CS 405/605 Data Science

Dr. Qianqian Tong

Big & High-Dimensional Data

- High-Dimensions = Lot of Features

Document classification

Features per document =

thousands of words/unigrams

millions of bigrams, contextual information

Surveys - Netflix

480189 users x 17770 movies

	movie 1	movie 2	movie 3	movie 4	movie 5	movie 6
Tom	5	?	?	1	3	?
George	?	?	3	1	2	5
Susan	4	3	1	?	5	1
Beth	4	3	?	2	4	2

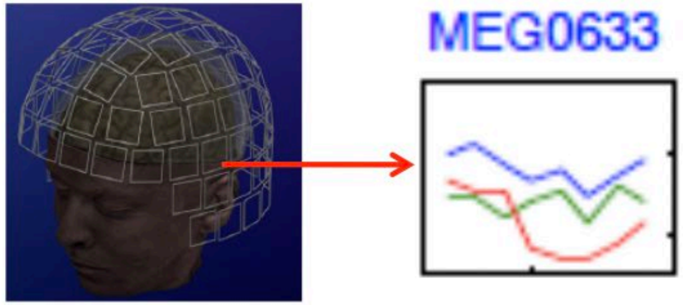
Big & High-Dimensional Data

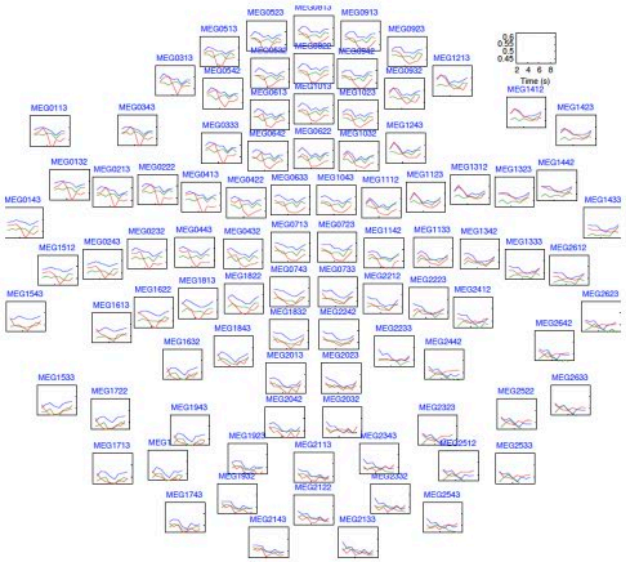
- High-Dimensions = Lot of Features

MEG Brain Imaging

120 locations x 500 time points

x 20 objects





Or any high-dimensional image data

overview

Big & High-Dimensional Data

Useful to learn lower dimensional representations of the data.

Dimensionality reduction

overview

- Today we'll cover an unsupervised learning algorithm: principal component analysis (PCA)
- Dimensionality reduction: map the data to a lower dimensional space
  - Save computation/memory
  - Reduce overfitting
  - Visualize data of high dimensionality
- PCA is a linear model, with a closed-form solution. It's useful for understanding lots of other algorithms.
  - Autoencoders
  - Matrix factorizations
  - Image compression
  - Face recognition
  - Gene expression analysis
- Today's lecture is very linear-algebra-heavy.
  - Especially orthogonal matrices and eigendecompositions.
  - Don't worry if you don't get it immediately -- next few lectures won't build on it

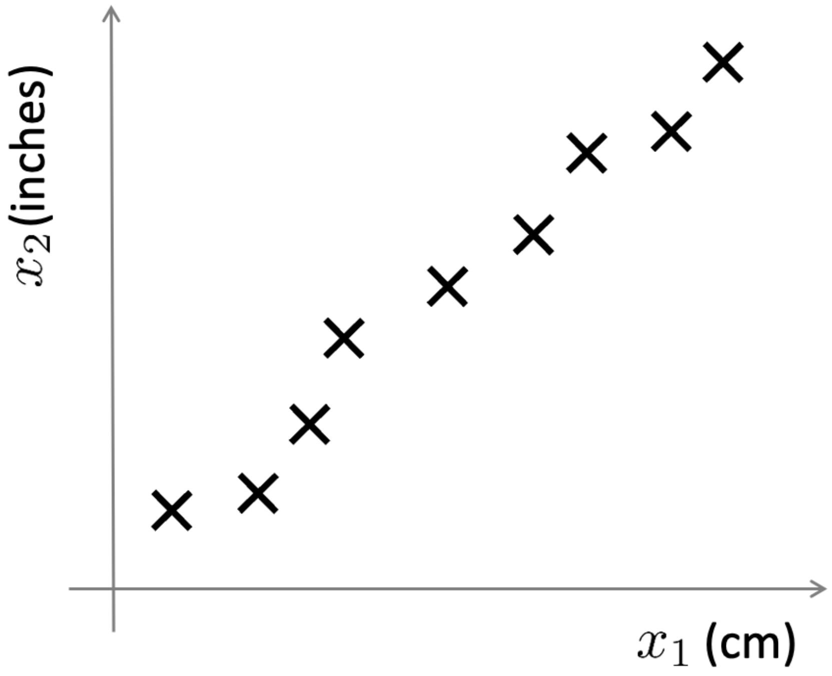
Motivation

- Dimensionality reduction
  - To simplify complex high-dimensional data
  - Summarize data with a lower dimensional real valued vector

