y_D^2.$$  

(2.16)

Then one may simply call a point $x$ an “outlier” if

$$C(x) \doteq y_{d+1}^2 + y_{d+2}^2 + \cdots + y_D^2 \geq \tau$$

(2.17)

for some chosen threshold $\tau > 0$. However, this criterion does not take into account how the rest of the points are distributed away from the principal subspace. To fix this oversight, notice that as a random variable, $y_i$ has the (estimated) variance $\sigma_i^2$. Therefore, if we assume $x - \hat{x}$ has a Gaussian distribution (and so are $y_i$), then we have

$$-\log p(x - \hat{x}) \propto y_{d+1}^2/\sigma_{d+1}^2 + y_{d+2}^2/\sigma_{d+2}^2 + \cdots + y_D^2/\sigma_D^2.$$  

(2.18)

Therefore, we should use the following criterion

$$C_n(x) \doteq y_{d+1}^2/\sigma_{d+1}^2 + y_{d+2}^2/\sigma_{d+2}^2 + \cdots + y_D^2/\sigma_D^2 \geq \tau$$

(2.19)

to detect outliers. The left hand is nothing but the geometric error normalized by the respective variance of $x$ in the direction of each eigenvector. When the variances $\sigma_{d+1}^2, \ldots, \sigma_D^2$ are approximately the same, the two criteria (2.17) and (2.19) coincide. In practice, we find either criterion (2.17) or (2.19) can be very effective in detecting outliers, depending on the nature of the data.

**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 a model from the data. 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 proper subset of the data. This is possible when the number of data points required for a unique solution for model estimation can be 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 correct model. Thus, one should try on many different subsets:

$$X_1, X_2, \ldots, X_n \subset X.$$  

(2.20)

The rationale is that if the percentage of outliers is relatively small, one of the trial subsets, say $X_i$, 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 $X_i$ gives a consistent estimate of the subspace $\hat{U}_d(X_i)$ if the following criterion is satisfied:

$$\# \{ x \in X : \| x - \hat{U}_d(X_i) \| \leq \tau \} \geq N \cdot \delta \%,$$  \hspace{1cm} (2.21)

for some error threshold $\tau > 0$ and some percentage threshold $\delta$ (normally larger than 50 percent). This scheme is typically called Random Sample Consensus (RANSAC), and it normally improves the robustness of the model selection/estimation to outliers. As a word of caution, in practice, in order to design a successful RANSAC algorithm, one needs to carefully choose a few key parameters: the size of each subset, the number of subsets, and the consensus criterion.\(^7\)

There is a vast amount of literature on RANSAC-type algorithms, and for a more thorough introduction, we refer interested readers to [Fischler and Bolles, 1981].

**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:

$$I(x_i, U_d) = d(\hat{U}_d, \hat{U}_{d(i)}),$$  \hspace{1cm} (2.22)

where $d(\cdot, \cdot)$ is a proper distance measure between the subspace basis estimated with the $i$th sample $\hat{U}_d$ and that without the $i$th sample $\hat{U}_{d(i)}$.\(^8\) The larger the difference is, the larger is the influence and more likely is the sample $x_i$ an outlier. Thus, we may eliminate a sample $x$ as an outlier if

$$I(x, U_d) \geq \tau$$  \hspace{1cm} (2.23)

for some threshold $\tau > 0$. However, this method does not come without an extra cost. We need to compute the principal components for $N$ times: one time for all the samples together and another $N - 1$ times with one sample eliminated from each time. There is also a vast amount of literature on sample influence of PCA, we refer interested readers to [Jollife, 2002].

**2.2.2 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 $x = [x_1, x_2, \ldots, x_D]^T$, we

---

\(^7\)That is, the criterion that verifies whether each point is consistent with the model derived from the subset.

