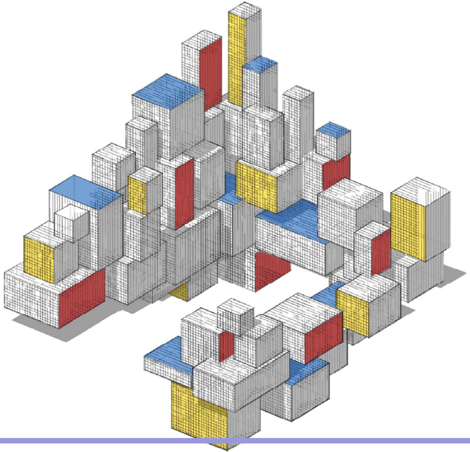
Principal Component Analysis



Purpose

The main idea of principal component analysis (PCA) is to reduce the dimensionality of a data set consisting of many variables. PCA is a *feature reduction* (or *feature extraction*) mechanism, that helps us to handle high-dimensional data with more features than is convenient to interpret.

Motivation: Principal Axes of an Ellipsoid

A d-dimensional normal distribution with mean vector $\mathbf{0}$ and covariance matrix $\mathbf{\Sigma}$ has pdf

$$f(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^n \, |\mathbf{\Sigma}|}} \, \mathrm{e}^{-\frac{1}{2} \, \boldsymbol{x}^\top \mathbf{\Sigma}^{-1} \boldsymbol{x}}, \quad \boldsymbol{x} \in \mathbb{R}^d.$$

The contour surfaces $\boldsymbol{x}^\top \mathbf{\Sigma}^{-1} \boldsymbol{x} = c$, with $c \geqslant 0$, are ellipsoids.

In particular, consider the ellipsoid

$$\boldsymbol{x}^\top \mathbf{\Sigma}^{-1} \boldsymbol{x} = 1, \quad \boldsymbol{x} \in \mathbb{R}^d. \tag{1}$$

Let $\mathbf{\Sigma} = \mathbf{B}\mathbf{B}^\top$. Then, the ellipsoid (1) can be viewed as the linear transformation of d-dimensional unit sphere via matrix $\mathbf{B}$.

Moreover, the principal axes of the ellipsoid can be found via a singular value decomposition (SVD) of $\mathbf{B}$ (or $\mathbf{\Sigma}$).

Singular Value Decomposition

Suppose that an SVD of $\mathbf{B}$ is

$$\mathbf{B} = \mathbf{U}\mathbf{D}\mathbf{V}^\top \qquad \text{(note that an SVD of } \boldsymbol{\Sigma} \text{ is then } \mathbf{U}\mathbf{D}^2\mathbf{U}^\top\text{).}$$

- The columns of the matrix $\mathbf{UD}$ correspond to the principal axes of the ellipsoid.
- The relative magnitudes of the axes are given by the elements of the diagonal matrix $\mathbf{D}$.

If some of these magnitudes are small compared to the others, a reduction in the dimension of the space may be achieved by projecting each point $\boldsymbol{x} \in \mathbb{R}^d$ onto the subspace spanned by the main (say $k \ll d$) columns of $\mathbf{U}$ — the so-called principal components.

Principal Components

Suppose without loss of generality that the first k principal components are given by the first k columns of $\mathbf{U}$, and let $\mathbf{U}_k$ be the corresponding $d \times k$ matrix.

W.r.t. the standard basis $\{\boldsymbol{e}_i\}$, the vector $\boldsymbol{x} = x_1\boldsymbol{e}_1 + \cdots + x_d\boldsymbol{e}_d$ is represented by the d-dimensional vector $[x_1, \ldots, x_d]^\top$.

W.r.t. the orthonormal basis $\{\boldsymbol{u}_i\}$ formed by the columns of matrix $\mathbf{U}$, the representation of $\boldsymbol{x}$ is $\mathbf{U}^\top\boldsymbol{x}$.

Similarly, the projection of any point $\boldsymbol{x}$ onto the subspace spanned by the first k principal vectors is represented by the k-dimensional vector $\mathbf{U}_k^\top\boldsymbol{x}$, w.r.t. the orthonormal basis formed by the columns of $\mathbf{U}_k$.

So, the idea is that if a point $\boldsymbol{x}$ lies close to its projection $\mathbf{U}_k\mathbf{U}_k^\top\boldsymbol{x}$, we may represent it via k numbers instead of d, using the combined features given by the k principal components.

Example: Principal Components

Consider the matrices

$$\boldsymbol{\Sigma} = \begin{bmatrix} 14 & 8 & 3 \\ 8 & 5 & 2 \\ 3 & 2 & 1 \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix},$$

where $\boldsymbol{\Sigma} = \mathbf{B}\mathbf{B}^\top$. The ellipsoid $\boldsymbol{x}^\top \boldsymbol{\Sigma} \boldsymbol{x} = 1$ is depicted below.

