Dimensional Reduction Techniques

Outline

- Stats
- EigenTrash
- PCA
- Motivation
- Techniques:
  - LDA
  - PCA
  - CCA
  - ICA
- LDA in-depth

+ Journey to the PCA

Covariance

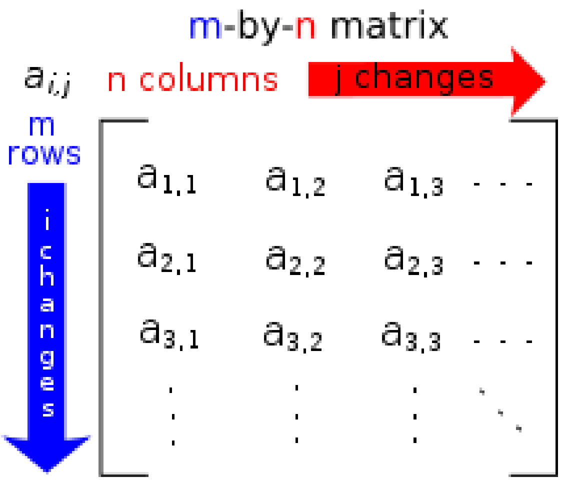
↓

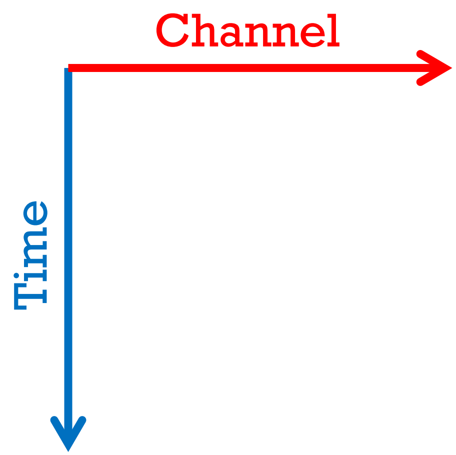
EigenValues & EigenVectors

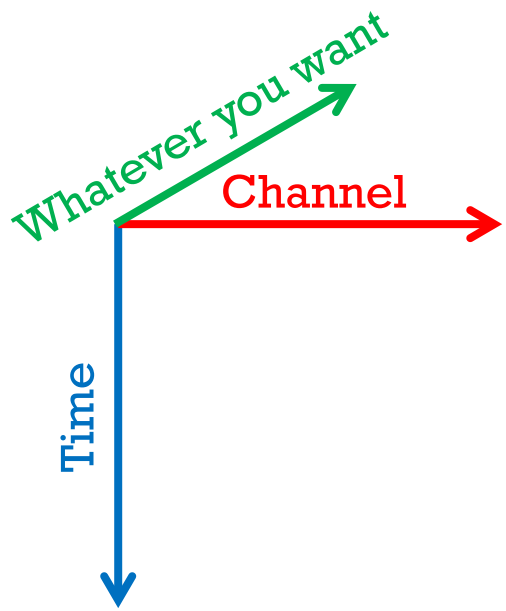
↓

Principal Component Analysis

Data as Matrices







Etc. etc…
Into higher dimensions!

Stats: Variance & Std. Dev.

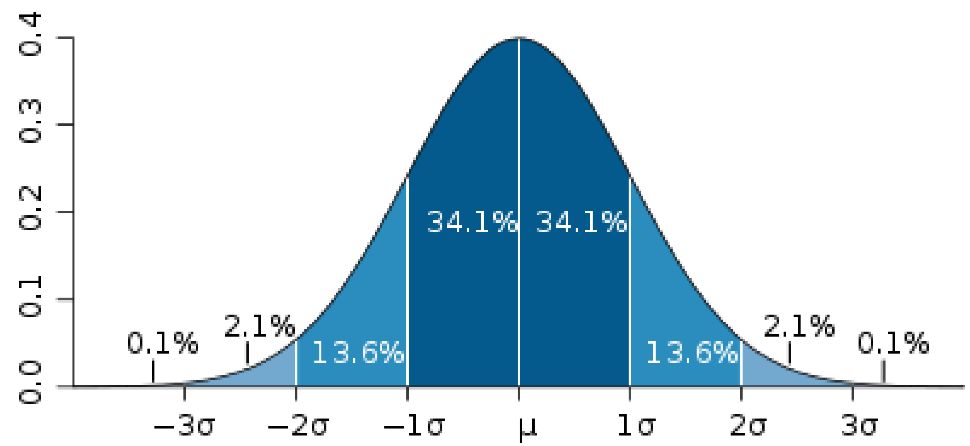
Variance of X

Expected Value (Avg)

