Principle Component Analysis

Dimensionality Reduction with Principal Component Analysis

Working directly with high-dimensional data, such as images, comes with some difficulties:

- It is hard to analyze
- interpretation is difficult,
- visualization is nearly impossible, and
- (from a practical point of view) storage of the data vectors can be expensive.

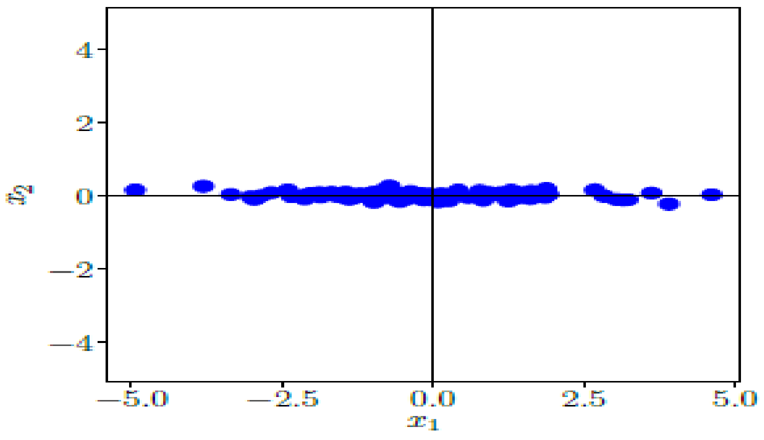
However, high-dimensional data often has properties that we can exploit.

- High-dimensional data is often over complete, i.e., many dimensions are redundant and can be explained by a combination of other dimensions.
- Dimensionality reduction exploits structure and correlation and allows us to work with a more compact representation of the data.
- Dimensionality reduction as a compression technique, similar to jpeg or mp3, which are compression algorithms for images and music.

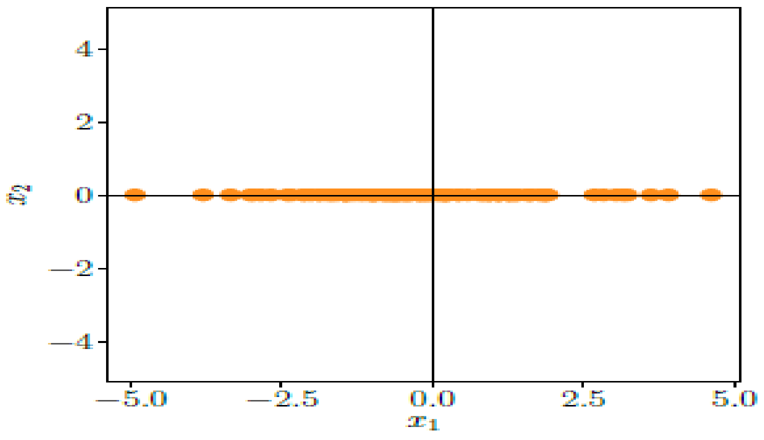
Principle Component Analysis

- An algorithm for linear *dimensionality reduction.*
- PCA, proposed by Pearson Dimensionality Reduction (1901) and Hotelling (1933).
- One of the most commonly used techniques for data compression and data visualization.
- It is also used for the identification of simple patterns, latent factors, and structures of high-dimensional data.
- In signal processing community, PCA is also known as the *Karhunen-Lo`eve* transform.
- Dimensionality reduction generally exploits a property of high-dimensional data (e.g., images) that it often lies on a low-dimensional subspace.

Examples of Dimensional Reduction



(a) Dataset with x_1 and x_2 coordinates.