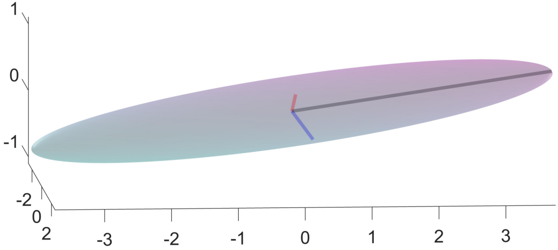


Figure: A "surfboard" ellipsoid $\boldsymbol{x}^\top \boldsymbol{\Sigma} \boldsymbol{x} = 1$.

Projections

The principal axes and sizes of the ellipsoid are found through a singular value decomposition $\mathbf{B} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$, where $\mathbf{U}$ and $\mathbf{D}$ are

$$\mathbf{U} = \begin{bmatrix} 0.8460 & 0.4828 & 0.2261 \\ 0.4973 & -0.5618 & -0.6611 \\ 0.1922 & -0.6718 & 0.7154 \end{bmatrix} \quad \text{and} \quad \mathbf{D} = \begin{bmatrix} 4.4027 & 0 & 0 \\ 0 & 0.7187 & 0 \\ 0 & 0 & 0.3160 \end{bmatrix}.$$

The columns of $\mathbf{U}$ show the directions of the principal axes of the ellipsoid, and the diagonal elements of $\mathbf{D}$ indicate the relative magnitudes of the principal axes.

The first principal component, $\boldsymbol{u}_1$, is given by the first column of $\mathbf{U}$, and the second principal component by the second column of $\mathbf{U}$.

The projection, $\boldsymbol{z}$, of $\boldsymbol{x} = [1.052, 0.6648, 0.2271]^\top$ onto the 1-d space spanned by $\boldsymbol{u}_1$ is $\boldsymbol{z} = \boldsymbol{u}_1\boldsymbol{u}_1^\top\boldsymbol{x} = [1.0696, 0.6287, 0.2429]^\top$.

With respect to the basis vector $\boldsymbol{u}_1$, $\boldsymbol{z}$ is represented by the number $\boldsymbol{u}_1^\top\boldsymbol{z} = 1.2643$. That is, $\boldsymbol{z} = 1.2643\boldsymbol{u}_1$.

PCA

In principal component analysis (PCA) we start with data $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$, where each $\boldsymbol{x}$ is d-dimensional.

Think of the data as iid draws from a multivariate normal pdf.

Let us collect the data in a matrix $\mathbf{X}$ in the usual way; that is,

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \ldots & x_{1d} \\ x_{21} & x_{22} & \ldots & x_{2d} \\ \vdots & \vdots & \vdots & \vdots \\ x_{n1} & x_{n2} & \ldots & x_{nd} \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}_1^\top \\ \boldsymbol{x}_2^\top \\ \vdots \\ \boldsymbol{x}_n^\top \end{bmatrix}.$$

The matrix $\mathbf{X}$ will be the PCA's input. Under this setting, the data consists of points in d-dimensional space, and our goal is to present the data using n feature vectors of dimension $k < d$.

Centered Data

In accordance with the previous motivation, we assume that underlying distribution of the data has expectation vector $\mathbf{0}$. In practice, this means that before PCA is applied, the data needs to be *centered* by subtracting the *column* mean in every column:

$$x'_{ij} = x_{ij} - \overline{x}_j, \quad \text{where} \quad \overline{x}_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij}.$$

We assume from now on that the data comes from a general d-dimensional distribution with mean vector $\mathbf{0}$ and some covariance matrix $\mathbf{\Sigma}$.

The covariance matrix $\mathbf{\Sigma}$ is by definition equal to the expectation of the random matrix $\boldsymbol{X}\boldsymbol{X}^\top$, and can be estimated from the data $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ via the sample average

$$\widehat{\mathbf{\Sigma}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_i^\top = \frac{1}{n}\mathbf{X}^\top\mathbf{X}.$$

SVD

As $\widehat{\boldsymbol{\Sigma}}$ is a covariance matrix, we can consider an SVD decomposition.

Suppose $\widehat{\boldsymbol{\Sigma}} = \mathbf{U}\mathbf{D}^2\mathbf{U}^\top$ is an SVD of $\widehat{\boldsymbol{\Sigma}}$ and let $\mathbf{U}_k$ be the matrix whose columns are the k principal components.

The transformation $\boldsymbol{z}_i = \mathbf{U}_k\mathbf{U}_k^\top\boldsymbol{x}_i$ maps each $\boldsymbol{x}_i \in \mathbb{R}^d$ to a $\boldsymbol{z}_i \in \mathbb{R}^d$ lying in the k-dimensional subspace spanned by the columns of $\mathbf{U}_k$.