X Data Values

Avg. of X

$$\mathrm{Var}(X) = \mathrm{E}[(X - \mu)^2]$$



Stats: Covariance

It's like the variance

$$\mathrm{Cov}(X,Y) = \mathrm{E}\big[(X - \mathrm{E}[X])(Y - \mathrm{E}[Y])\big]$$

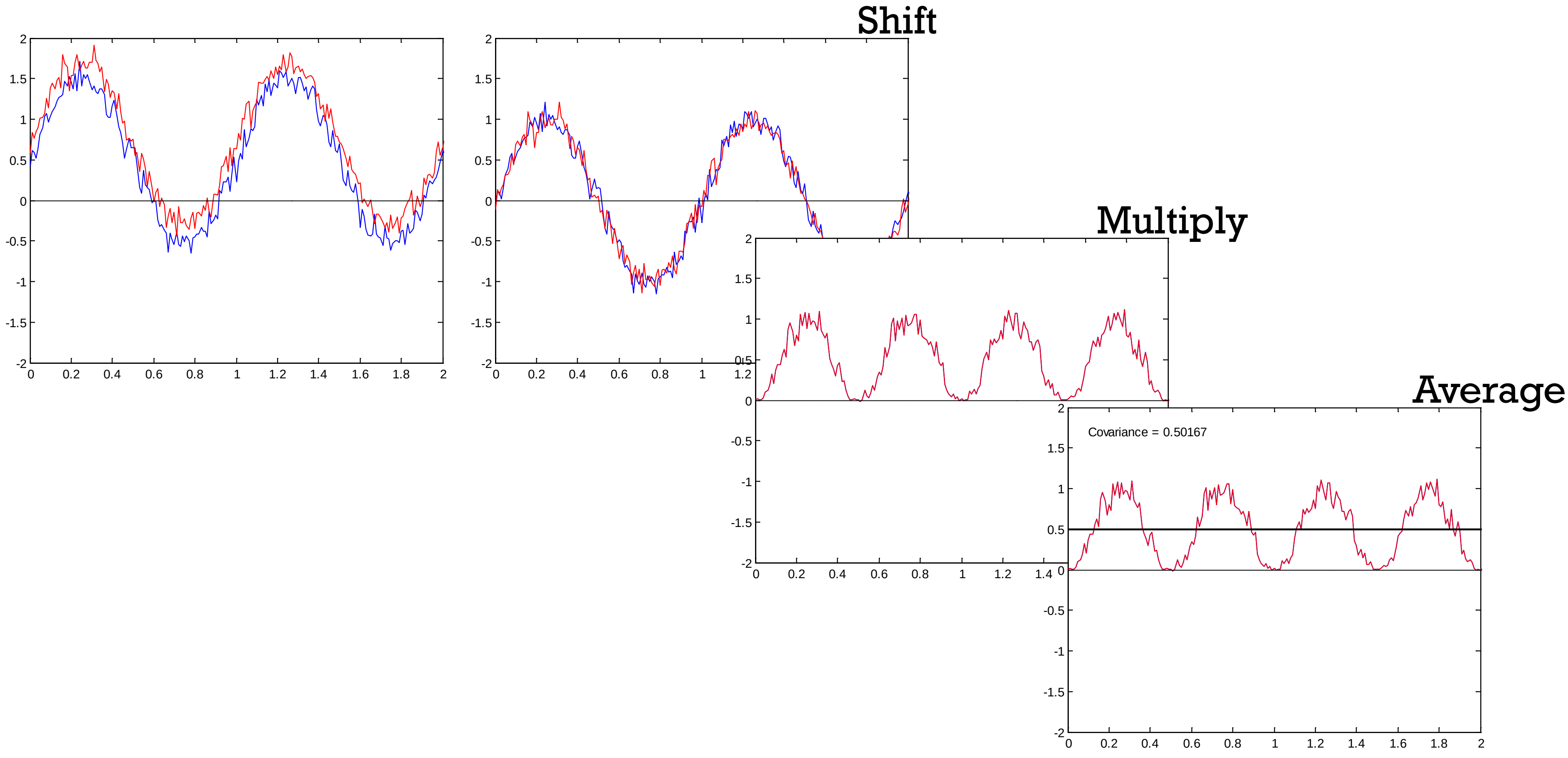
What is Cov(X,X)?

