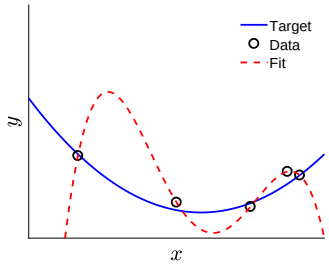
Review of lecture 3

- **Overfitting**
  Fitting the data more than it is warranted



Stochastic noise + **Deterministic noise**

VC analysis allows it; does not predict it

- **Regularization**
  Looking for smoother hypothesis

$$E_{\text{aug}}(h) = E_{\text{in}}(h) + \lambda\Omega(h)$$

  Ridge and Lasso penalties

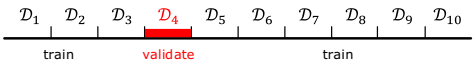
- **Validation**

$$E_{\text{val}}(g^-) \text{ estimates } E_{\text{out}}(g^-)$$

  $\mathcal{D}_{\text{val}}$ is slightly contaminated

  Needs for a test set

- **Cross-validation**



10-fold cross-validation

Machine Learning: Lecture 4

K-means
Principal Component Analysis

Mirko Mazzoleni

25 May 2017

University of Bergamo
Department of Management, Information and Production Engineering
mirko.mazzoleni@unibg.it

Outline

- Unsupervised learning
- K-means clustering
- Principal Components Analysis

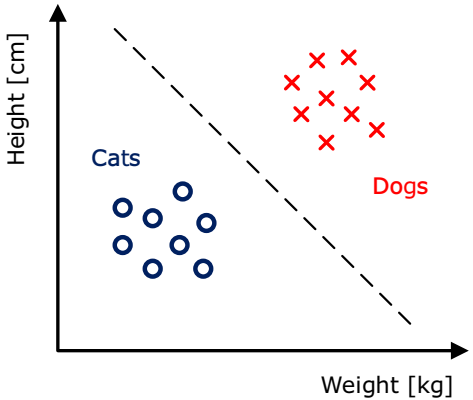
Outline

- **Unsupervised learning**
- K-means clustering
- Principal Components Analysis

Supervised learning

In *supervised* learning, we have a dataset $\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_N, y_N)\}$

- The outcome variable y is given
- The aim is to provide a prediction or a classification



Supervised learning

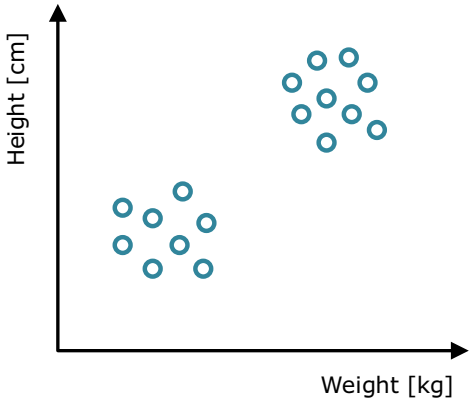
Regression case: the output y is given. The output is **mandatory** in order to compute the cost function $\rightarrow$ which, minimized, gives the estimated model's parameters

Size $[\text{feet}^2]$	Number of bedrooms	Number of floors	Age of home [year]	Price [\$]
2104	5	1	45	$4.60 \cdot 10^5$
1416	3	2	40	$2.32 \cdot 10^5$
1534	2	1	30	$3.15 \cdot 10^5$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$
x_1	x_2	x_3	x_4	y

Unsupervised learning

In *unsupervised* learning, we have a dataset $\mathcal{D} = \{(\mathbf{x}_1, ?), (\mathbf{x}_2, ?), \ldots, (\mathbf{x}_N, ?)\}$

- The outcome variable y is **not** given
- The aim to discover interesting things about the measurements



Unsupervised learning

Unsupervised learning case: the output y is **not** given. Only intrinsic information that are present in the inputs $\mathbf{x}$ can be used $\rightarrow$ each row is an input vector $\mathbf{x}$

Size $[\text{feet}^2]$	Number of bedrooms	Number of floors	Age of home $[\text{year}]$	Price [$]
2104	5	1	45	–
1416	3	2	40	–
1534	2	1	30	–
⋮	⋮	⋮	⋮	⋮
↓	↓	↓	↓	
x_1	x_2	x_3	x_4	