- Given data points in *d* dimensions
- Convert them to data points in $r<d$ dimensions
- With minimal loss of information

Principal Components Analysis (PCA)

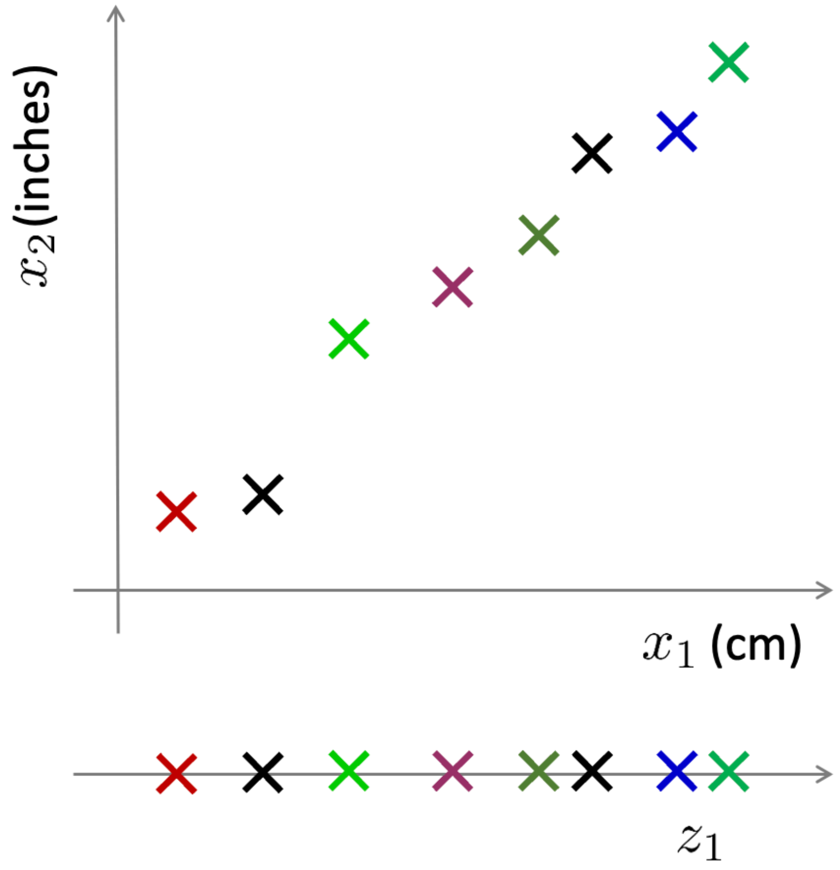
Data Compression



Reduce data from 2D to 1D

Principal Components Analysis (PCA)

