Lecture 7

Principal Component Analysis (PCA)

Dr. Le Huu Ton

Outline

- PCA Introduction
- PCA Implementation

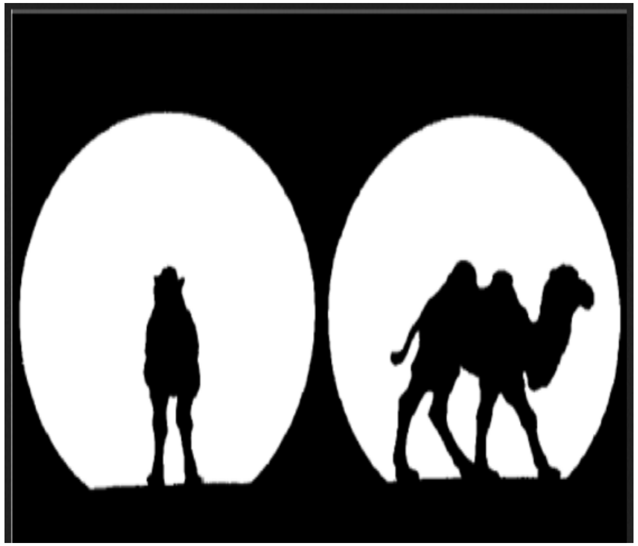
Outline

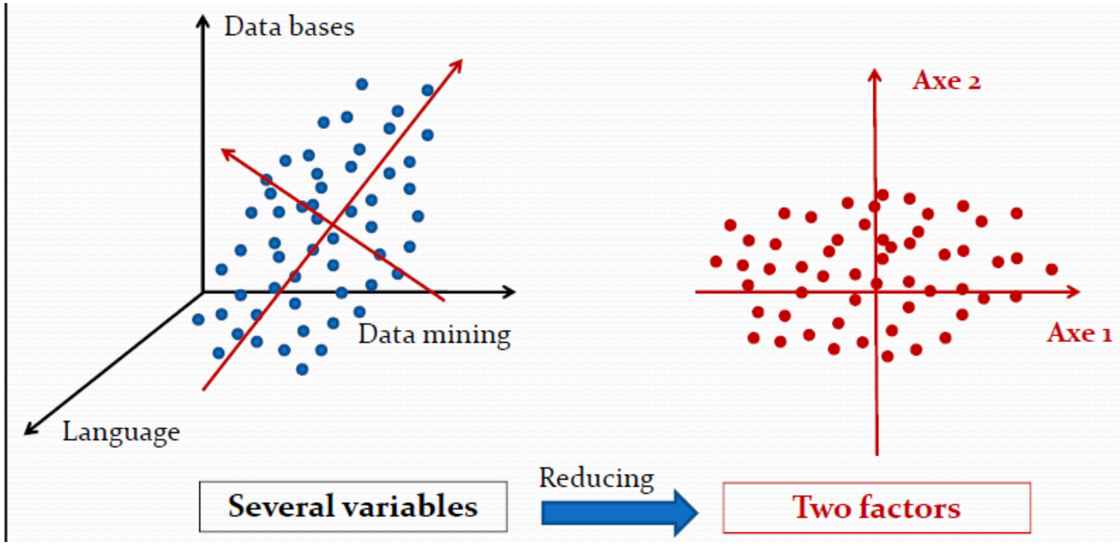
- PCA Introduction
- PCA Implementation

PCA- Introduction

Principal Component Analysis (PCA): A method to analysis and compressed data, to convert data from n-dimension space and converts it into smaller dimension space