It is the variance!

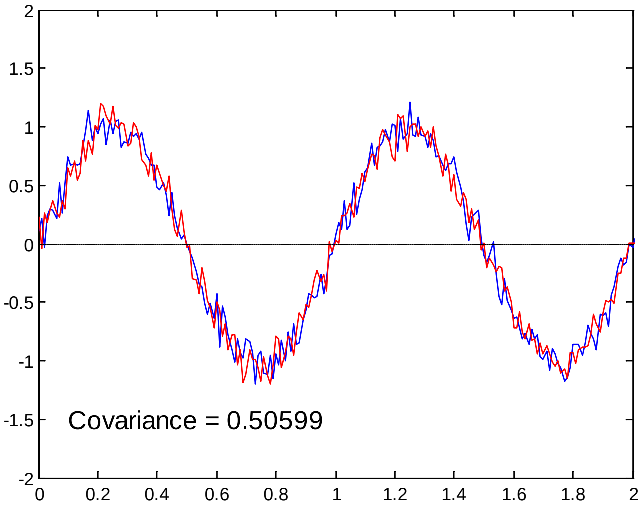
$$\mathrm{Cov}(X,X) = \mathrm{Var}(X)$$

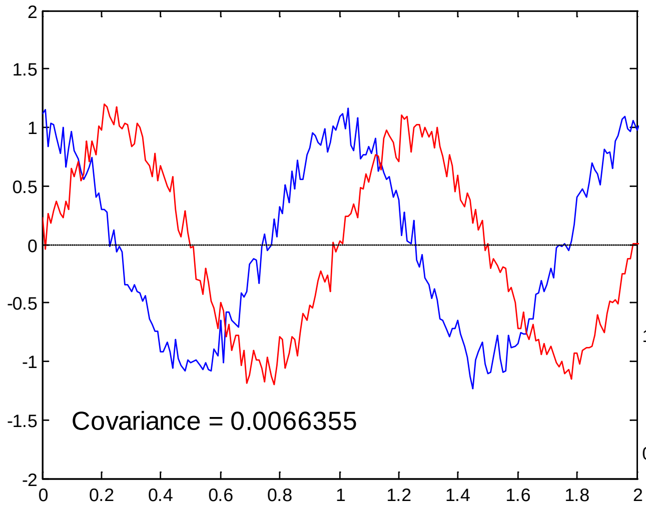
Stats: Covariance

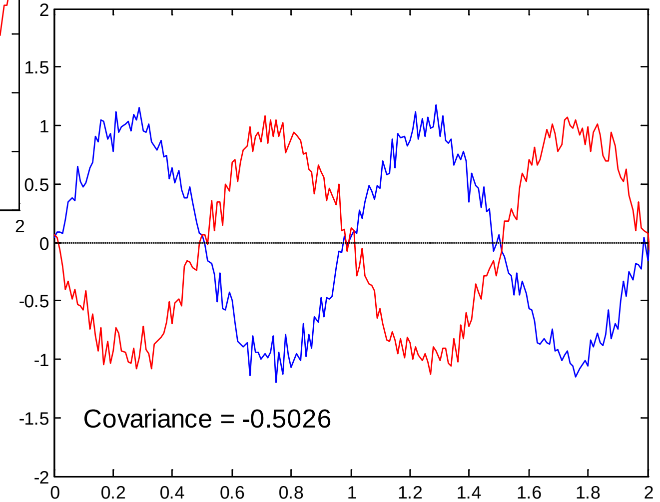
$$\mathrm{Cov}(X,Y) = \mathrm{E}\big[(X - \mathrm{E}[X])(Y - \mathrm{E}[Y])\big]$$