Data Compression



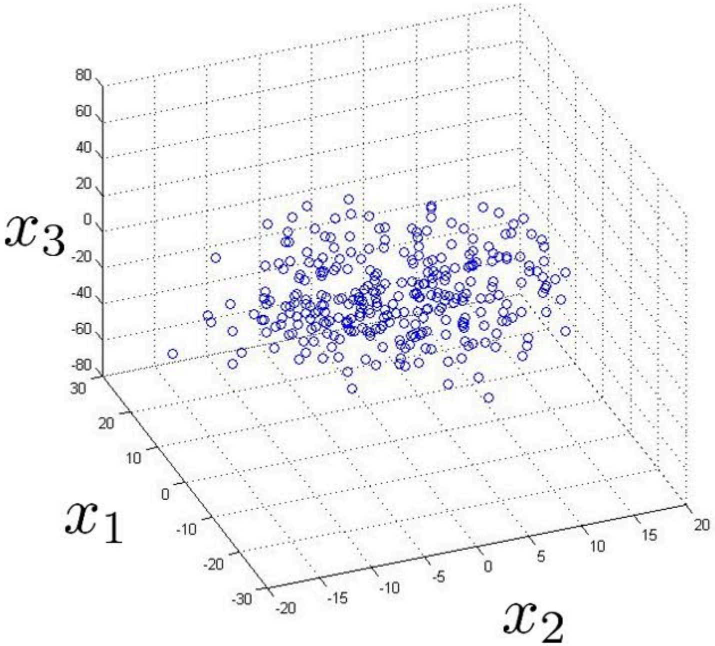
Reduce data from 2D to 1D

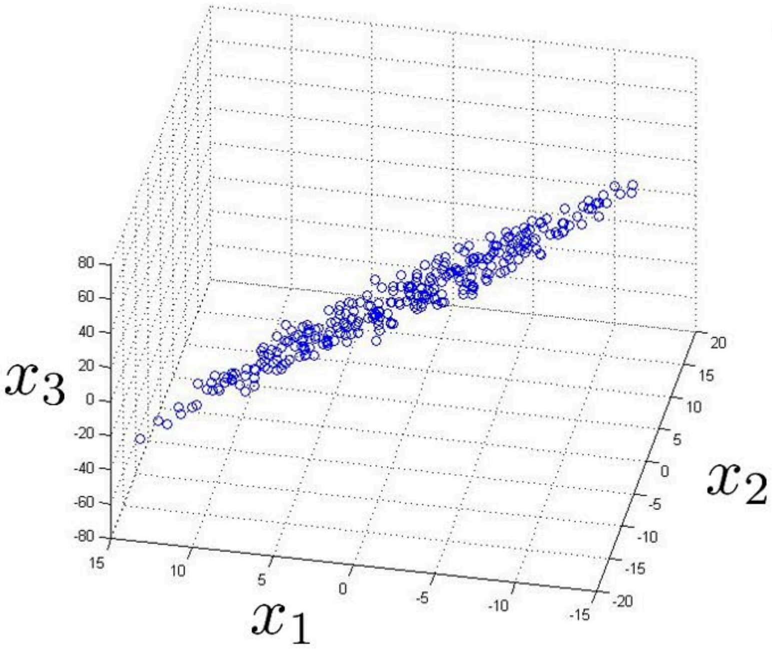
$$
\begin{aligned}
x^{(1)} &\rightarrow z^{(1)} \\
x^{(2)} &\rightarrow z^{(2)} \\
&\vdots \\
x^{(m)} &\rightarrow z^{(m)}
\end{aligned}
$$

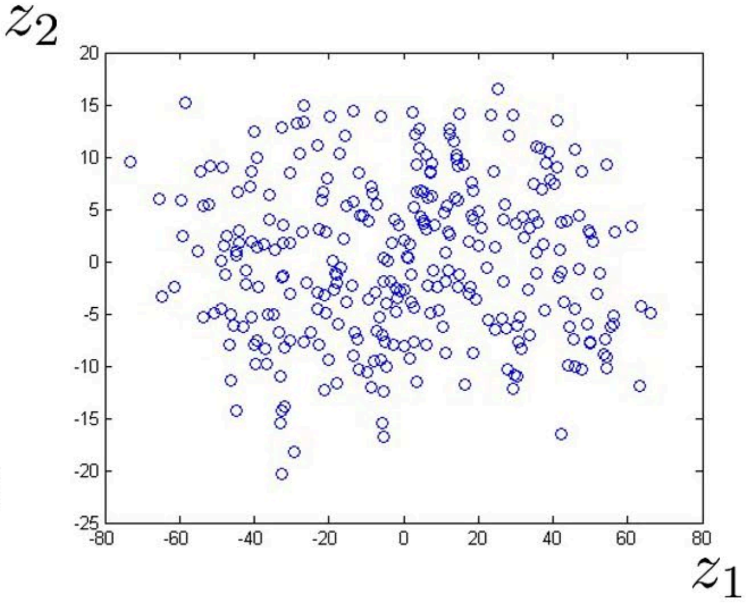
Principal Components Analysis (PCA)