Goals of unsupervised learning

The main questions that unsupervised learning can answer are:

1. Can we discover subgroups among the variables or among the observations?
   - Subgroups of breast cancer patients grouped by their gene expression measurements
   - Groups of shoppers characterized by their browsing and purchase histories
   - Movies grouped by the ratings assigned by movie viewers
   - Search results showed to a particular individual based on the click histories of other individuals with similar search patterns
2. Is there an informative way to visualize the data?
   - Find transformations which enhances certain data characteristics
   - Reduce the dimensionality of the data to visualize them in a 2-D plot

Other advantages

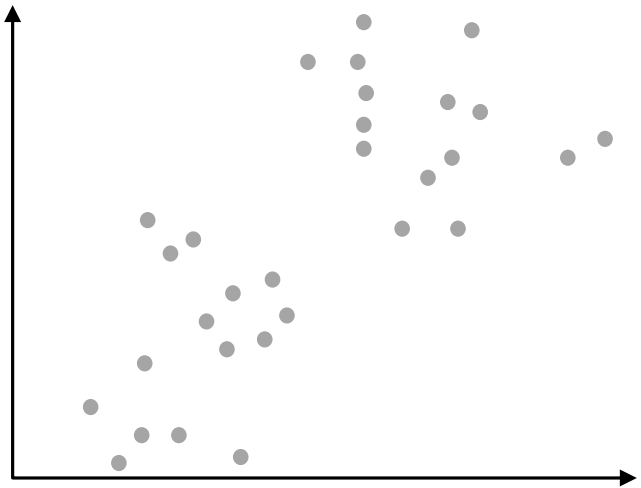
Unsupervised learning presents relative advantages with respect to supervised learning

- It is often easier to obtain unlabeled data than labeled data
- For example there is the need to perform a dedicated experiment
- Another example is the difficulty to assess the overall sentiment of a movie review
- If data pertain naturally into different groups, than observation in each group can have their own characteristic. In this case, a different supervised model can be trained specifically for the data in each group
- Clustering can be used to perform collaborative filtering, a technique used in recommendation systems