With respect to this basis, the point $\boldsymbol{z}_i$ has representation $\boldsymbol{z}_i = \mathbf{U}_k^\top\boldsymbol{x}_i \in \mathbb{R}^k$ (thus with k features).

The covariance matrix of the $\boldsymbol{z}_i, i = 1, \ldots, n$ is diagonal.

The quantity $v = \sum_{\ell=1} d_{\ell\ell}^2$ is a measure for the amount of variance in the data.

The proportion $d_{\ell\ell}^2/v$ indicates how much of the variance in the data is explained by the ℓ-th principal component.

Another View Point of PCA

How can we best project the data onto a k-dimensional subspace in such a way that the total squared distance between the projected points and the original points is minimal?

Recall that any orthogonal projection to a k-dimensional subspace $\mathcal{V}_k$ can be represented by a matrix $\mathbf{U}_k\mathbf{U}_k^\top$, where $\mathbf{U}_k = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k]$ and the $\{\boldsymbol{u}_\ell, \ell = 1, \ldots, k\}$ are orthogonal vectors of length 1 that span $\mathcal{V}_k$.

The above question can thus be formulated as the minimization program:

Another View Point of PCA

Now observe that

$$\frac{1}{n}\sum_{i=1}^{n}\|\boldsymbol{x}_i - \mathbf{U}_k\mathbf{U}_k^\top\boldsymbol{x}_i\|^2 = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{x}_i^\top - \boldsymbol{x}_i^\top\mathbf{U}_k\mathbf{U}_k^\top)(\boldsymbol{x}_i - \mathbf{U}_k\mathbf{U}_k^\top\boldsymbol{x}_i)$$

$$= \underbrace{\frac{1}{n}\sum_{i=1}^{n}\|\boldsymbol{x}_i\|^2}_{c} - \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i^\top\mathbf{U}_k\mathbf{U}_k^\top\boldsymbol{x}_i = c - \frac{1}{n}\sum_{i=1}^{n}\sum_{\ell=1}^{k}\operatorname{tr}(\boldsymbol{x}_i^\top\boldsymbol{u}_\ell\boldsymbol{u}_\ell^\top\boldsymbol{x}_i)$$

$$= c - \frac{1}{n}\sum_{\ell=1}^{k}\sum_{i=1}^{n}\boldsymbol{u}_\ell^\top\boldsymbol{x}_i\boldsymbol{x}_i^\top\boldsymbol{u}_\ell = c - \sum_{\ell=1}^{k}u_\ell^\top\,\widehat{\boldsymbol{\Sigma}}\,\boldsymbol{u}_\ell,$$

where we have used the cyclic property of a trace and the fact that $\mathbf{U}_k\mathbf{U}_k^\top$ can be written as $\sum_{\ell=1}^{k}\boldsymbol{u}_\ell\boldsymbol{u}_\ell^\top$.

Another View Point of PCA

It follows that the minimization problem

$$\min_{\boldsymbol{u}_1,\ldots,\boldsymbol{u}_k} \sum_{i=1}^{n} \|\boldsymbol{x}_i - \mathbf{U}_k \mathbf{U}_k^\top \boldsymbol{x}_i\|^2.$$

is equivalent to the maximization problem

$$\max_{\boldsymbol{u}_1,\ldots,\boldsymbol{u}_k} \sum_{\ell=1}^{k} \boldsymbol{u}_\ell^\top \widehat{\boldsymbol{\Sigma}} \boldsymbol{u}_\ell.$$

This maximum can be at most $\sum_{\ell=1}^{k} d_{\ell\ell}^2$ and is attained precisely when $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k$ are the first k principal components of $\widehat{\boldsymbol{\Sigma}}$.

Example: Singular Value Decomposition

The following data set consists of independent samples from the three-dimensional Gaussian distribution with mean vector **0** and the "surfboard" covariance matrix **Σ**:

$$\mathbf{X} = \begin{bmatrix} 3.1209 & 1.7438 & 0.5479 \\ -2.6628 & -1.5310 & -0.2763 \\ 3.7284 & 3.0648 & 1.8451 \\ 0.4203 & 0.3553 & 0.4268 \\ -0.7155 & -0.6871 & -0.1414 \\ 5.8728 & 4.0180 & 1.4541 \\ 4.8163 & 2.4799 & 0.5637 \\ 2.6948 & 1.2384 & 0.1533 \\ -1.1376 & -0.4677 & -0.2219 \\ -1.2452 & -0.9942 & -0.4449 \end{bmatrix}.$$

