PRINCIPAL COMPONENT ANALYSIS

Unsupervised Learning

Lê Hồng Phương
Data Science Laboratory
Vietnam National University, Hanoi
<*phuonglh@hus.edu.vn*>

December 13, 2021

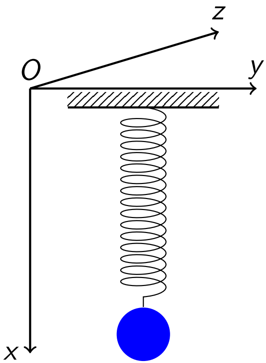
Content

Overview

- Principal Component Analysis (PCA) is a simple, non-parametric method of *extracting relevant information* from noisy datasets.
- PCA provides a method to reduce a complex dataset to a lower dimension to reveal hidden properties/structures of the dataset.
- PCA is widely used in many forms of analysis: neuroscience, computer graphics, natural language processing, *etc.*

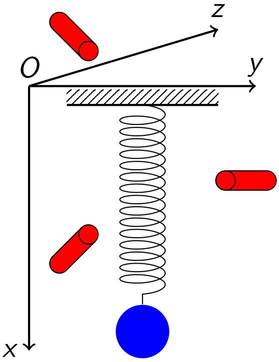
Motivation: A Toy Example

- We are studying the motion of an ideal spring.
- This system consists of a ball of mass m attached to a massless, frictionless spring.
- The ball is released a small distance away from equilibrium (the spring is stretched).
- The spring oscillates indefinitely along the x-axis about its equilibrium at some frequency.



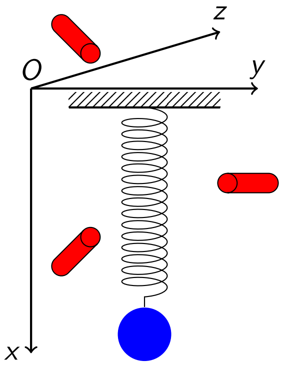
Motivation: A Toy Example

- This is a standard problem in physics, the motion along the x-axis is solved by an explicit function of time.
  - The underlying dynamics can be expressed as a function of a single variable x.
- However, suppose that we do not know which axes and dimensions are important to measure.
- Thus, we decide to measure the ball's position in a three-dimensional space.
  - We place 3 cameras around our system of interest.



Motivation: A Toy Example

- At 200 Hz, each camera records an image indicating a 2-dimensional position of the ball (a projection).
- Unfortunately, we do not even know what are the real "x", "y" and "z", so we choose 3 camera axes $\{\vec{a}, \vec{b}, \vec{c}\}$ at some arbitrary angles w.r.t. the system.
- The angles between our measurements might not even be 90^0!
- Now, we record the cameras for 2 minutes.
- How do we get from this dataset to a simple equation of x?



Motivation: A Toy Example

Some common problems:

- We sometimes record more dimensions than we actually need.
- We have to deal with noise (*e.g.* air, imperfect cameras, friction. . .)

Motivation: A Toy Example

Some common problems:

- We sometimes record more dimensions than we actually need.
- We have to deal with noise (*e.g.* air, imperfect cameras, friction...)

Goal: PCA computes the most meaningful *basis* to re-express a noisy, garbled dataset.

- The new basis will filter out the noise and reveal hidden dynamics (*e.g.* the dynamics are along the x-axis).

Content

A Naive Basis

A naive and simple choice of a basis is the identity matrix:

$$\mathbf{I} = \begin{pmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \\ \vdots \\ \mathbf{e}_D \end{pmatrix} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}$$

- Each row is a basis vector $\mathbf{e}_i$ with D components.
- Every data point is a vector that lies in a D-dimensional vector space spanned by an orthonormal basis.
- *All vectors in this space are a linear combination of this set of unit length basis vectors.*

Change of Basis

PCA question: *Is there another basis, which is a linear combination of the original basis, that best re-expresses our dataset?*

Change of Basis

PCA question: *Is there another basis, which is a linear combination of the original basis, that best re-expresses our dataset?*

Note: PCA makes a powerful assumption: *linearity*.