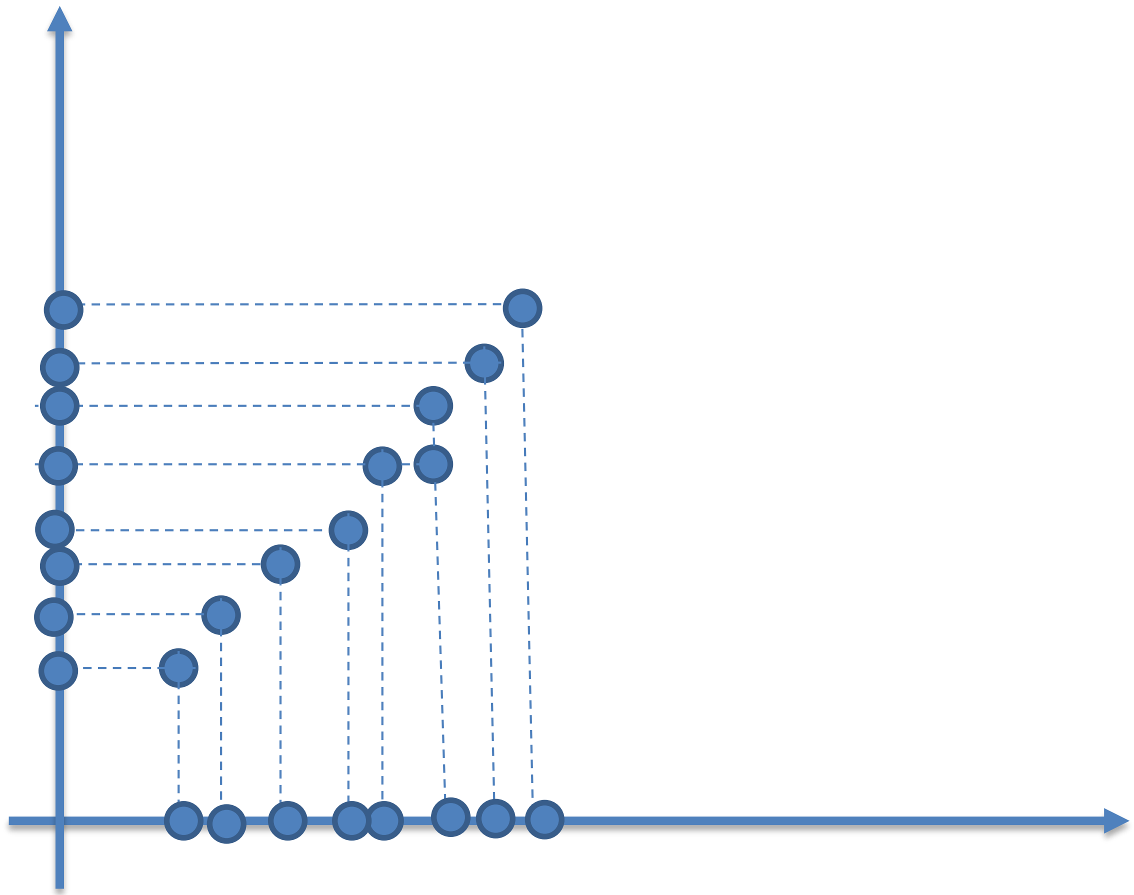
Reduce data dimension with minimize lost in variability