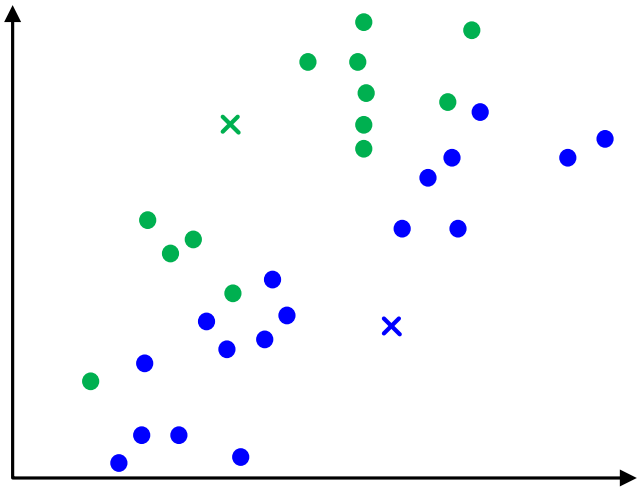
Outline

- Unsupervised learning
- **K-means clustering**
- Principal Components Analysis

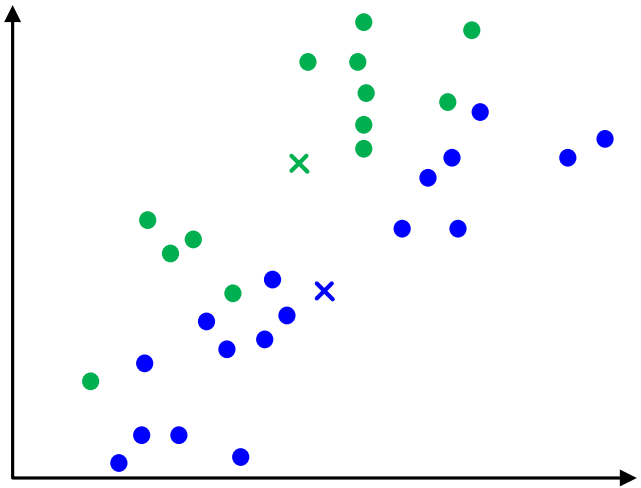
K-means example



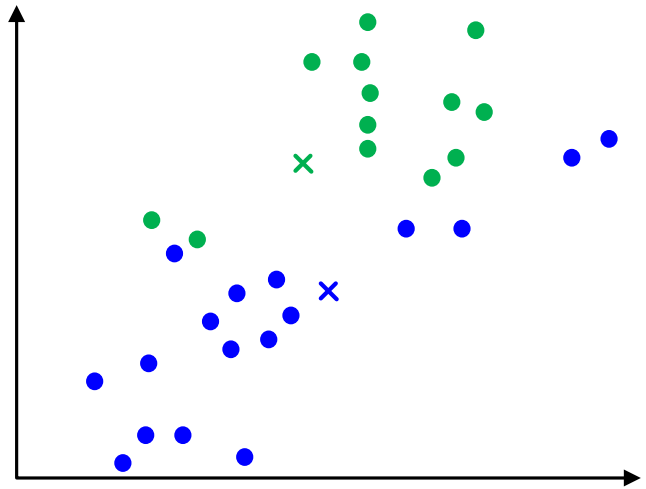
K-means example



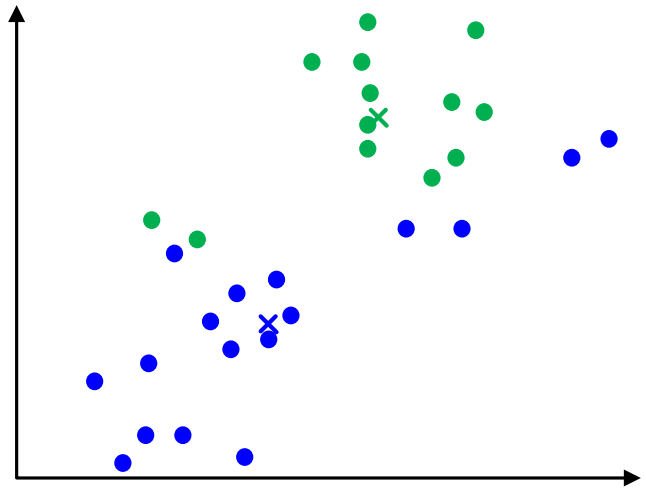
K-means example



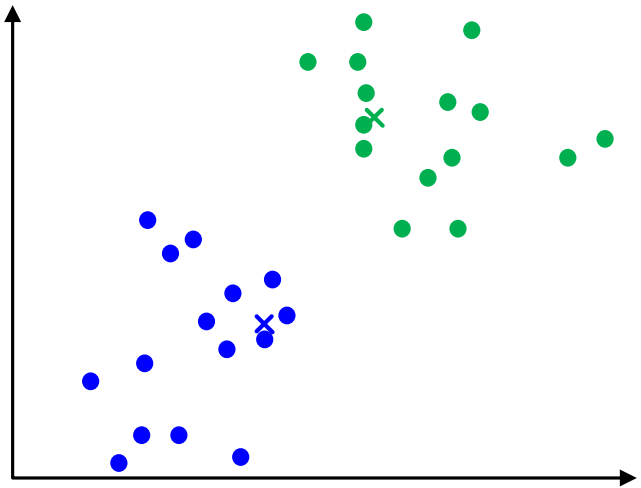
K-means example



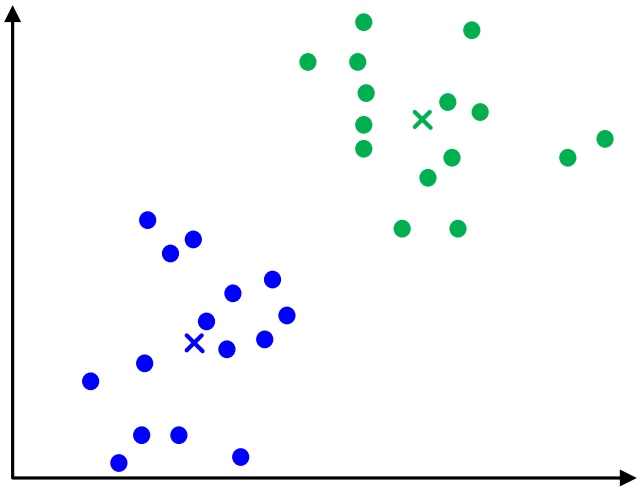
K-means example



K-means example



K-means example



K-means algorithm

The K-means algorithm is an iterative method which provides a solution to clustering problem