- The data characterizes/provides an ability to interpolate between the individual data points.

Change of Basis

Let $\mathbf{X}$ and $\mathbf{Z}$ be $N \times D$ matrices related by a linear transformation θ:

$$\boxed{\mathbf{X}\,\theta = \mathbf{Z}}$$

- $\mathbf{X}$ is the original recorded dataset;
- $\mathbf{Z}$ is a re-representation of that dataset.

Change of Basis

$$\boxed{\mathbf{X}\,\theta = \mathbf{Z}}$$

This change of basis has some interpretations:

- θ is a matrix that transforms $\mathbf{X}$ to $\mathbf{Z}$.
- Geometrically, θ is a rotation and a stretch which transforms $\mathbf{X}$ into $\mathbf{Z}$.
- The columns of θ are a set of new basis vectors for expressing the rows of $\mathbf{X}$:

$$\begin{pmatrix} - & \mathbf{x}_1 & - \\ - & \mathbf{x}_2 & - \\ \dots & \dots & \dots \\ - & \mathbf{x}_N & - \end{pmatrix} \begin{pmatrix} | & | & \vdots & | \\ \theta_1 & \theta_2 & \vdots & \theta_D \\ | & | & \vdots & | \end{pmatrix} = \mathbf{Z}$$

Change of Basis

- Each row of $\mathbf{Z}$ is

$$\mathbf{z}_i = (\mathbf{x}_i \cdot \theta_1, \mathbf{x}_i \cdot \theta_2, \ldots, \mathbf{x}_i \cdot \theta_D).$$

- Each element of $\mathbf{z}_i$ is a dot product of $\mathbf{x}_i$ with the corresponding column in θ.
- That is, the j-th element of $\mathbf{z}_i$ is a projection of $\mathbf{x}_i$ onto the j-th column of θ.

Change of Basis

- By assuming linearity, the problem reduces to finding the appropriate change of basis.
- The column vectors $\theta_1, \theta_2, \ldots, \theta_D$ will become the **principal components** of $\mathbf{X}$.

Change of Basis

- By assuming linearity, the problem reduces to finding the appropriate change of basis.
- The column vectors $\theta_1, \theta_2, \ldots, \theta_D$ will become the **principal components** of $\mathbf{X}$.

Questions:

- What is the best way to re-express $\mathbf{X}$?
- What is a good choice of basis θ?

Content

Noise

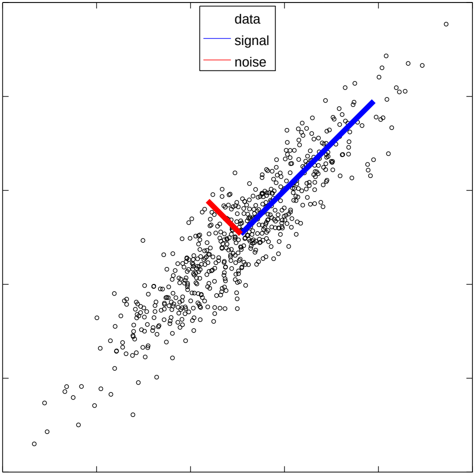
- Noise in any dataset must be low, otherwise, no useful information of a system can be extracted.
- All noise is measure relative to the measurement. A common measure is the *signal-to-noise ratio* (SNR), or a ratio of variance σ^2:

$$\mathrm{SNR} = \frac{\sigma^2_{\mathrm{signal}}}{\sigma^2_{\mathrm{noise}}}$$

- A high SNR ($\mathrm{SNR} \gg 1$) indicates high precision data, while a low SNR indicates noise contaminated data.

Noise

The SNR measures how “fat” the oval is.



Noise

- In our example, any individual camera should record motion in a straitght line.
- Therefore, any spread deviating from straight-line motion must be noise.

Redundancy

- Redundancy is more tricky issue. Multiple sensors record the same dynamics information.
- A simple way to quantify the redundancy between measurements is to calculate the their **covariance**.
- Covariance is a measure of how much two random variables change together:
  - If the variables tend to show similar behavior (greater/greater, smaller/smaller), the covariance is positive.
  - If the variables tend to show opposite behavior (greater/smaller, smaller/greater), the covariance is negative.
- The sign of the covariance therefore shows the tendency in the *linear relationship* between the variables.