PCA- Introduction

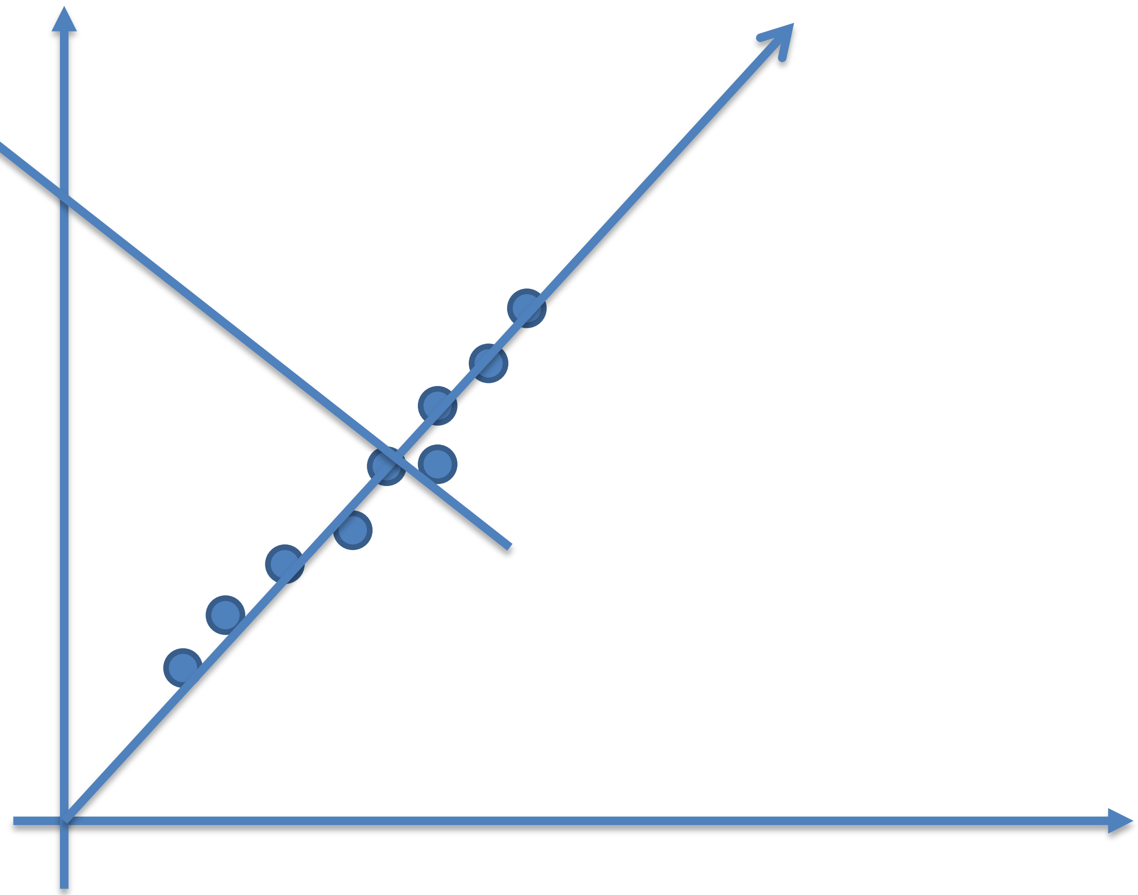
Dimension reduction: data projection



PCA- Introduction

Dimension reduction: how to find a good coordinate space to present data??

Variability



PCA- Introduction

Principal Component Analysis (PCA):

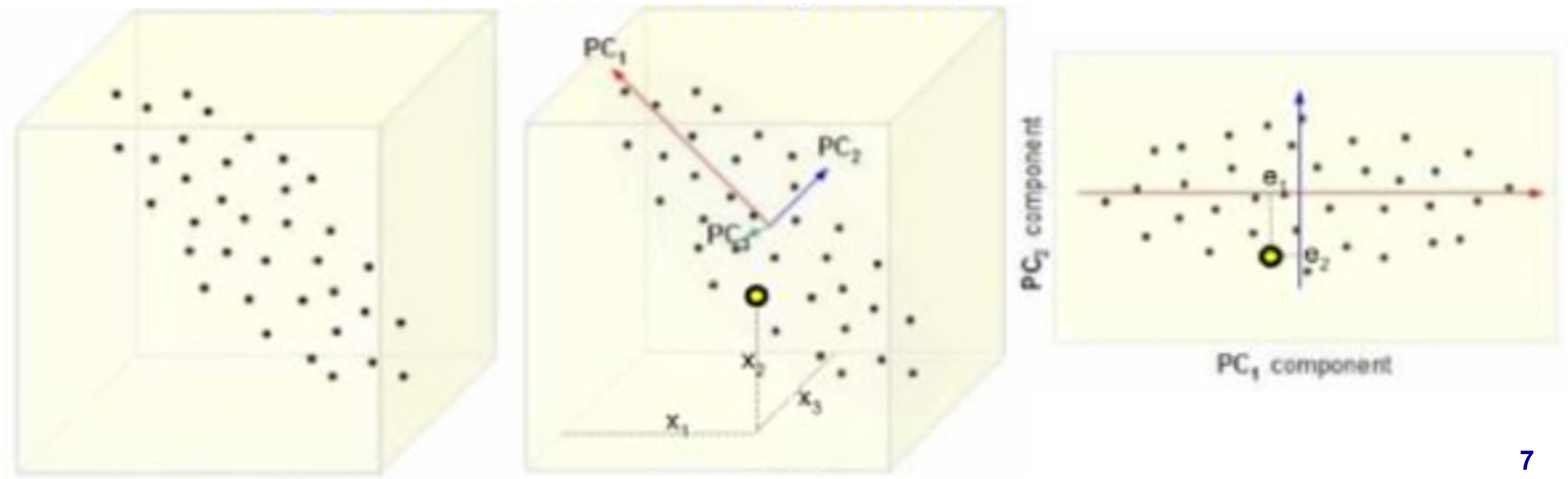
1. Defines a set of principal components:

1st : direction of greatest variability in the data

2$^{nd:}$ perpendicular to 1st , greatest variability of what is left

-… repeat n times (original dimensionality)

2. Select m first components as new dimension: convert data from original dimension to new one



Outline

- PCA Introduction
- PCA Implementation

PCA- Implementation

1. **Center the data at zero by subtracting it means**

$$for\ i = 1:n$$

$$x_i = x_i - \mu_i$$

Where μ_i **is the means of** x_i

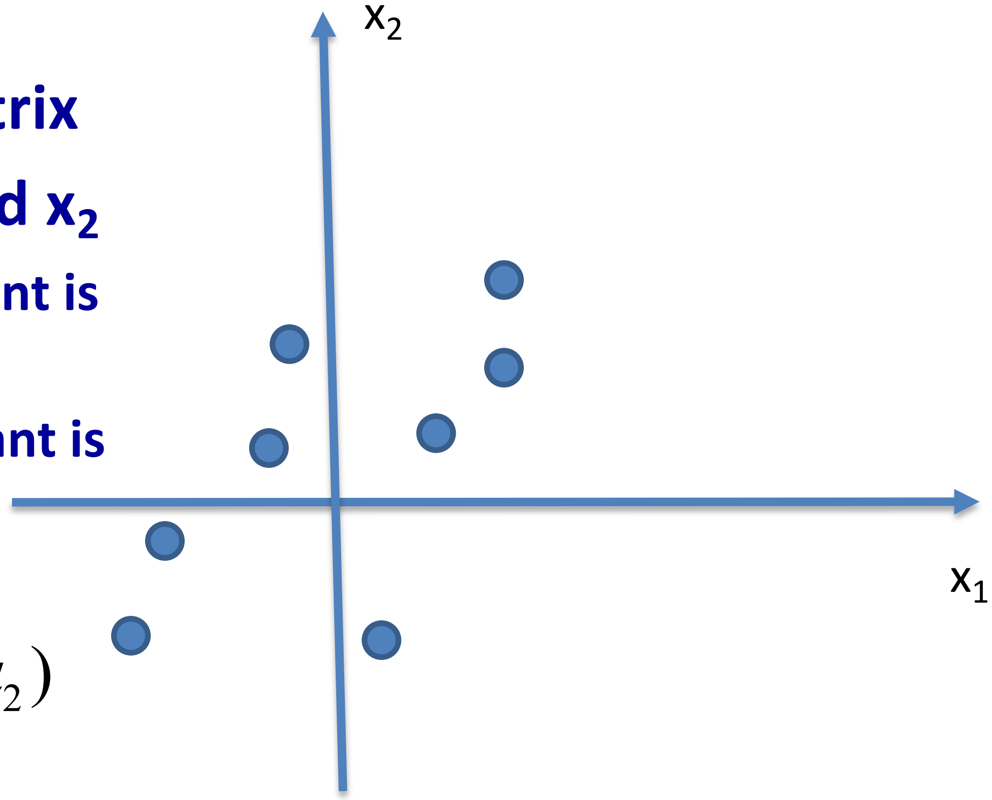
2. **Compute the covariance matrix**

- **Covariant of dimension x_1 and x_2**

x_1 increases as x_2 increases => covariant is positive

x_1 increases as x_2 decreases => covariant is negative

$$\mathrm{cov}(x_1, x_2) = \frac{1}{m}\sum_{i=1}^{m}(x_1^i - \mu_1)(x_2^i - \mu_2)$$