\(^8\)One can choose either the largest subspace angle between the two bases or the sum of squares of all the subspace angles.
mean that some of its entry or entries are missing or unspecified. For instance, if
the \( x_i \)-entry is missing from \( x \), it means that we know \( x \) only up to a line in \( \mathbb{R}^D \):
\[
x \in L \doteq \{ [x_1, \ldots, x_{i-1}, t, x_{i+1}, \ldots, x_D]^T, t \in \mathbb{R} \}. \tag{2.24}
\]
One should be aware that an incomplete data point is in nature rather different
from a noisy data point or an outlier.\(^9\) 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
\( L \), then knowing enough number of such lines, the principal subspace can be
uniquely determined. 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
key observation here is that the incomplete data point \( x \) is just as good as any
point on the line \( L \). Hence it is natural to choose a representative \( \tilde{x} \in L \) that is
the closest to the principal subspace. Let us denote \( B_d = I - U_d U_d^T \), then the closest
point \( \tilde{x}^* = [x_1, \ldots, x_{i-1}, t^*, x_{i+1}, \ldots, x_D]^T \) on \( L \) to the principal subspace can
be found by minimizing the following quadratic function in \( t \):
\[
t^* = \arg \min_{t} (x^T B^T B x). \tag{2.25}
\]
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 \( \tilde{x}^* \) on the principal subspace is to give
a rank-\( d \) approximation of the entire data set containing both complete and in-
complete data points. Mathematically, PCA with incomplete data is equivalent to
find a rank-\( d \) approximation/factorization of the data matrix \( X \) 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].

### 2.3 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 sig-
nificant 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.

#### 2.3.1 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

\(^9\)One can view incomplete data points as a very special type of noisy data points which have infinite
uncertainty only in certain directions.
nonlinear transformation (more precisely, usually an embedding):

\[ \phi(\cdot) : \mathbb{R}^D \rightarrow \mathbb{R}^M, \]

\[ x \mapsto \phi(x), \]

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

Define the matrix \( \Phi = [\phi(x_1), \ldots, \phi(x_N)] \in \mathbb{R}^{M \times N} \). The principal components in the feature space are given by the eigenvectors of the sample covariance matrix\(^{10}\)

\[
\Sigma_{\phi(x)} = \frac{1}{N} \sum_{i=1}^{N} \phi(x_i)\phi(x_i)^T = \frac{1}{N} \Phi \Phi^T \in \mathbb{R}^{M \times M}.
\]

Let \( v_i \in \mathbb{R}^M \) to be the eigenvectors:

\[
\Sigma_{\phi(x)} v_i = \lambda_i v_i, \quad i = 1, \ldots, M. \tag{2.26}
\]

Then the \( d \) “nonlinear principal components” of every data point \( x \) are given by

\[
y_i = v_i^T \phi(x) \in \mathbb{R}, \quad i = 1, \ldots, d. \tag{2.27}
\]

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 Mixtures of Subspaces).** As we will see later in this book, if the data points belong to multiple subspaces, then a natural choice of the transformation \( \phi(\cdot) \) is the Veronese map:

\[
\nu_n(\cdot) : \mathbb{R}^D \mapsto \mathbb{R}^{nD},
\]

\[
(x_1, \ldots, x_D) \mapsto (x_1, x_1^{n-1} x_2, \ldots, x_D^n)
\]

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.

\[^{10}\text{In principle, we should use the notation } \hat{\Sigma}_{\phi(x)} \text{ to indicate that it is the estimate of the actual covariance matrix. But for simplicity, we will drop the hat in the sequel and simply use } \Sigma_{\phi(x)}. \text{ The same goes for the eigenvectors and the principal components.}\]**
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.

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 \( \Phi \):\(^{11}\)

\[
\Phi \Phi^T v = \lambda v \iff v = \Phi(\lambda^{-1} \Phi^T v) \in \text{range}(\Phi).
\]  

(2.28)

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

\[
v = \Phi w.
\]  

(2.29)

Therefore the \( i \)th nonlinear principal component of \( x \) under the map \( \phi(\cdot) \) can be computed as:

\[
y_i = v_i^T \phi(x) = w_i^T \Phi^T \phi(x) \in \mathbb{R},
\]  

(2.30)

where \( w_i \in \mathbb{R}^M \) is the \( i \)th leading eigenvector of \( \Phi^T \Phi \).