Redundancy

The covariance between two jointly distributed real-valued random variables X and Y is defined as:

$$\begin{aligned}\sigma_{XY}^2 &= \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])] \\ &= \mathbb{E}[XY] - \mathbb{E}[X]\,\mathbb{E}[Y].\end{aligned}$$

Two important facts about covariance:

- $\sigma_{XY}^2 = 0$ iff X and Y are entirely uncorrelated.
- $\sigma_{XY}^2 = \sigma_X^2$ iff $X = Y$.

Redundancy

- For random vectors $\mathbf{X}$ and $\mathbf{Y}$, both of dimension D, their $D \times D$ *covariance matrix* is

$$\sigma^2_{\mathbf{X}\,\mathbf{Y}} = \mathbb{E}[\mathbf{X}\,\mathbf{Y}^T] - \mathbb{E}[\mathbf{X}]\,\mathbb{E}[\mathbf{Y}^T].$$

- For a vector X of D jointly distributed real-valued random variables, its covariance matrix is

$$\Sigma(\mathbf{X}) = \sigma^2_{\mathbf{X}\,\mathbf{X}}.$$

Redundancy

The Iris dataset:

$$\Sigma(\mathbf{X}) = \begin{pmatrix} 0.665822 & -0.026056 & 1.235005 & 0.500998 \\ -0.026056 & 0.190509 & -0.308566 & -0.111119 \\ 1.235005 & -0.308566 & 3.071335 & 1.279612 \\ 0.500998 & -0.111119 & 1.279612 & 0.576284 \end{pmatrix}$$

Note that the diagonal elements of $\Sigma(\mathbf{X})$ are the variances of particular features.

Redundancy

If $\mathbf{X}$ is a dataset containing N examples, each example $\mathbf{x}_i$ has D features with *zero mean*. Then:

$$\Sigma(\mathbf{X}) = \frac{1}{N-1}\mathbf{X}^T\mathbf{X}.$$

$\Sigma(\mathbf{X})$ is a square symmetric $D \times D$ matrix.

Diagonalize the Covariance Matrix

- Our goal is to reduce redundancy, then we want each feature to co-vary as little as possible with other features.
- In order to remove redundancy, we want that all the covariances between separate features to be zero.
- That is, we want to transform from $\mathbf{X}$ to $\mathbf{Z}$ such that $\Sigma(\mathbf{Z})$ is a diagonal matrix.

Diagonalize the Covariance Matrix

- There are many methods for diagonalizing $\Sigma(\mathbf{Z})$. PCA uses the easiest method.
- First, PCA assumes that all basis vectors are orthonormal, that is

$$\theta_i \cdot \theta_j \equiv \delta(i = j).$$

  In other words, θ is an orthonormal matrix.
- Second, PCA assumes the directions with the largest variances are the most "*important*", or most "*principal*".

Diagonalize the Covariance Matrix

How PCA works:

- First, it selects a normalized direction in D-dimensional space along which the variance in $\mathbf{Z}$ is maximized. It saves this as θ_1.
- Then, it find another direction θ_2 along which the variance is maximized. Because of the orthonormality condition, it restricts the search to all directions perpindicular to all previous selected directions $(\theta_2 \cdot \theta_1 = 0)$.
- This continues until D directions are selected.
- The resulting ordered set of θ_j are the **principal components**.

PCA Problem

Problem

Find some orthonormal matrix θ where $\mathbf{Z} = \mathbf{X}\,\theta$ such that $\Sigma(\mathbf{Z}) = \frac{1}{N-1}\,\mathbf{Z}^T\,\mathbf{Z}$ is diagonalized.

The columns of θ are the principal components of $\mathbf{X}$.