Data Compression

Reduce data from 3D to 2D



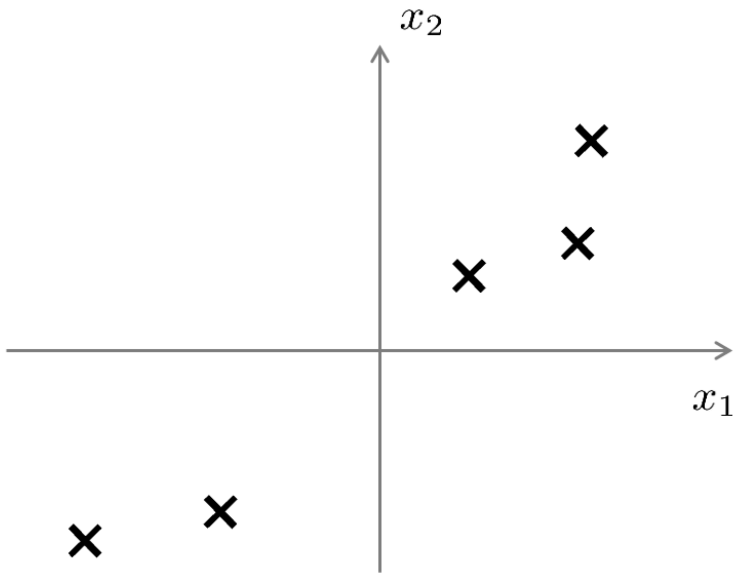




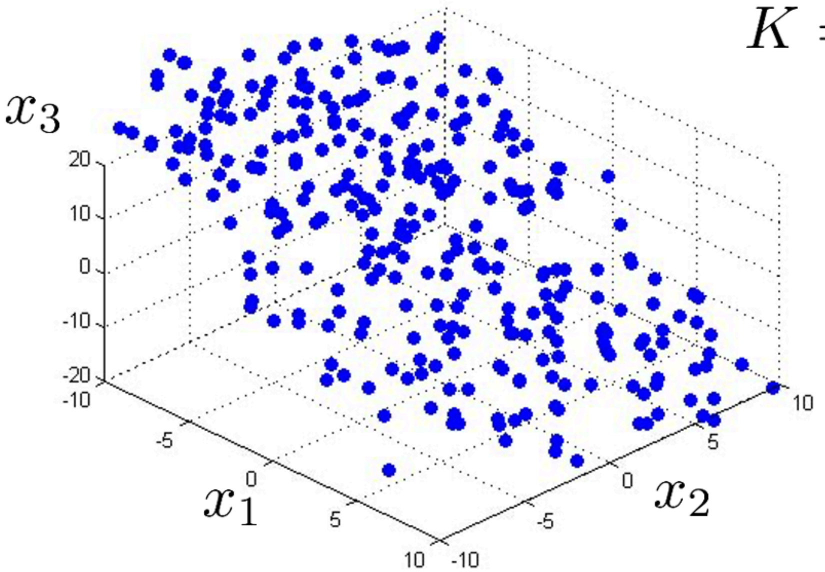
UNC GREENSBORO

Principal Components Analysis (PCA)

Principal Component Analysis (PCA) problem formulation



$3D \rightarrow 2D$
$K = 2$



Reduce from 2-dimension to 1-dimension: Find a direction (a vector $u^{(1)} \in \mathbb{R}^n$) onto which to project the data so as to minimize the projection error.

Reduce from n-dimension to k-dimension: Find k vectors $u^{(1)}, u^{(2)}, \ldots, u^{(k)}$ onto which to project the data, so as to minimize the projection error.

Principal Components Analysis (PCA)

Definition:

Choosing a subspace to maximize the projected variance, or minimize the reconstruction error, is called principal component analysis (PCA).

Principal Component Analysis

Goal: Find r-dim projection that best preserves variance

1. Compute mean vector μ and covariance matrix Σ of original points
2. Compute eigenvectors and eigenvalues of Σ
3. Select top r eigenvectors
4. Project points onto subspace spanned by them:

$$y = A(x - \mu)$$

where y is the new point, x is the old one, and the rows of A are the eigenvectors

Variance and Covariance

- Variance and Covariance:
  - Measure of the “spread” of a set of points around their center of mass(mean)
- Variance:
  - Measure of the deviation from the mean for points in one dimension
- Covariance:
  - Measure of how much each of the dimensions vary from the mean with **respect to each other**

Variance and Covariance

one attribute first

- Question: how much spread is in the data along the axis? (distance to the mean)
- Variance=Standard deviation^2

$$s^2 = \frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{(n-1)}$$

Temperature
42
40
24
30
15
18
15
30
15
30
35
30
40
30

Variance and Covariance

Now consider two dimensions

Covariance: measures the correlation between X and Y

- cov(X,Y)=0: independent
- Cov(X,Y)>0: move same dir
- Cov(X,Y)<0: move oppo dir

$$\mathrm{cov}(X,Y) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{(n-1)}$$

X=Temperature	Y=Humidity
40	90
40	90
40	90
30	90
15	70
15	70
15	70
30	90
15	70
30	70
30	70
30	90
40	70
30	90

UNC GREENSBORO

More than two attributes: covariance matrix

- Contains covariance values between all possible dimensions (=attributes):

$$C^{nxn} = (c_{ij} \mid c_{ij} = \text{cov}(Dim_i, Dim_j))$$

- Example for three attributes (x,y,z):