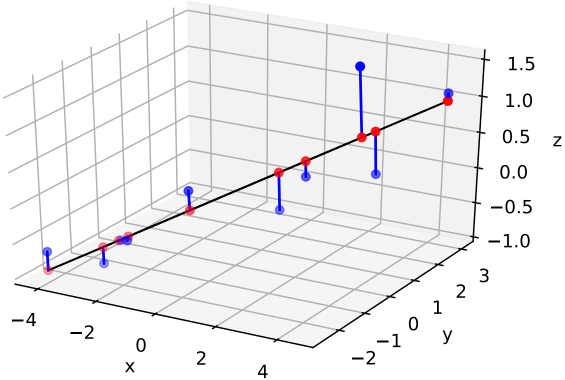
After replacing $\mathbf{X}$ with its centered version, an SVD $\mathbf{U}\mathbf{D}^2\mathbf{U}^\top$ of $\widehat{\mathbf{\Sigma}} = \mathbf{X}^\top\mathbf{X}/n$ yields the principal component matrix $\mathbf{U}$ and diagonal matrix $\mathbf{D}$:

$$\mathbf{U} = \begin{bmatrix} -0.8277 & 0.4613 & 0.3195 \\ -0.5300 & -0.4556 & -0.7152 \\ -0.1843 & -0.7613 & 0.6216 \end{bmatrix} \quad \text{and} \quad \mathbf{D} = \begin{bmatrix} 3.3424 & 0 & 0 \\ 0 & 0.4778 & 0 \\ 0 & 0 & 0.1038 \end{bmatrix}.$$

Note that these are similar to those in the surfboard example.

We see that 97.90% of the total variance is explained by the first principal component.

The figure shows the projection of the centered data onto the subspace spanned by this principal component.



The following Python code was used.

PCAdat.py

```
import numpy as np
X = np.genfromtxt('pcadat.csv', delimiter=',')
n = X.shape[0]

X = X - X.mean(axis=0)
G = X.T @ X
U, _ , _ = np.linalg.svd(G/n)

# projected points
Y = X @ np.outer(U[:,0],U[:,0])

import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.w_xaxis.set_pane_color((0, 0, 0, 0))
ax.plot(Y[:,0], Y[:,1], Y[:,2], c='k', linewidth=1)
ax.scatter(X[:,0], X[:,1], X[:,2], c='b')
ax.scatter(Y[:,0], Y[:,1], Y[:,2], c='r')

for i in range(n):
    ax.plot([X[i,0], Y[i,0]], [X[i,1],Y[i,1]], [X[i,2],Y[i,2]], 'b')

ax.set_xlabel('x')
ax.set_ylabel('y')
ax.set_zlabel('z')
plt.show()
```

PCA for the Iris Data Set

The **`iris`** data set contains measurements on four features of the iris plant: sepal length and width, and petal length and width, for a total of 150 specimens.

There is significant correlation between the different features. Can we perhaps describe the data using fewer features by taking certain linear combinations of the original features?

To investigate this, let us perform a PCA, first centering the data. The following Python code implements the PCA. It is assumed that a CSV file `irisX.csv` has been made that contains the iris data set (without the species information).

PCAiris.py

```python
import seaborn as sns, numpy as np
np.set_printoptions(precision=4)

X = np.genfromtxt('IrisX.csv',delimiter=',')
n = X.shape[0]
X = X - np.mean(X, axis=0)

[U,D2,UT]= np.linalg.svd((X.T @ X)/n)
print('U = \n', U); print('\n diag(D^2) = ', D2)

z =  U[:,0].T @ X.T

sns.kdeplot(z, bw=0.15)
```

```
U =
 [[-0.3614 -0.6566  0.582   0.3155]
 [ 0.0845 -0.7302 -0.5979 -0.3197]
 [-0.8567  0.1734 -0.0762 -0.4798]
 [-0.3583  0.0755 -0.5458  0.7537]]

 diag(D^2) =  [4.2001 0.2411 0.0777 0.0237]
```

Iris Data

The output above shows the principal component matrix (which we called **U**) as well as the diagonal of matrix $\mathbf{D}^2$.

A large proportion of the variance, $4.2001/(4.2001 + 0.2411 + 0.0777 + 0.0237) = 92.46\%$, is explained by the first principal component. Thus, it makes sense to transform each data point $\boldsymbol{x} \in \mathbb{R}^4$ to $\boldsymbol{u}_1^\top \boldsymbol{x} \in \mathbb{R}$.

In a KDE of the transformed data we see two modes, indicating at least two clusters in the data.

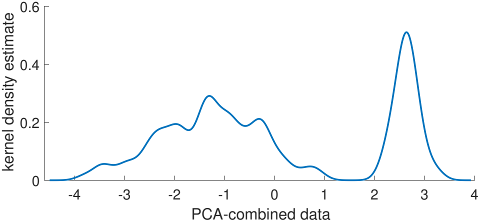


Figure: Kernel density estimate of the PCA-combined `iris` data.