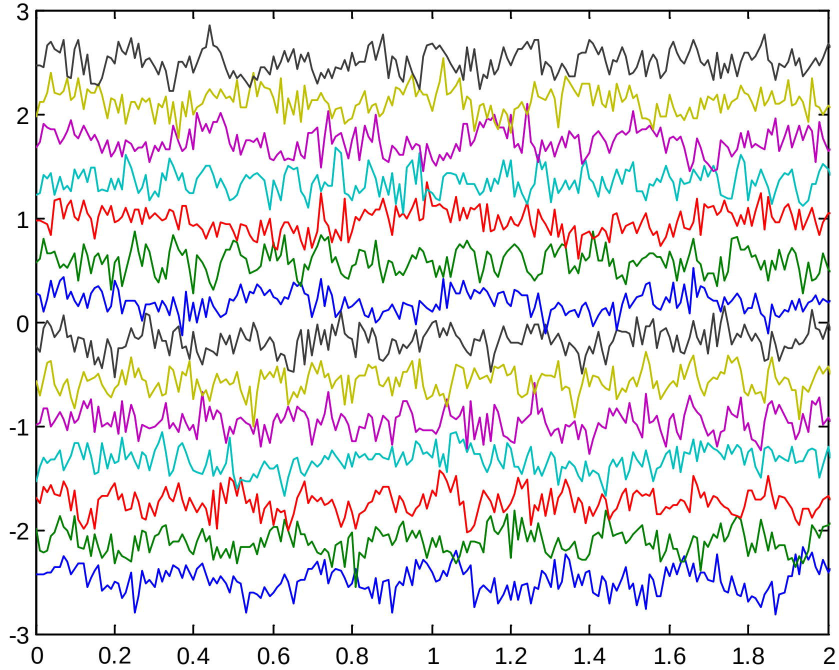
Stats: Covariance







Stats: Covariance Matrix



Note: Diag. Variance & Symmetry

$$\text{Cov} = \begin{bmatrix} \text{cov}(X_1, X_1) & \text{cov}(X_1, X_2) & \cdots & \text{cov}(X_1, X_n) \\ \text{cov}(X_2, X_1) & \text{cov}(X_2, X_2) & \cdots & \text{cov}(X_2, X_n) \\ \vdots & \vdots & \ddots & \vdots \\ \text{cov}(X_n, X_1) & \text{cov}(X_n, X_2) & \cdots & \text{cov}(X_n, X_n) \end{bmatrix}$$

EigenValues & EigenVectors

Back to Matrices

Data Matrix, EigenVector, EigenValue, EigenVector

$$\mathbf{A}\boldsymbol{v} = \lambda\boldsymbol{v} \xrightarrow{\text{rearrange}} (\mathbf{A} - \lambda\mathbf{I})\boldsymbol{v} = \mathbf{0}$$

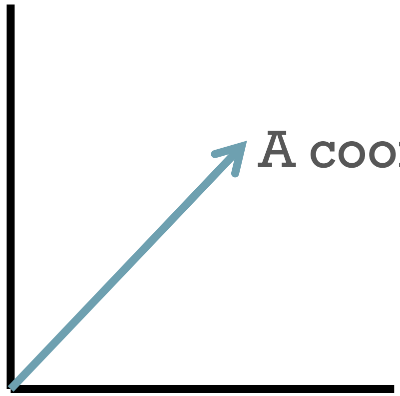
Identity Matrix

Think about:

- What is a vector?
- What is a matrix?

EigenValues & EigenVectors

- What is a vector?



A coordinate in some space (n dimensions)