Inputs

- The number of clusters K
- The training set $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ $\quad \mathbf{x} \in \mathbb{R}^d$ *(do not consider $x_0 = 1$)*

Notice how the label y is not required to perform the clustering procedure

K-means algorithm

The steps of the K-means algorithm are:

Algorithm

1. Randomly initialize K cluster centroids $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots \boldsymbol{\mu}_K \in \mathbb{R}^d$
2. **Repeat**
   - 2.1 **For** $n = 1 : N$
     $c_n :=$ index (from 1 to K) of cluster centroid closest to $\mathbf{x}_n$
   - 2.2 **For** $k = 1 : K$
     $\boldsymbol{\mu}_k :=$ average (mean) of points assigned to cluster k

The steps 2.1 and 2.2 are repeated until the centroids do not change

K-means algorithm

If a cluster has no points:

- Eliminate the cluster (end with $K - 1$ clusters)
- Re-initialize the cluster centroids (if you need K clusters)

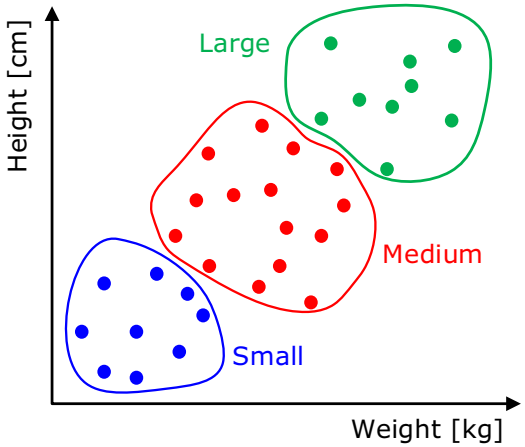
K-means for non-separated clusters

Clustering can be useful even when the clusters are not so separated

Consider a t-shirt producer that wants to know the size with which he has to produce its shirts

By clustering the population in three groups, he knows he has to satisfy the people in each cluster with that shirt size

This is an example of market segmentation



K-means optimization objective

The K-means, just as linear and logistic regression, has a cost function which tries to minimize. Define:

- c_n : index of cluster $(1, 2, \ldots, K)$ to which example $\mathbf{x}_n$ is currently assigned
- $\boldsymbol{\mu}_k$: cluster centroid k ($\boldsymbol{\mu}_k \in \mathbb{R}^d$)
- $\boldsymbol{\mu}_{c_n}$: cluster centroid of cluster to which example $\mathbf{x}_n$ has been assigned

Distortion cost function

$$J(c_1, \ldots, c_N, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K) = \frac{1}{N} \sum_{n=1}^{N} ||\mathbf{x}_n - \boldsymbol{\mu}_{c_n}||^2$$

The minimization of this cost function is computationally difficult (NP-hard). For this reason, greedy algorithms are used that converge to a local optimum

K-means optimization objective

Algorithm

1. Randomly initialize K cluster centroids $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots \boldsymbol{\mu}_K \in \mathbb{R}^d$

2. **Repeat**

   2.1 **For** $n = 1 : N \rightarrow$ Minimize $J(\ldots)$ with respect to $c_1, c_2, \ldots, c_N$ while holding $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_K$ constant

   $c_n =$ index (from 1 to K) of cluster centroid closest to $\mathbf{x}_n$

   2.2 **For** $k = 1 : K \rightarrow$ Minimize $J(\ldots)$ with respect to $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_K$ while holding $c_1, c_2, \ldots, c_N$ constant

   $\boldsymbol{\mu}_k =$ average (mean) of points assigned to cluster k

The distiortion cost function $J(c_1, \ldots, c_N, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K)$ decreases at every step

Random initialization

The initialization of the centroid is done randomly. A recommended way is:

- Randomly pick K training examples
- Set $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$ equal to these K examples