Content

Solving PCA: Eigenvectors of Covariance

We have

$$
\begin{aligned}
\Sigma(\mathbf{Z}) &= \frac{1}{N-1}\mathbf{Z}^T\mathbf{Z} \\
&= \frac{1}{N-1}(\mathbf{X}\,\theta)^T(\mathbf{X}\,\theta) \\
&= \frac{1}{N-1}\theta^T\mathbf{X}^T\mathbf{X}\,\theta \\
&= \frac{1}{N-1}\theta^T\mathbf{A}\theta,
\end{aligned}
$$

where we define $\mathbf{A} = \mathbf{X}^T\mathbf{X}$, which is a *symmetric* matrix.

Solving PCA: Eigenvectors of Covariance

Theorem

A symmetric matrix is diagonalized by a matrix of its orthonormal eigenvectors.

Because of this theorem, there exists a diagonal matrix $\mathbf{D}$ such that

$$\mathbf{A} = \mathbf{E}\mathbf{D}\mathbf{E}^T,$$

where $\mathbf{E}$ is a matrix of eigenvectors of $\mathbf{A}$ arranged as columns.

Solving PCA: Eigenvectors of Covariance

- The matrix $\mathbf{A}$ has $L \leq D$ orthonormal eigenvectors where L is the rank of the matrix.
- The rank of $\mathbf{A}$ is less than D when $\mathbf{A}$ degenerate, or all data occupy a subspace of dimension $L < D$.
- So, *we select the matrix θ to be a matrix where each column θ_j is an eigenvector of $\mathbf{X}^T\mathbf{X}$.*
- By this selection, we have $\theta = \mathbf{E}$. So,

$$\mathbf{A} = \theta \mathbf{D} \theta^T.$$

Solving PCA: Eigenvectors of Covariance

Therefore,

$$
\begin{aligned}
\Sigma(\mathbf{Z}) &= \frac{1}{N-1}\theta^T \mathbf{A}\theta \\
&= \frac{1}{N-1}\theta^T(\theta \mathbf{D}\theta^T)\theta \\
&= \frac{1}{N-1}(\theta^T\theta)\mathbf{D}(\theta^T\theta) \\
&= \frac{1}{N-1}(\theta^{-1}\theta)\mathbf{D}(\theta^{-1}\theta) \\
&= \frac{1}{N-1}\mathbf{D}.
\end{aligned}
$$

That is, the choice of θ diagonalizes $\Sigma(\mathbf{Z})$.

Content

Theoretical Basis

Theorem

The inverse of an orthogonal matrix is its transpose.

Theorem

If $\mathbf{X}$ is any matrix, the matrices $\mathbf{X}^T\mathbf{X}$ and $\mathbf{X}\mathbf{X}^T$ are both symmetric.

Theorem

A matrix is symmetric if and only if it is orthogonally diagonalizable.

Theoretical Basis

Theorem

A symmetric matrix is diagonalized by a matrix of its orthonormal eigenvectors.

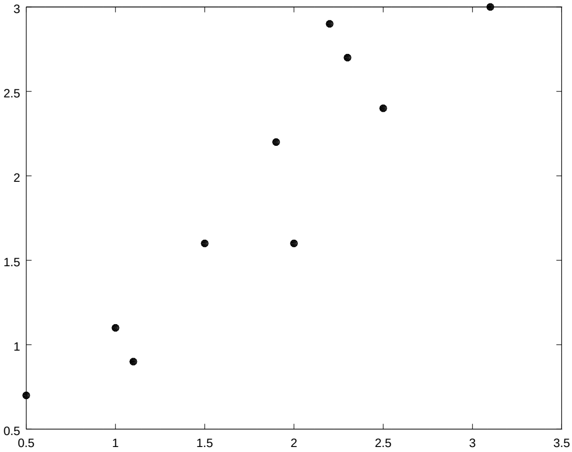
Theorem

For any arbitrary $N \times D$ *matrix* $\mathbf{X}$*, the symmetric matrix* $\mathbf{X}^T\mathbf{X}$ *has a set of orthonormal eigenvectors* $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_D\}$ *and a set of associated eigenvalues* $\{\lambda_1, \lambda_2, \dots, \lambda_D\}$*. The set of vectors* $\{\mathbf{X}\mathbf{v}_1, \mathbf{X}\mathbf{v}_2, \dots, \mathbf{X}\mathbf{v}_D\}$ *form an orthogonal basis, where each vector* $\mathbf{X}\mathbf{v}_j$ *is of length* $\sqrt{\lambda_j}$*.*