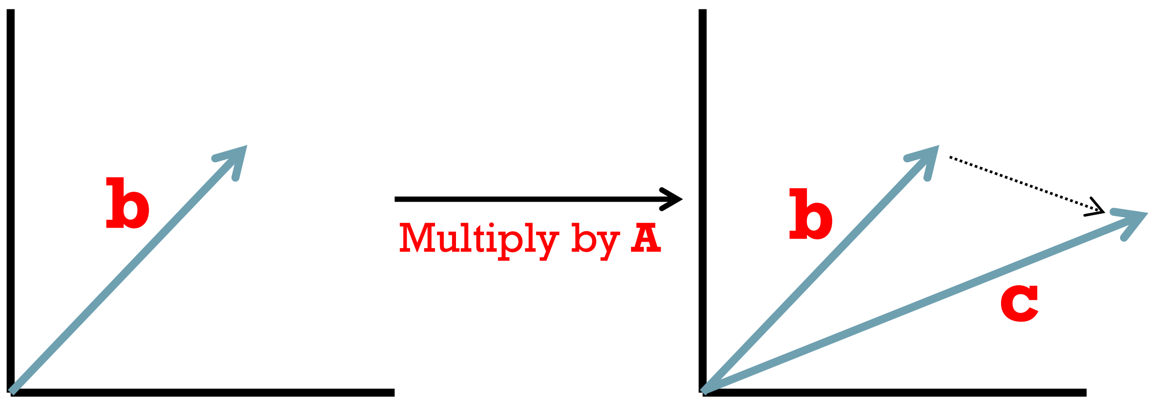
EigenValues & EigenVectors

- What is a matrix?
  - What is matrix multiplication?

A **b** **c**

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} a_{11}b_1 + a_{12}b_2 \\ a_{21}b_1 + a_{22}b_2 \end{bmatrix}$$

- Matrix **A** linearly transforms vector **b** into vector **c**

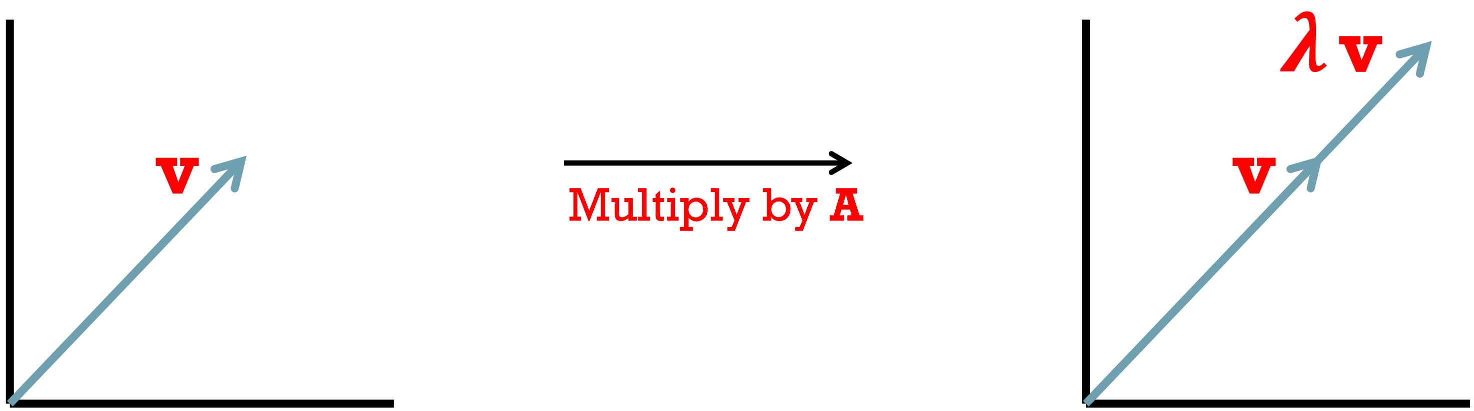


- Rotation
- Scaling

EigenValues & EigenVectors

- Back to the definition

$$\mathbf{Av} = \lambda\mathbf{v}$$



- ~~Rotation~~
- Scaling only

EigenValues & EigenVectors

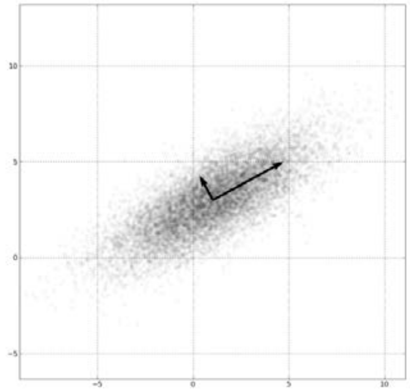
Some neat properties

- For N x N matrix, N eigenvalues & eigenvectors
- Eigenvectors are orthogonal
  - What does orthogonal/perpendicular mean?
- Sum of Eigenvalues = trace of matrix
- Largest eigenvalue: “principal eigenvalue”
  - Associated eigenvector “principal eigenvector”