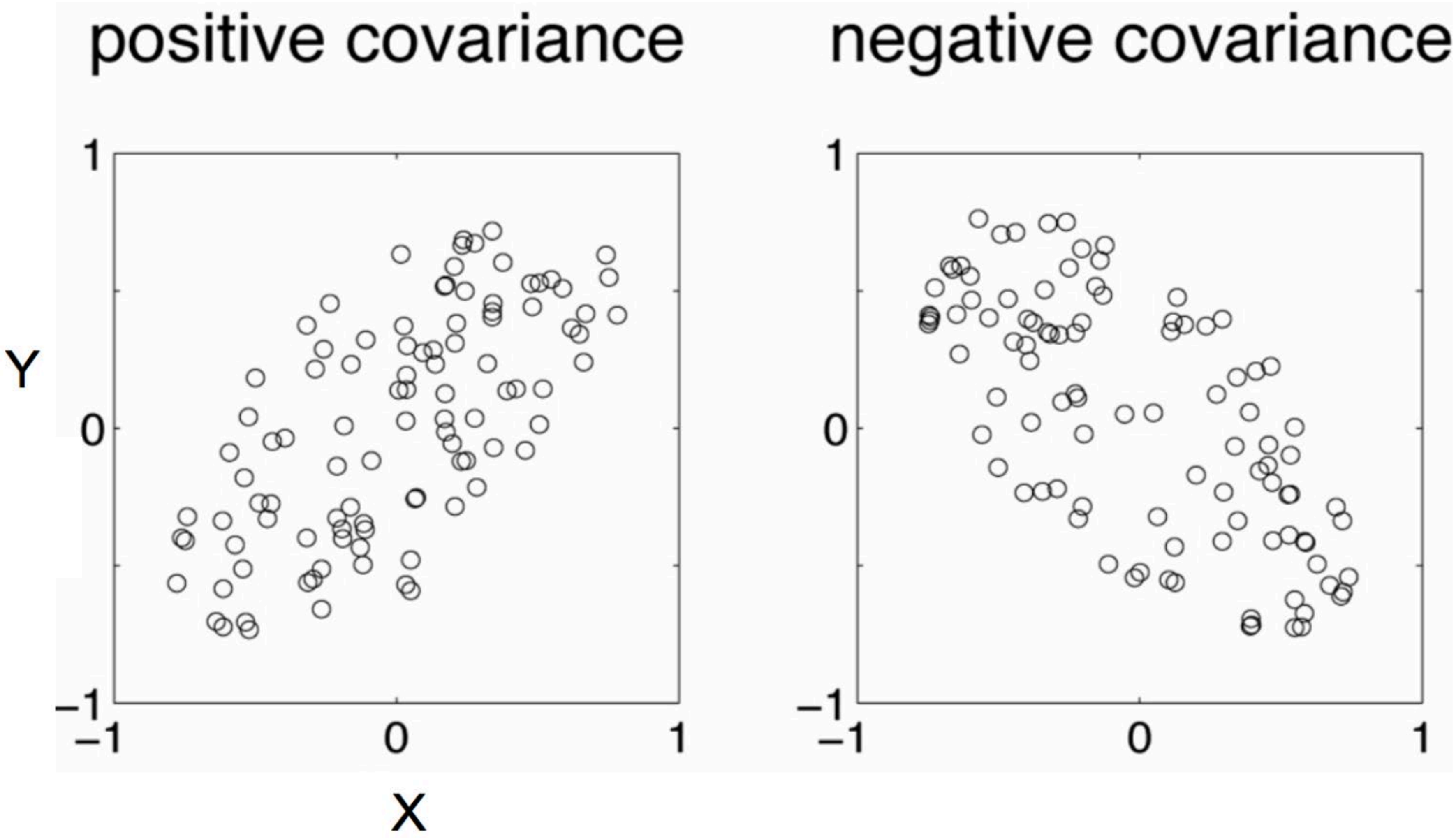
$$C = \begin{pmatrix} \text{cov}(x,x) & \text{cov}(x,y) & \text{cov}(x,z) \\ \text{cov}(y,x) & \text{cov}(y,y) & \text{cov}(y,z) \\ \text{cov}(z,x) & \text{cov}(z,y) & \text{cov}(z,z) \end{pmatrix}$$

Variance and Covariance

- Variance and Covariance:
  - Measure of the “spread” of a set of points around their center of mass(mean)
- Variance:
  - Measure of the deviation from the mean for points in one dimension
- Covariance:
  - Measure of how much each of the dimensions vary from the mean with **respect to each other**

- **Covariance is measured between two dimensions**
- **Covariance sees if there is a relation between two dimensions**
- **Covariance between one dimension is the variance**

Covariance



Positive: Both dimensions increase or decrease together

Negative: While one increase the other decrease

Eigenvector and Eigenvalue

$$\mathbf{Ax = \lambda x}$$

A: Square Matrix

X: Eigenvector or characteristic vector

λ : Eigenvalue or characteristic value

- Eigenvectors having same direction as *A*

Eigenvector and Eigenvalue

$$\mathbf{Ax} = \lambda\mathbf{x}$$

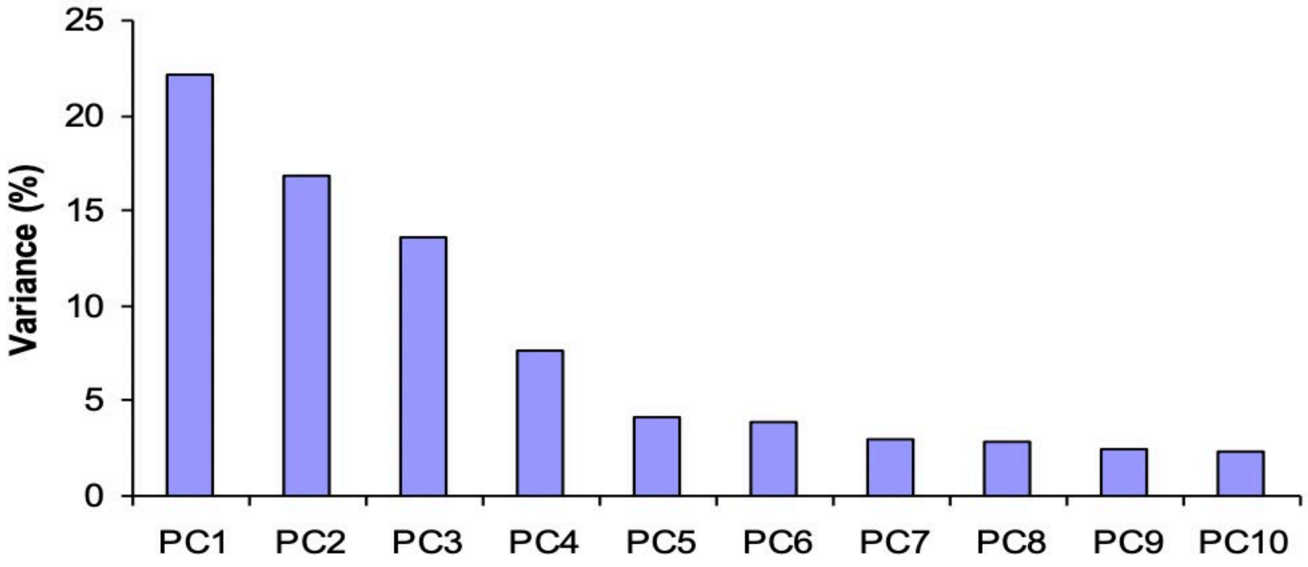
- $A\mathbf{x}=\lambda\mathbf{x} \Leftrightarrow (A-\lambda \mathrm{I})\mathbf{x}=0$
- How to calculate $\mathbf{x}$ and λ:
  - Calculate $det(A-\lambda I)$, yields a polynomial (degree n)
  - Determine roots to $det(A-\lambda I)=0$, roots are eigenvalues λ
  - Solve $(A- \lambda I)\ \mathbf{x}=0$ for each λ to obtain eigenvectors $\mathbf{x}$