PCA- Implementation

1. **Center the data at zero by subtracting it means**

$$for\ i = 1:n$$

$$x_i = x_i - \mu_i$$

Where μ_i **is the means of** x_i

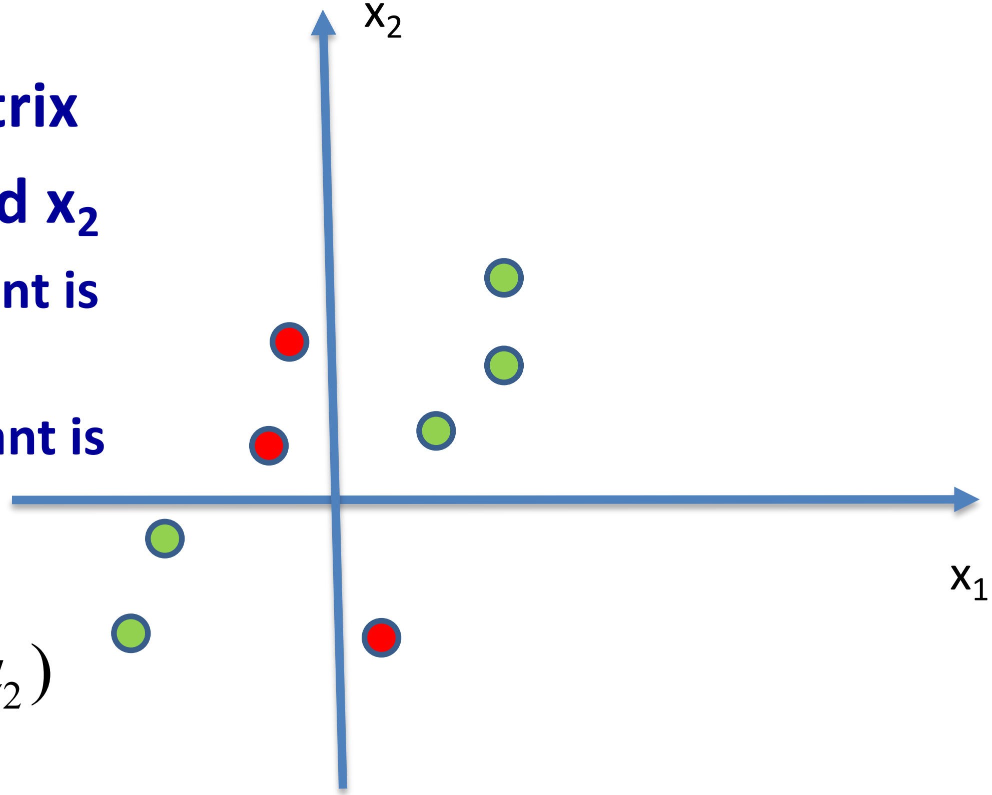
2. **Compute the covariance matrix**

- **Covariant of dimension x_1 and x_2**

x_1 increases as x_2 increases => covariant is positive

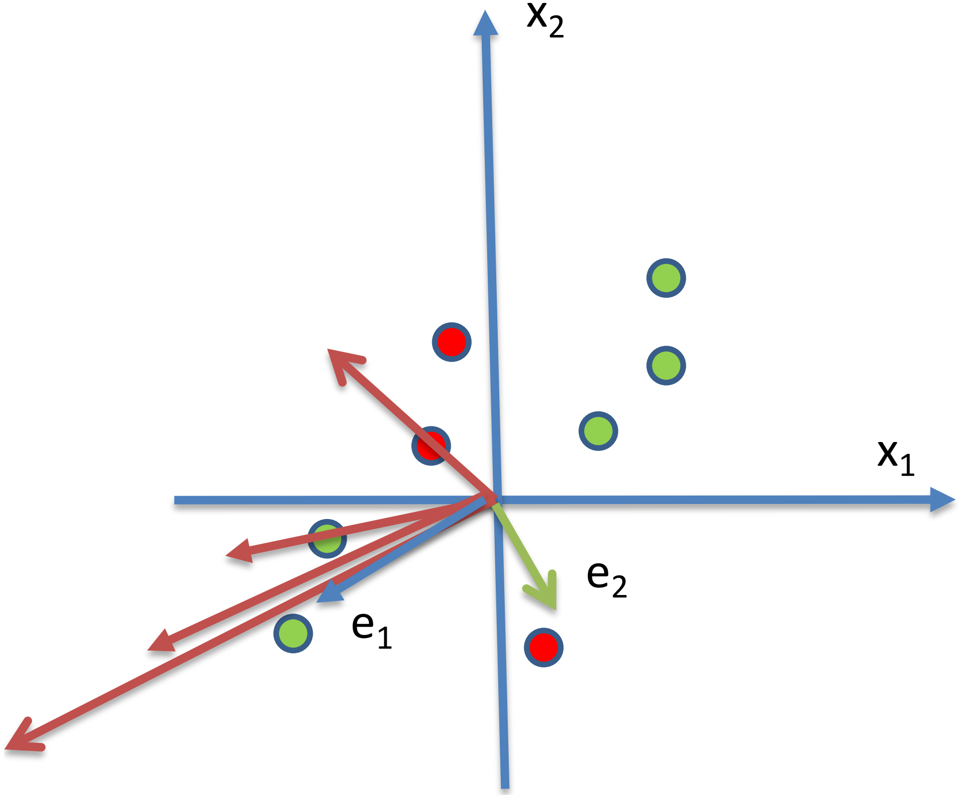
x_1 increases as x_2 decreases => covariant is negative

$$\mathrm{cov}(x_1, x_2) = \frac{1}{m}\sum_{i=1}^{m}(x_1^i - \mu_1)(x_2^i - \mu_2)$$



PCA- Implementation

Principal Component Analysis (PCA):



A vector, multiply by the covariance matrix will turn toward direction of invariance

$$\begin{pmatrix} 2 & 0.8 \\ 0.8 & 0.6 \end{pmatrix} * \begin{pmatrix} -1 \\ 1 \end{pmatrix} = \begin{pmatrix} -1.2 \\ -0.2 \end{pmatrix}$$

$$\begin{pmatrix} 2 & 0.8 \\ 0.8 & 0.6 \end{pmatrix} * \begin{pmatrix} -1.2 \\ -0.2 \end{pmatrix} = \begin{pmatrix} -2.5 \\ -1.0 \end{pmatrix}$$