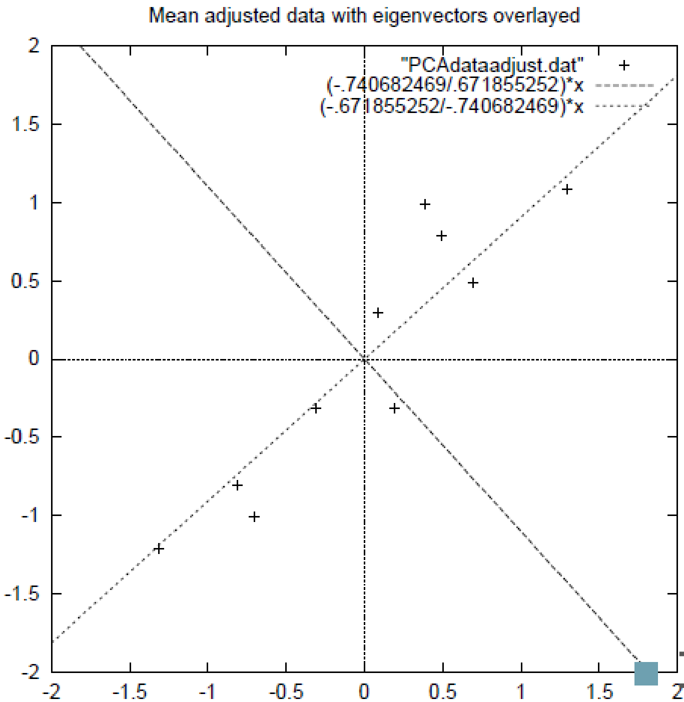
- Note: Not all matrices have eigen*

Principal Component Analysis

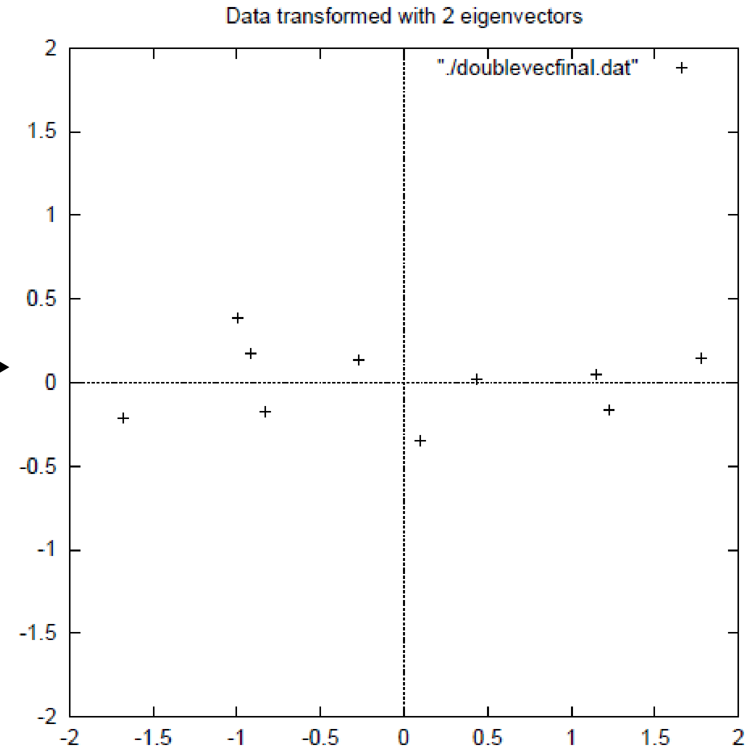
1. Collect data into matrix **D**
2. Compute **Covariance Matrix, C,** from **D**
3. Compute **Eigenvectors & Eigenvalues** of **C**
4. Rank Eigenvectors by magnitude of Eigenvalues
5. Build matrix, **P**, of ranked Eigenvectors
   - Include arbitrary number of eigenvectors
     - Choice affects # of dimensions in transformed data
6. Transform (ie. Multiply) **D** by **P**