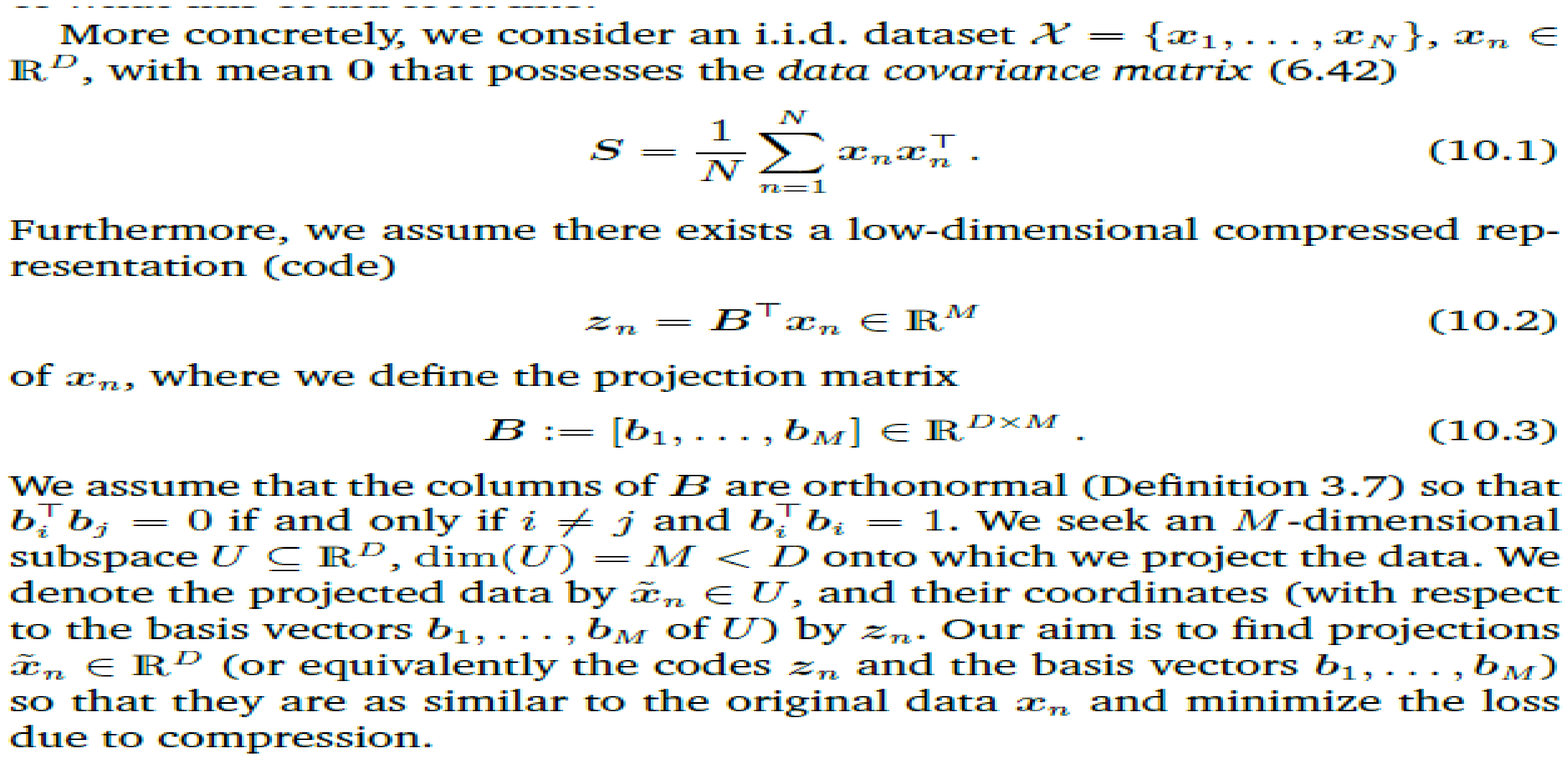


(b) Compressed dataset where only the x_1 coordinate is relevant.

- Dimensional data (e.g., images) that it often lies on a low-dimensional subspace.
- Figure gives an illustrative example in two dimensions.
- Although, the data in Figure (a) does not quite lie on a line the data does not vary much in the x2-direction, so that we can express it as if it were on a line – with nearly no loss;
- Figure (b). To describe the data in Figure 10.1(b), only the x1-coordinate is required, and the data lies in a one-dimensional subspace of R2.