$$\begin{pmatrix} -2.5 \\ -1.0 \end{pmatrix} \rightarrow \begin{pmatrix} -6 \\ -2.7 \end{pmatrix} \rightarrow \begin{pmatrix} -14.1 \\ -6.4 \end{pmatrix} \rightarrow \begin{pmatrix} -33.3 \\ -15.1 \end{pmatrix}$$

$$slope \quad 0.4 \qquad 0.45 \qquad 0.454 \qquad 0.454$$

$$\mathrm{cov}(x_1, x_2)(\sum) = \begin{vmatrix} 2 & 0.8 \\ 0.8 & 0.6 \end{vmatrix}$$

PCA- Implementation

Principal Component Analysis (PCA):

$\Rightarrow$ **We want to find the converged vector (vector that does not change its direction when multiplied by the covariance matrix)**

$$\sum * e = \lambda e$$

Those vectors that satisfy the above equation is called the eigenvectors

https://www.expunctis.com/2019/02/19/Covariance-linalg.html

PCA- Implementation

Principal Component Analysis (PCA):

1. Data preprocessing

Training set: $x^{(1)}, x^{(2)}, \ldots, x^{(m)}$

Preprocessing (feature scaling/mean normalization):

$$\mu_j = \frac{1}{m} \sum_{i=1}^{m} x_j^{(i)}$$

Replace each $x_j^{(i)}$ with $x_j - \mu_j$.