Principal components

- 1. principal component (PC1)
  - The eigenvalue with the largest absolute value will indicate that the data have the largest variance along its eigenvector, the direction along which there is greatest variation
- 2. principal component (PC2)
  - the direction with maximum variation left in data, orthogonal to the 1. PC
- In general, only few directions manage to capture most of the variability in the data.

Principal components

- How many principal components?
  - For n original dimensions, sample covariance matrix is nxn, and has up to n eigenvectors. So n PCs.
- Where does dimensionality reduction come from?
  - Can ignore the components of lesser significance.



Principal components

- How many principal components?
  - For n original dimensions, sample covariance matrix is nxn, and has up to n eigenvectors. So n PCs.
- Where does dimensionality reduction come from?
  - Can ignore the components of lesser significance.
  - You do lose some information, but if the eigenvalues are small, you don't lose much
    - n dimensions in original data
    - calculate n eigenvectors and eigenvalues
    - choose only the first p eigenvectors, based on their eigenvalues
    - final data set has only p dimensions

Principal Components Analysis (PCA)

Definition:

Choosing a subspace to maximize the projected variance, or minimize the reconstruction error, is called principal component analysis (PCA).

Principal Component Analysis

Goal: Find r-dim projection that best preserves variance

1. Compute mean vector μ and covariance matrix Σ of original points
2. Compute eigenvectors and eigenvalues of Σ
3. Select top r eigenvectors
4. Project points onto subspace spanned by them:

$$y = A(x - \mu)$$

where y is the new point, x is the old one, and the rows of A are the eigenvectors

Principal Components Analysis (PCA)

Application: Image compression

Original Image

- Divide the original 372x492 image into patches:
  - Each patch is an instance that contains 12x12 pixels on a grid
- View each as a 144-D vector

Principal Components Analysis (PCA)