Content

Example 1: Toy Dataset

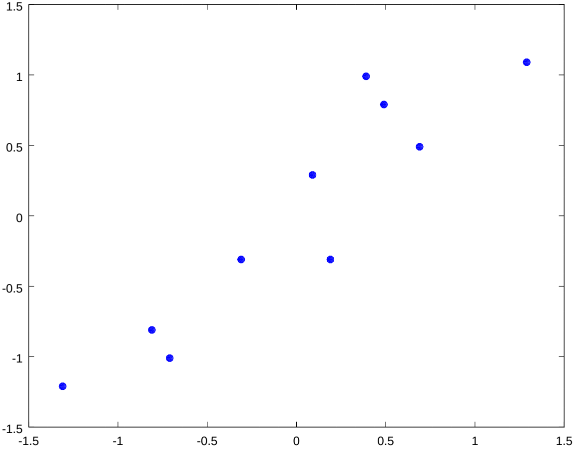
x_1	x_2
2.5	2.4
0.5	0.7
2.2	2.9
1.9	2.2
3.1	3.0
2.3	2.7
2.0	1.6
1.0	1.1
1.5	1.6
1.1	0.9



Example 1: Toy Dataset

x_1	x_2
0.69	0.49
-1.31	-1.21
0.39	0.99
0.09	0.29
1.29	1.09
0.49	0.79
0.19	-0.31
-0.81	-0.81
-0.31	-0.31
-0.71	-1.01

$$\mu = (1.81, 1.91)$$



Example 1: Toy Dataset

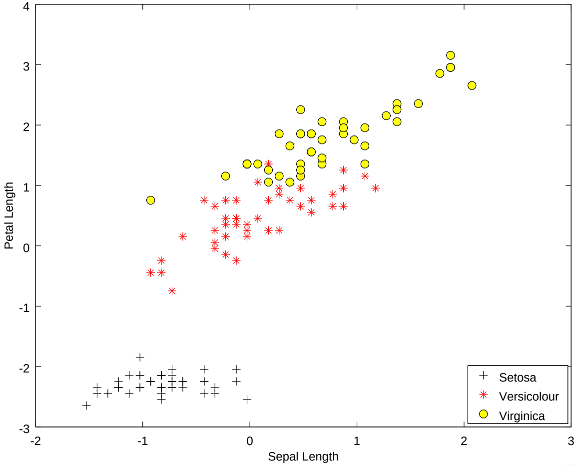
$$\theta = \begin{pmatrix} 0.67787 & -0.73518 \\ 0.73518 & 0.67787 \end{pmatrix}$$

$$\mathbf{Z} = \begin{pmatrix} 0.827970 & -0.175115 \\ -1.777580 & 0.142857 \\ 0.992197 & 0.384375 \\ 0.274210 & 0.130417 \\ 1.675801 & -0.209498 \\ 0.912949 & 0.175282 \\ -0.099109 & -0.349825 \\ -1.144572 & 0.046417 \\ -0.438046 & 0.017765 \\ -1.223821 & -0.162675 \end{pmatrix}$$

Content

Iris Dataset

Four features, reorder features as "`Sepal Length`", "`Petal Length`", "`Sepal Width`", "`Petal Width`".



Iris Dataset

$$\theta = \begin{pmatrix} 0.356687 & 0.657221 & 0.578737 & 0.325419 \\ 0.858455 & -0.176179 & -0.060299 & -0.477891 \\ -0.079358 & 0.729440 & -0.589941 & -0.337032 \\ 0.359904 & -0.070280 & -0.559819 & 0.743056 \end{pmatrix}$$

First and second principal component:

$$\theta_1 = \begin{pmatrix} 0.356687 \\ 0.858455 \\ -0.079358 \\ 0.359904 \end{pmatrix}; \quad \theta_2 = \begin{pmatrix} 0.657221 \\ -0.176179 \\ 0.729440 \\ -0.070280 \end{pmatrix}$$