A different initialization can lead to a different clustering

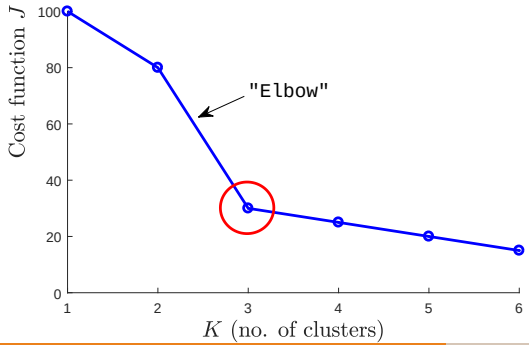
- Try multiple random initialization
- Pick the clustering that gave lowest $J(c_1, \ldots, c_N, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K)$
- With $K = 2 - 10$ trying a high number of initialization can improve performance
- With $K > 10, 100$ multiple initialization does not change a lot the final result. The first clustering should be a fairly decent solution

Chosing the number of clusters

The number of clusters K should be dictated by problem understanding:

- In the T-shirts example, one can ask: "if I have 5 clusters, how well they fit the customers? How many shirts can I sell? Will the customers be more happy?

However, there exist heuristics to choose K



Outline

- Unsupervised learning
- K-means clustering
- **Principal Components Analysis**

Principal Components Analysis motivation

Principal Component Analysis (PCA) can be used for different purposes:

1. Data compression: project the data into a lower dimensionality
2. Data visualization: visualize in a 2-D plot high-dimensional data

PCA produces a low-dimensional representation of a dataset

- Finds a sequence of linear combinations of the variables that have maximal variance
- The derived variables are mutually uncorrelated
- The derived variables can be used in supervised learning problems to speed up the computation

Data compression

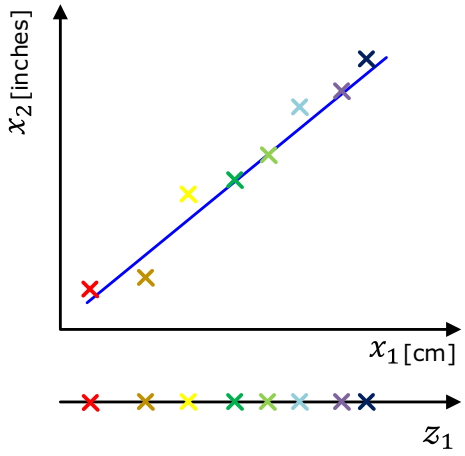
Reduce the data dimensionality from $d = 2$ to $q = 1$

Highly redundant representation

The compressed data retain the main information

Approximation but reduced space requirement

$$\mathbf{x}_1 \in \mathbb{R}^2 \longrightarrow z_1 \in \mathbb{R}$$
$$\mathbf{x}_2 \in \mathbb{R}^2 \longrightarrow z_2 \in \mathbb{R}$$
$$\vdots$$
$$\mathbf{x}_N \in \mathbb{R}^2 \longrightarrow z_N \in \mathbb{R}$$



Data visualization

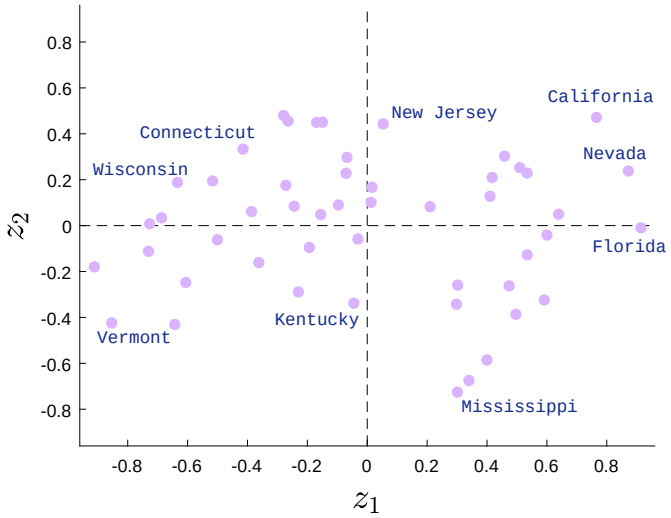
Suppose we want to explore the $\mathrm{USarrest}$ dataset, which contains crime statistics per 100.000 residents in 50 states of USA. The dimensions are:

1. Murder arrests (per 100.000)
2. Assault arrests (per 100.000)
3. Percent urban population
4. Rape arrests (per 100.000)

How to visualize the data? We have 4 features...