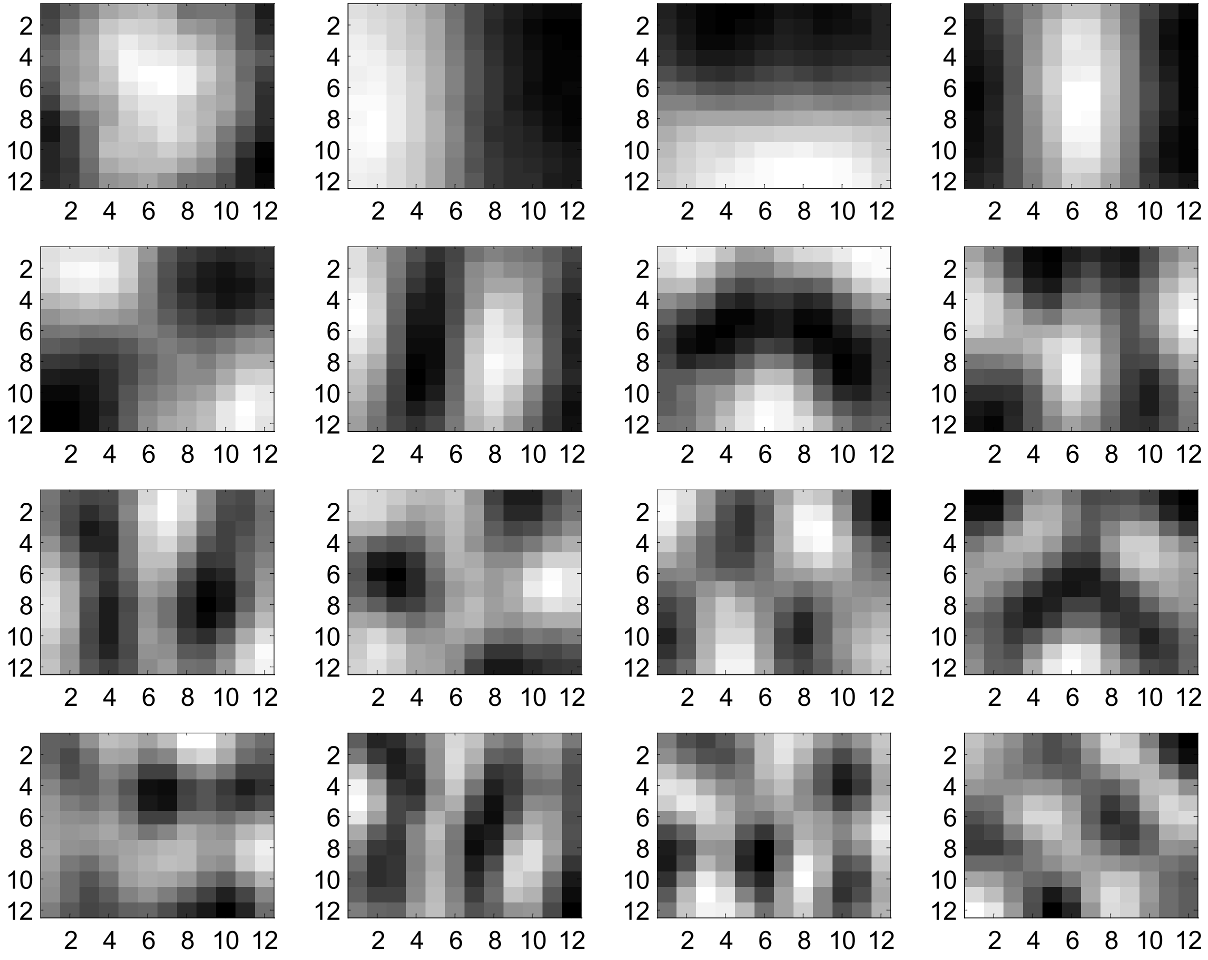
PCA compression: 144D → 60D

Principal Components Analysis (PCA)

PCA compression: 144D → 16D

Principal Components Analysis (PCA)

16 most important eigenvectors



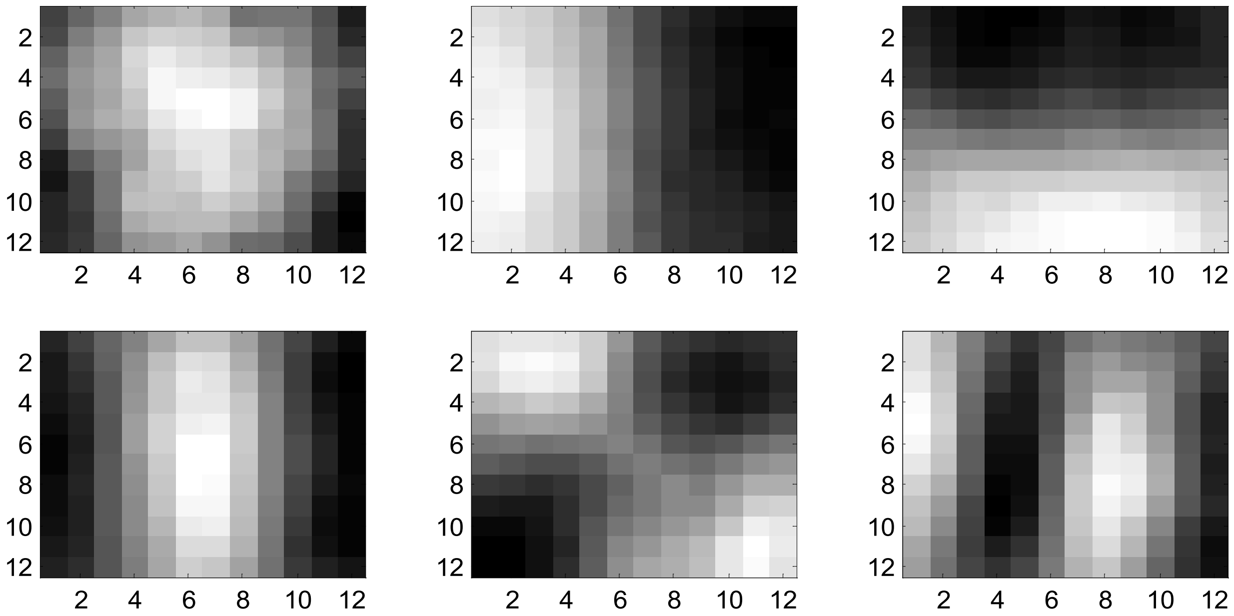
Principal Components Analysis (PCA)

PCA compression: 144D → 6D

UNC GREENSBORO

Principal Components Analysis (PCA)