If different features on different scales (e.g., $x_1 =$ size of house, $x_2 =$ number of bedrooms), scale features to have comparable range of values.

PCA- Implementation

Principal Component Analysis (PCA):

2. Compute the eigenvectors

Principal Component Analysis (PCA) algorithm

Reduce data from n-dimensions to k-dimensions

Compute "covariance matrix":

$$\Sigma = \frac{1}{m} \sum_{i=1}^{n} (x^{(i)})(x^{(i)})^T$$

Compute "eigenvectors" of matrix Σ:

```
[U,S,V] = svd(Sigma);
```

U is a matrix made of eigenvectors, where each column of U present a eigenvectors

1st column of U => greatest variability in the data => first PCA component

2nd column of U => second PCA component

PCA- Implementation

How to convert data to the new k-dimension space (k<n)

$$U = \begin{bmatrix} | & | & & | \\ u^{(1)} & u^{(2)} & \dots & u^{(n)} \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{n \times n}$$

Take the first k column of U, called U_reduce

The coordinate of a data point x in new dimension space is

$$z = (u\_reduce)^T * x$$

U_reduce is a n*k matrix X is a n*1 vector

=> z is a k*1 vector

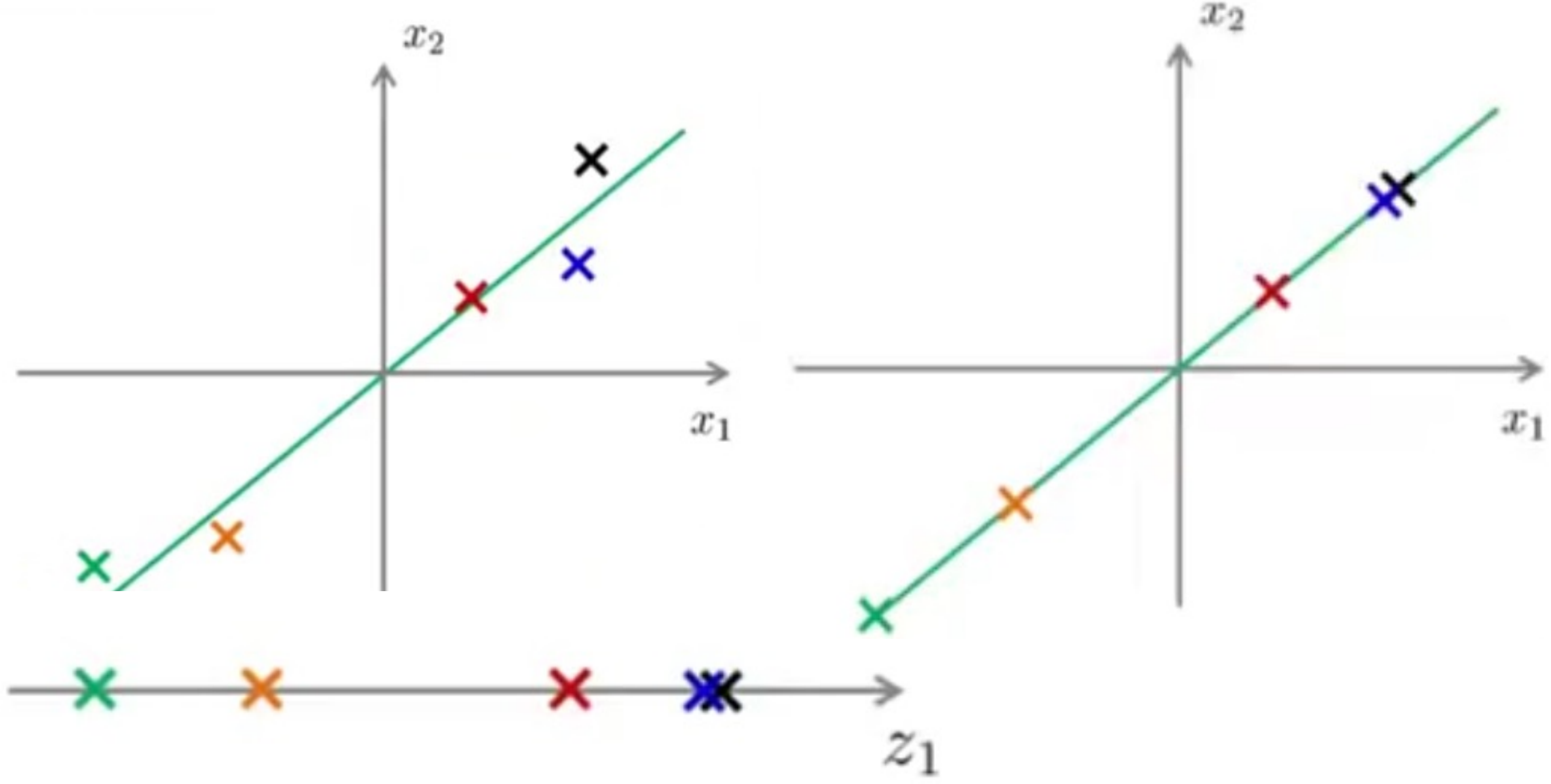
PCA- Implementation

Principal Component Analysis (PCA): Data reconstruction

$$x_{approx} = U\_reduce * z$$

Where *z* is the PCA transformation of data *x*

x_{approx} is the reconstruction of x from *z*



PCA- Implementation

Principal Component Analysis (PCA): PCA evaluation, how to choose k

Average squared projection error: $\frac{1}{m}\sum_{i=1}^{m}\|x^{(i)} - x_{approx}^{(i)}\|^2$