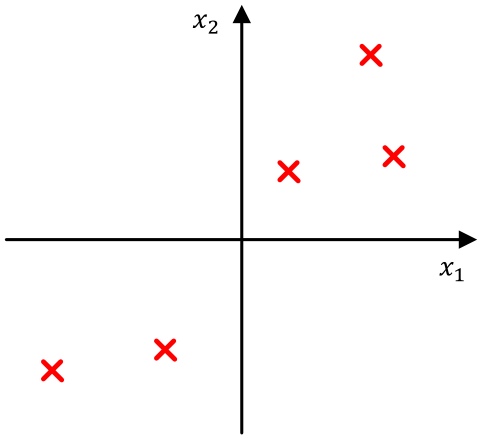
State	Murder	Assault	UrbanPop	Rape
Alabama	13.2	236	58	21.2
Alaska	10	260	48	44.5
Arizona	8.1	294	80	31
⋮	⋮	⋮	⋮	⋮

Data visualization



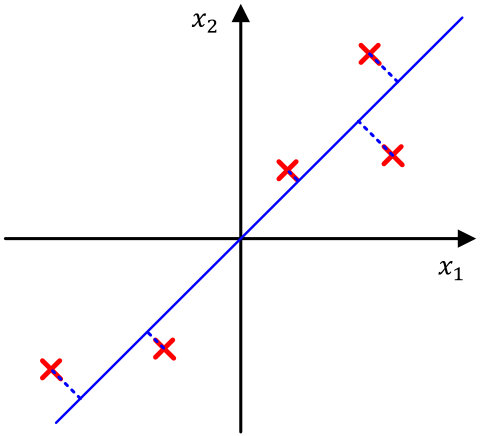
Problem formulation

Onto which direction it is better to project the data? Pick the direction which minimizes the projection error. This direction is also that on which the data vary the most



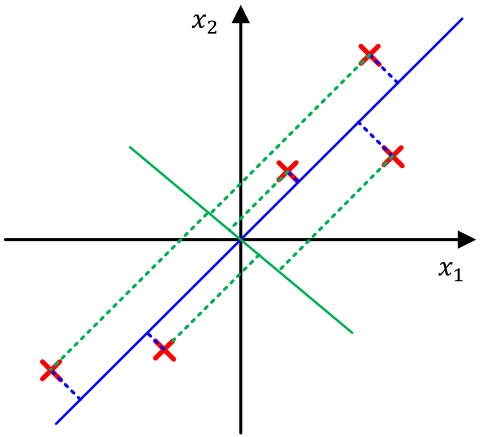
Problem formulation

Onto which direction it is better to project the data? Pick the direction which minimizes the projection error. This direction is also that on which the data vary the most



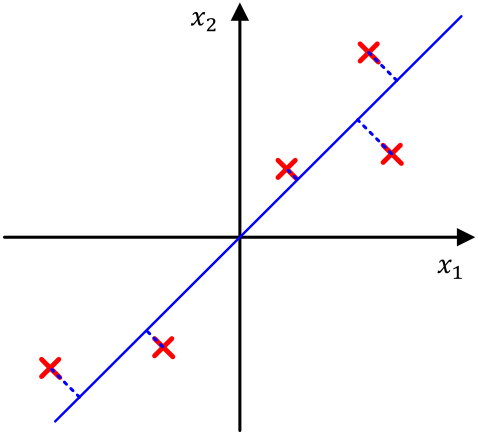
Problem formulation

Onto which direction it is better to project the data? Pick the direction which minimizes the projection error. This direction is also that on which the data vary the most

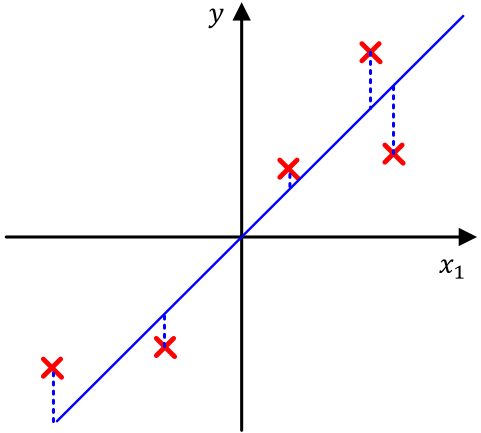


PCA is NOT linear regression

PCA



Linear regression



PCA algorithm

Inputs

- The number of dimesion to retain $q \rightarrow$ can be $q = d$
- The training set $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ $\quad \mathbf{x} \in \mathbb{R}^d$ *(do not consider $x_0 = 1$)*