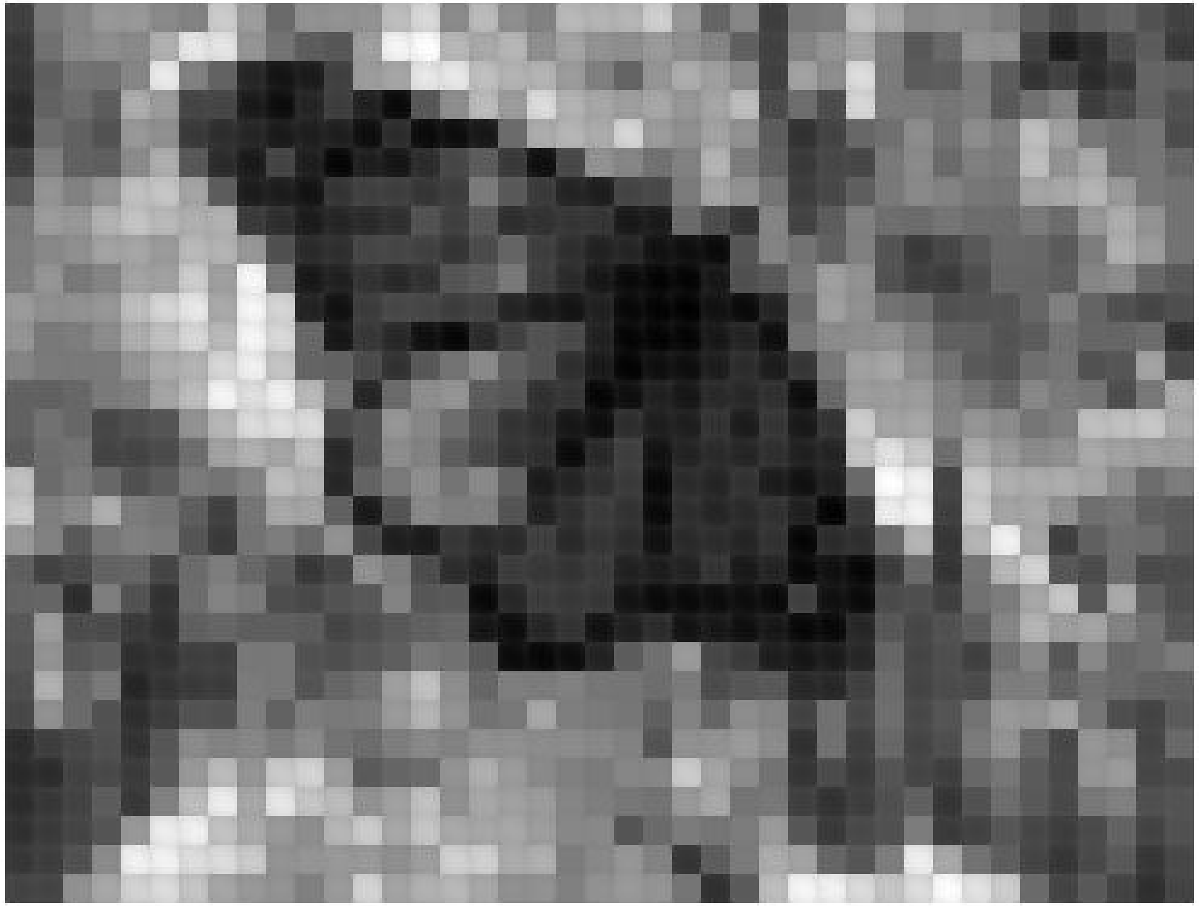
6 most important eigenvectors



UNC GREENSBORO

Principal Components Analysis (PCA)

PCA compression: 144D → 1D



Dimensionality reduction

Further reading:

- PCA (Principal Component Analysis):
  - Find projection that maximize the variance
- ICA (Independent Component Analysis):
  - Very similar to PCA except that it assumes non-Gaussian features
- Multidimensional Scaling:
  - Find projection that best preserves inter-point distances
- LDA(Linear Discriminant Analysis):
  - Maximizing the component axes for class-separation
- …

Exercise (15 minutes)

- **Objective**: To understand and apply PCA for dimensionality reduction and visualize the Iris dataset in a 2D space.

1. Data loading and exploration:

```
from sklearn.datasets import load_iris
import pandas as pd
iris = load_iris()
df = pd.DataFrame(iris.data, columns=iris.feature_names)
df['species'] = iris.target
```

2. Data Standardization:

```
from sklearn.preprocessing import StandardScaler
features = iris.feature_names
x = df.loc[:, features].values
x = StandardScaler().fit_transform(x)
```

3. Applying PCA (reduce the data to 2 principal components for visualization):

```
from sklearn.decomposition import PCA
pca = PCA(n_components=2)
principalComponents = pca.fit_transform(x)
principalDf = pd.DataFrame(data=principalComponents, columns=['Principal Component 1’, 'Principal Component 2'])
```

4. Visualization:

```
import matplotlib.pyplot as plt

fig = plt.figure(figsize=(8, 8))
ax = fig.add_subplot(1, 1, 1)
ax.set_xlabel('Principal Component 1', fontsize=15)
ax.set_ylabel('Principal Component 2', fontsize=15)
ax.set_title('2 Component PCA', fontsize=20)

targets = [0, 1, 2]
colors = ['r', 'g', 'b’]
for target, color in zip(targets, colors):
        indicesToKeep = df['species'] == target
        ax.scatter(principalDf.loc[indicesToKeep, 'Principal Component 1'], principalDf.loc[indicesToKeep,
        'Principal Component 2'], c=color, s=50)
ax.legend(iris.target_names)
ax.grid()
```