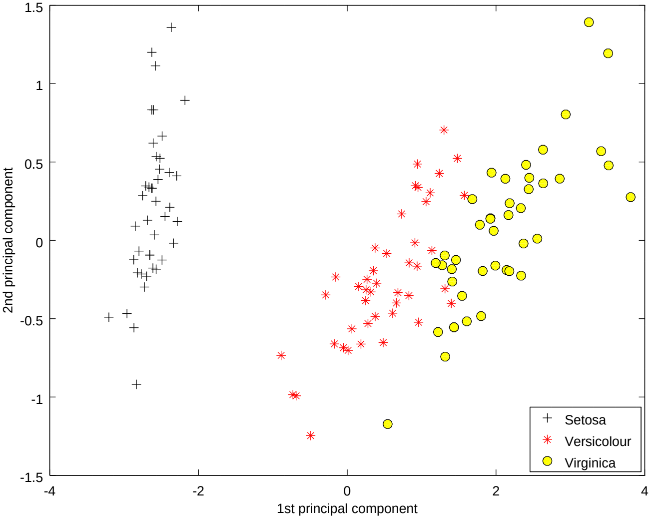
Iris Dataset

Projection of **X** into 2 two-dimensional space:

$$\mathbf{Z} = \mathbf{X} * [\theta_1, \theta_2]$$

This can be viewed as a "data compression" technique (dimensionality reduction).

Iris Dataset



Iris Dataset

- In practice, if we were using a learning algorithm (linear regression, neural networks,...), we could now use the projected data instead of the original data.
- By using the projected data, we can train our model faster as there are less dimensions in the input.

Data Reconstruction

- After projecting the data onto the lower dimensional space, we can approximately recover the data by projecting them back to the original high dimensional space:

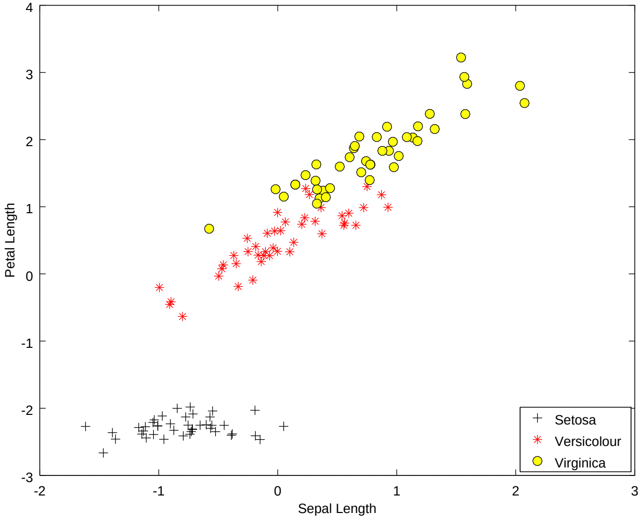
$$\mathbf{X}' = \mathbf{Z}\,\theta^T,$$

where $\theta = [\theta_1, \theta_2, \dots, \theta_K]$ contains K principal components.

- The recovered data $\mathbf{X}'$ is generally a coarsed-grained version of the original data $\mathbf{X}$:
  - Some information is lost, some hidden semantics/structures are retained.
- Reconstruction error:

$$J(\theta) = \frac{1}{N}\sum_{i=1}^{N} \| \mathbf{x}_i - \mathbf{x}_i' \|^2.$$

Data Reconstruction – Iris Dataset



Content

Face Image Dataset

- We run PCA on face images to see how it can be used in practice for dimension reduction.
- The face image dataset contains 5000 face images, each of size 32×32 in grayscale.[1]
- Each row of $\mathbf{X}$ corresponds to one face image (a row vector of length 1024).

[1] A subset of the Labeled Face in the Wild Home.

Face Image Dataset – 100 Original Faces

Face Image Dataset – 36 Principal Components

Face Image Dataset – 100 Principal Components

Original faces

Recovered faces

Exercises

1. Implement the PCA algorithm.
2. Test the algorithm on different datasets.
3. Run a classification algorithm on the projected Iris dataset (using first two principal components) and report the classification accuracy.