Total variation in the data: $\frac{1}{m}\sum_{i=1}^{m}\|x^{(i)}\|^2$

$$\frac{\frac{1}{m}\sum_{i=1}^{m}\|x^{(i)} - x_{approx}^{(i)}\|^2}{\frac{1}{m}\sum_{i=1}^{m}\|x^{(i)}\|^2} \leq 0.01$$

"99% of variance is retained"

PCA- Implementation

Principal Component Analysis (PCA): PCA evaluation, how to choose k

Algorithm:

Try PCA with $k = 1$

Compute $U_{reduce}, z^{(1)}, z^{(2)}, \dots, z^{(m)}, x^{(1)}_{approx}, \dots, x^{(m)}_{approx}$

Check if

$$\frac{\frac{1}{m}\sum_{i=1}^{m} \|x^{(i)} - x^{(i)}_{approx}\|^2}{\frac{1}{m}\sum_{i=1}^{m} \|x^{(i)}\|^2} \le 0.01?$$

[U,S,V]=svd(sigma)

$$S = \begin{vmatrix} s_{11} & 0 & \dots & 0 \\ 0 & s_{22} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & s_{nn} \end{vmatrix}$$

s_{ii} tell how much variability (information) is stored on ith component of PCA

$$\frac{\sum_{i=1}^{k} s_{ii}}{\sum_{i=1}^{n} s_{ii}}$$ presents how much variability is retained

PCA- Implementation

Principal Component Analysis (PCA): PCA evaluation, how to choose k

```
[U,S,V] = svd(Sigma)
```

Pick smallest value of k for which

$$\frac{\sum_{i=1}^{k} S_{ii}}{\sum_{i=1}^{m} S_{ii}} \geq 0.99$$

(99% of variance retained)

PCA- Implementation

Exercise: Calculate the covariance matrix of the following data

Use the 1st and 2nd component to transform the data

Data	x_1	x_2
1	4	8
2	5	9
3	5	10
4	3	7
5	2	5
6	2	3

References

http://openclassroom.stanford.edu/MainFolder/CoursePage.php?course=MachineLearning

http://www.svm-tutorial.com/2014/11/svm-understanding-math-part-2/

https://www.youtube.com/channel/UCs7alOMRnxhzfKAJ4JjZ7Wg