### 2.3.2 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 \( \Phi^T \phi(x) \) (in the expression for \( y_i \)) are all inner products of two features, i.e., of the form \( \phi(x)^T \phi(y) \). 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 \( x, y \in \mathbb{R}^D \) to be the inner product of their features

\[
k(x, y) = \phi(x)^T \phi(y) \in \mathbb{R}.
\]  

(2.31)

The so-defined function \( k(\cdot, \cdot) \) is a symmetric semi-positive definite function in \( x \) and \( y \).\(^{12}\) The entries of the matrix \( \Phi^T \Phi \) are nothing but \( k(x_i, x_j) \).

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:

---

\(^{11}\)The remaining \( M - N \) eigenvectors of \( \Phi \Phi^T \) are associated with the eigenvalue zero.

\(^{12}\)A function \( k(x, y) \) is semi-positive definite if \( \int_{\mathbb{R}^D} f(x)k(x, y)f(y) \, dx \, dy \geq 0 \)
Theorem 2.5 (Mercer’s Theorem). Given a symmetric function \( k(x, y) \) with \( |k(\cdot, \cdot)| \leq K \) for some \( K \), if the linear operator \( \mathcal{L} : L^2(\mathbb{R}^D) \to L^2(\mathbb{R}^D) \):

\[
\mathcal{L}(f)(x) = \int_{\mathbb{R}^D} k(x, y) f(y) dy
\]

is semi-positive definite, then:

- The operator \( \mathcal{L} \) has an eigenvalue-eigenvector decomposition \( \{ (\lambda_i, \phi_i(\cdot)) \} \) such that \( \sum_i |\lambda_i| < \infty \) and \( |\phi_i(\cdot)| < K \) for some \( K \).

- The kernel \( k(x, y) = \sum_i \lambda_i \phi_i(x) \phi_i(y) \) for almost all \((x, y)\).

The interested readers may refer to [10] 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:

\[
k_1(x, y) = (x^T y)^n, \quad k_2(x, y) = \exp \left( -\frac{\|x - y\|^2}{2} \right).
\]

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.

2.4 Bibliographic Notes

As a matrix decomposition tool, SVD was initially developed independently from PCA in the numerical linear algebra literature, also known as the Eckart and Young decomposition [Eckart and Young, 1936, Hubert et al., 2000]. The result regarding the least-squares optimality of SVD given in Theorem 2.1 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.

\(^{13} \text{Almost all means except for a zero-measure set.} \)
Algorithm 2.1 (Nonlinear Kernel PCA).

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

1. Compute the inner product matrix
   \[
   \Phi^T \Phi = (\phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)) \text{ or } (k(\mathbf{x}_i, \mathbf{x}_j)) \in \mathbb{R}^{N \times N};
   \]
   \( \Phi \) is the kernel matrix.

2. Compute the eigenvectors \( \mathbf{w}_i \in \mathbb{R}^N \) of \( \Phi^T \Phi \):
   \[
   \Phi^T \Phi \mathbf{w}_i = \lambda_i \mathbf{w}_i,
   \]
   and normalize \( \|\mathbf{w}_i\|^2 = \lambda_i^{-1} \);

3. For any data point \( \mathbf{x} \), its \( i \)th nonlinear principal component is given by
   \[
   y_i = \mathbf{w}_i^T \Phi \phi(\mathbf{x}) \text{ or } \mathbf{w}_i^T [k(\mathbf{x}_1, \mathbf{x}), \ldots, k(\mathbf{x}_N, \mathbf{x})]^T,
   \]
   for \( i = 1, \ldots, d \).

[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 are by no means simple problems. Learning kernels is still an active research topic in the machine learning community.