Problem Setting

- In PCA, we are interested in finding projections ~xn of data points xn that are as similar to the original data points as possible, but which have a significantly lower intrinsic dimensionality.



More concretely, we consider an i.i.d. dataset $\mathcal{X} = \{x_1, \dots, x_N\}$, $x_n \in \mathbb{R}^D$, with mean 0 that possesses the *data covariance matrix* (6.42)

$$S = \frac{1}{N} \sum_{n=1}^{N} x_n x_n^\top . \tag{10.1}$$

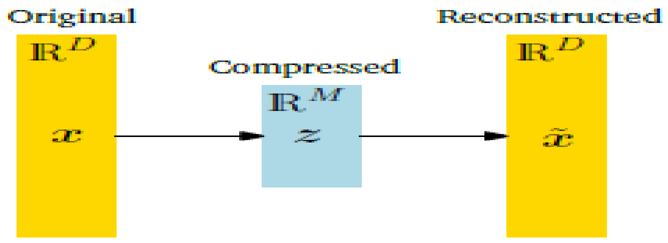
Furthermore, we assume there exists a low-dimensional compressed representation (code)

$$z_n = B^\top x_n \in \mathbb{R}^M \tag{10.2}$$

of x_n, where we define the projection matrix

$$B := [b_1, \dots, b_M] \in \mathbb{R}^{D \times M} . \tag{10.3}$$

We assume that the columns of B are orthonormal (Definition 3.7) so that $b_i^\top b_j = 0$ if and only if $i \neq j$ and $b_i^\top b_i = 1$. We seek an M-dimensional subspace $U \subseteq \mathbb{R}^D$, $\dim(U) = M < D$ onto which we project the data. We denote the projected data by $\tilde{x}_n \in U$, and their coordinates (with respect to the basis vectors $b_1, \dots, b_M$ of U) by z_n. Our aim is to find projections $\tilde{x}_n \in \mathbb{R}^D$ (or equivalently the codes z_n and the basis vectors $b_1, \dots, b_M$) so that they are as similar to the original data x_n and minimize the loss due to compression.



Graphical illustration of PCA. In PCA, we find a compressed version z of original data x. The compressed data can be reconstructed into ~x, which lives in the original data space, but has an intrinsic lower-dimensional representation than x.

- where z represents the lower-dimensional representation of the compressed data ~x and plays the role of a bottleneck.
- which controls how much information can flow between x and ~x.
- In PCA, we consider a linear relationship between the original data x and its low-dimensional code z so that z = B>x and ~x = Bz for a suitable matrix B.

The linear mapping represented by B can be thought of as a decoder, which maps the low-dimensional code $z \in \mathbb{R}^M$ back into the original data space $\mathbb{R}^D$. Similarly, $B^\top$ can be thought of an encoder, which encodes the original data x as a low-dimensional (compressed) code z.

Example 10.1 (Coordinate Representation/Code)
Consider $\mathbb{R}^2$ with the canonical basis $e_1 = [1,0]^\top$, $e_2 = [0,1]^\top$. From

Chapter 2, we know that $x \in \mathbb{R}^2$ can be represented as a linear combination of these basis vectors, e.g.,