Data pre-processing

- Remove the feature (column) mean for each column of the matrix $X \in \mathbb{R}^{N \times d}$
- If the features have different scales, standardize each feature element for the respective feature standard deviation
- Each feature now has mean 0 and standard deviation 1
- Call the normalized feature matrix as $\tilde{X} \in \mathbb{R}^{N \times d}$

PCA algorithm

Covariance matrix decomposition

- Perform a Singular Value Decomposition (SVD) of the matrix $\tilde{X} \in \mathbb{R}^{N \times d}$

$$\tilde{X} = USV^{\mathsf{T}}$$

$$\underset{N \times N}{U} = \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix} \quad \underset{N \times d}{S} = \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix} \quad \underset{d \times d}{V^{\mathsf{T}}} = \begin{bmatrix} & \\ & \end{bmatrix}$$

- The columns $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_N$ of U form an orthonormal basis of $\mathbb{R}^N$
- The columns $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_d$ of V form an orthonormal basis of $\mathbb{R}^d$
- The diagonal elements $s_1, s_2, s_{min\{N,d\}}$ in S are nonnegative and called singular values of $\tilde{X}$

PCA algorithm

From $\tilde{X} = USV^\mathsf{T}$ we get $V = \begin{bmatrix} | & | & & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \dots & \mathbf{v}_d \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{d \times d}$

- The columns $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_d$ are called **principal component loadings** and they are the eigenvectors of the covariance matrix $\Sigma = \frac{1}{N}\tilde{X}^\mathsf{T}\tilde{X}$
- Select the first $q \leq d$ columns of V, obtaining the reduced matrix $V_q \in \mathbb{R}^{d \times q}$
- Compute the projected data $Z = \tilde{X}V_q \in \mathbb{R}^{N \times q}$
- These are called the **principal component scores**
- It is possible to recover the data (with a loss) into their original dimension by $\tilde{X}_r = Z \cdot V_q^\mathsf{T} \in \mathbb{R}^{N \times d}$

Choosing the number of components

The singular values $s_1, s_2, \ldots, s_{\min\{N,d\}}$ in the matrix S can be used to compute the explained variance of each principal component

The explained variance of the jth component is:

$$\frac{s_j^2}{\sum_{i=1}^{\min\{N,d\}} s_i^2}$$

One can choose to retain a number of component which explain a determined level of variance in the data

Example with the USarrest data

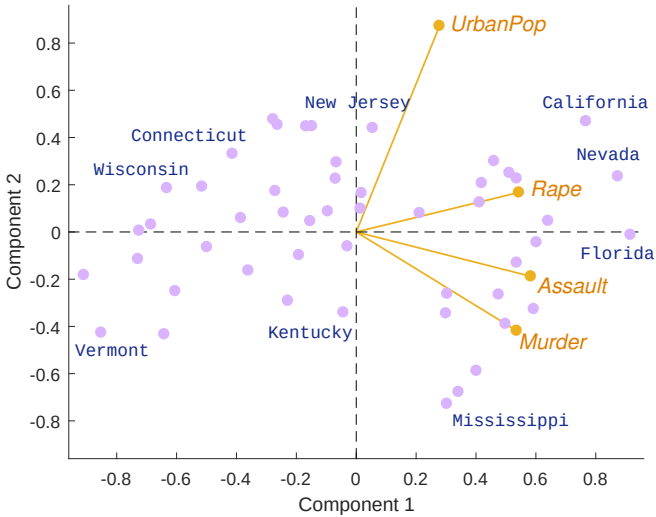
The first 2 principal component loadings for the USarrest dataset are the first 2 columns of the matrix V

	PC1	PC2
Murder	-0.5359	0.4182
Assault	-0.5832	0.1880
UrbanPop	-0.2782	-0.8728
Rape	-0.5432	-0.1673

It is possible to plot both the projected data in the new dimensions given by the principal components loadings, and the new dimensions relates to the original ones

For example, the direction of the Murder feature is the direction of the line from the origin to the point $[-0.5359, 0.4182]$

Biplot


UrbanPop
New Jersey
California
Connecticut
Nevada
Wisconsin
Rape
Florida
Assault
Murder
Kentucky
Vermont
Mississippi
Component 2
Component 1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8