$$\begin{bmatrix} 5 \\ 3 \end{bmatrix} = 5e_1 + 3e_2 \,. \tag{10.4}$$

However, when we consider vectors of the form

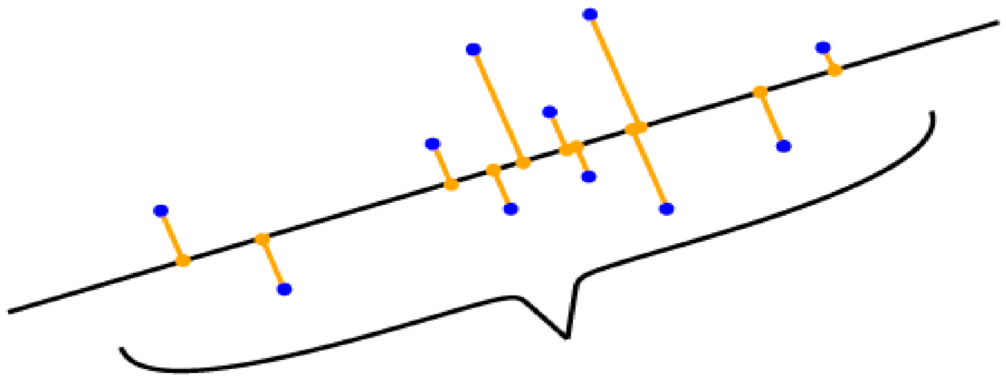
$$\tilde{x} = \begin{bmatrix} 0 \\ z \end{bmatrix} \in \mathbb{R}^2 \,, \quad z \in \mathbb{R} \,, \tag{10.5}$$

they can always be written as $0e_1 + ze_2$. To represent these vectors it is sufficient to remember/store the *coordinate/code* z of $\tilde{x}$ with respect to the e_2 vector.

More precisely, the set of $\tilde{x}$ vectors (with the standard vector addition and scalar multiplication) forms a vector subspace U (see Section 2.4) with $\dim(U) = 1$ because $U = \text{span}[e_2]$.

Maximum Variance Perspective

- Information contained in the data is interpreted by how space is filled by dataset.
- Variance is an indicator of the spread of the data.
- PCA as a dimensionality reduction algorithm that maximizes the variance in the low-dimensional representation.
- Retaining most information after data compression is equivalent to capturing the largest amount of variance in the low-dimensional code.
- Our aim is to find a matrix B (see (10.3)) that retains as much information as possible when compressing data by projecting it onto the subspace spanned by the columns b_1; : : : ; b_M of B.



10.2.1 *Direction with Maximal Variance*

We maximize the variance of the low-dimensional code using a sequential approach. We start by seeking a single vector $\boldsymbol{b}_1 \in \mathbb{R}^D$ that maximizes the variance of the projected data, i.e., we aim to maximize the variance of the first coordinate z_1 of $\boldsymbol{z} \in \mathbb{R}^M$ so that

$$V_1 := \mathbb{V}[z_1] = \frac{1}{N} \sum^{N} z_{1n}^2 \qquad (10.7)$$

is maximized, where we exploited the i.i.d. assumption of the data and defined z_{1n} as the first coordinate of the low-dimensional representation $\boldsymbol{z}_n \in \mathbb{R}^M$ of $\boldsymbol{x}_n \in \mathbb{R}^D$. Note that first component of $\boldsymbol{z}_n$ is given by

$$z_{1n} = \boldsymbol{b}_1^\top \boldsymbol{x}_n \,, \tag{10.8}$$

i.e., it is the coordinate of the orthogonal projection of $\boldsymbol{x}_n$ onto the one-dimensional subspace spanned by $\boldsymbol{b}_1$ (Section 3.8). We substitute (10.8) into (10.7), which yields

$$V_1 = \frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{b}_1^\top \boldsymbol{x}_n)^2 = \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{b}_1^\top \boldsymbol{x}_n \boldsymbol{x}_n^\top \boldsymbol{b}_1 \tag{10.9a}$$

$$= \boldsymbol{b}_1^\top \left(\frac{1}{N}\sum_{n=1}^{N}\boldsymbol{x}_n \boldsymbol{x}_n^\top\right)\boldsymbol{b}_1 = \boldsymbol{b}_1^\top \boldsymbol{S}\boldsymbol{b}_1 \,, \tag{10.9b}$$

where $\boldsymbol{S}$ is the data covariance matrix defined in (10.1). In (10.9a), we have used the fact that the dot product of two vectors is symmetric with respect to its arguments, that is, $\boldsymbol{b}_1^\top \boldsymbol{x}_n = \boldsymbol{x}_n^\top \boldsymbol{b}_1$.

Notice that arbitrarily increasing the magnitude of the vector $\boldsymbol{b}_1$ increases V_1, that is, a vector $\boldsymbol{b}_1$ that is two times longer can result in V_1 that is potentially four times larger. Therefore, we restrict all solutions to $\|\boldsymbol{b}_1\|^2 = 1$, which results in a constrained optimization problem in which we seek the direction along which the data varies most.

$\|\boldsymbol{b}_1\|^2 = 1$ $\iff \|\boldsymbol{b}_1\| = 1.$

With the restriction of the solution space to unit vectors the vector $\boldsymbol{b}_1$ that points in the direction of maximum variance can be found by the constrained optimization problem

$$\begin{aligned} &\max_{\boldsymbol{b}_1} \boldsymbol{b}_1^\top \boldsymbol{S} \boldsymbol{b}_1 \\ &\text{subject to } \|\boldsymbol{b}_1\|^2 = 1\,. \end{aligned} \tag{10.10}$$

Following Section 7.2, we obtain the Lagrangian

$$\mathfrak{L}(\boldsymbol{b}_1, \lambda) = \boldsymbol{b}_1^\top \boldsymbol{S} \boldsymbol{b}_1 + \lambda_1(1 - \boldsymbol{b}_1^\top \boldsymbol{b}_1) \tag{10.11}$$

to solve this constrained optimization problem. The partial derivatives of $\mathfrak{L}$ with respect to $\boldsymbol{b}_1$ and λ_1 are

$$\frac{\partial \mathfrak{L}}{\partial \boldsymbol{b}_1} = 2\boldsymbol{b}_1^\top \boldsymbol{S} - 2\lambda_1 \boldsymbol{b}_1^\top , \qquad \frac{\partial \mathfrak{L}}{\partial \lambda_1} = 1 - \boldsymbol{b}_1^\top \boldsymbol{b}_1 , \tag{10.12}$$

respectively. Setting these partial derivatives to $\mathbf{0}$ gives us the relations

$$\boldsymbol{S}\boldsymbol{b}_1 = \lambda_1 \boldsymbol{b}_1 , \tag{10.13}$$

$$\boldsymbol{b}_1^\top \boldsymbol{b}_1 = 1 . \tag{10.14}$$

By comparing this with the definition of an eigenvalue decomposition (Section 4.4), we see that $\boldsymbol{b}_1$ is an eigenvector of the data covariance matrix $\boldsymbol{S}$, and the Lagrange multiplier λ_1 plays the role of the corresponding eigenvalue. This eigenvector property (10.13) allows us to rewrite our variance objective (10.10) as ing eigenvalue. This eigenvector property (10.13) allows us to rewrite our variance objective (10.10) as

$$V_1 = \boldsymbol{b}_1^\top \boldsymbol{S} \boldsymbol{b}_1 = \lambda_1 \boldsymbol{b}_1^\top \boldsymbol{b}_1 = \lambda_1 \,, \tag{10.15}$$

i.e., the variance of the data projected onto a one-dimensional subspace equals the eigenvalue that is associated with the basis vector $\boldsymbol{b}_1$ that spans this subspace. Therefore, to maximize the variance of the low-dimensional code, we choose the basis vector associated with the largest eigenvalue of the data covariance matrix. This eigenvector is called the first *principal component*. We can determine the effect/contribution of the principal component $\boldsymbol{b}_1$ in the original data space by mapping the coordinate z_{1n} back into data space, which gives us the projected data point

$$\tilde{\boldsymbol{x}}_n = \boldsymbol{b}_1 z_{1n} = \boldsymbol{b}_1 \boldsymbol{b}_1^\top \boldsymbol{x}_n \in \mathbb{R}^D \tag{10.16}$$

in the original data space.