Principal Component Analysis

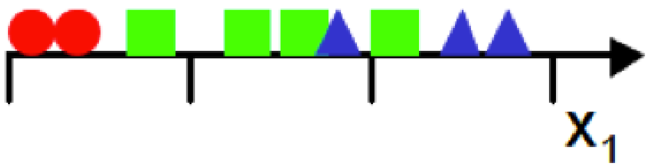


PCA

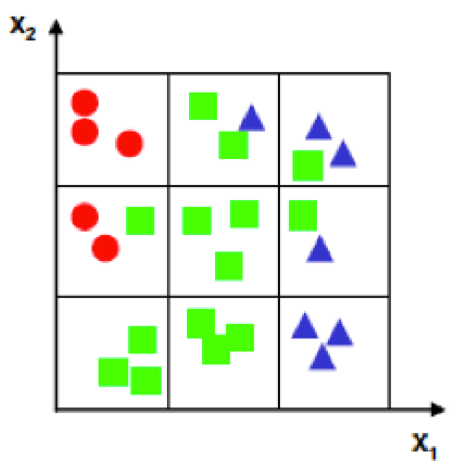


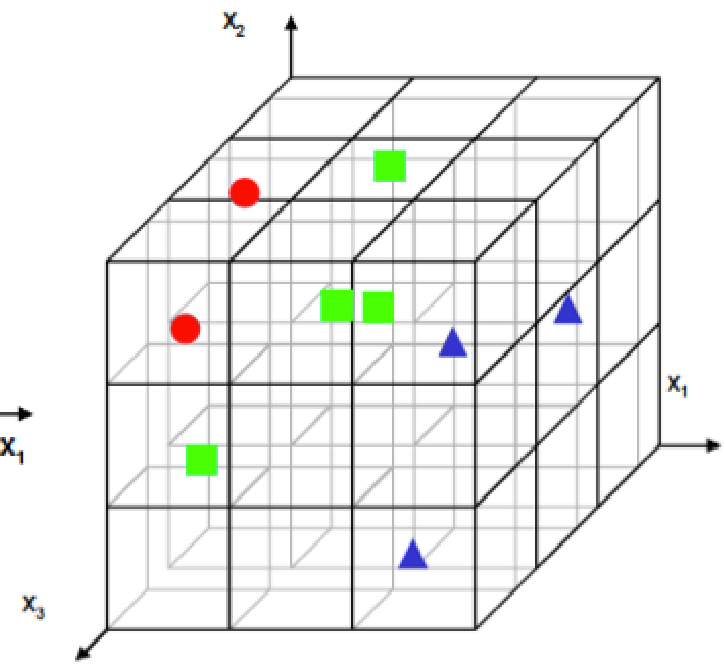
- Note: Not all matrices have eigen*

Motivation



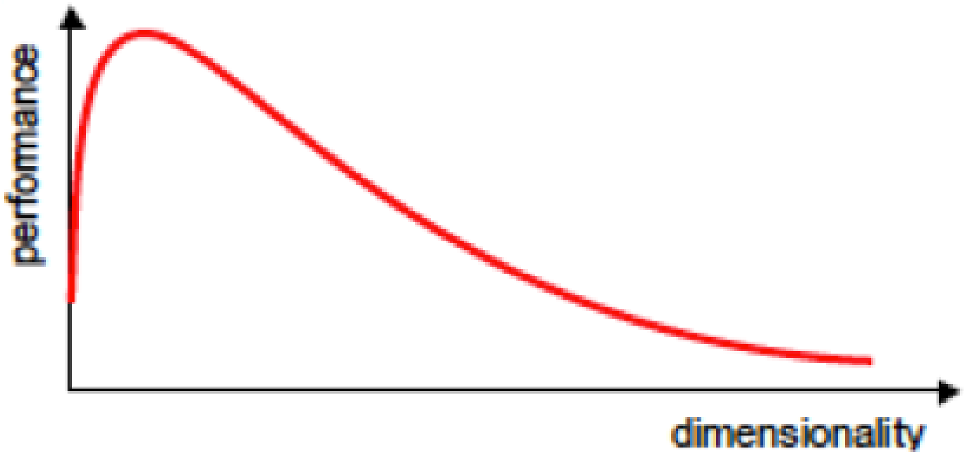
Constant density





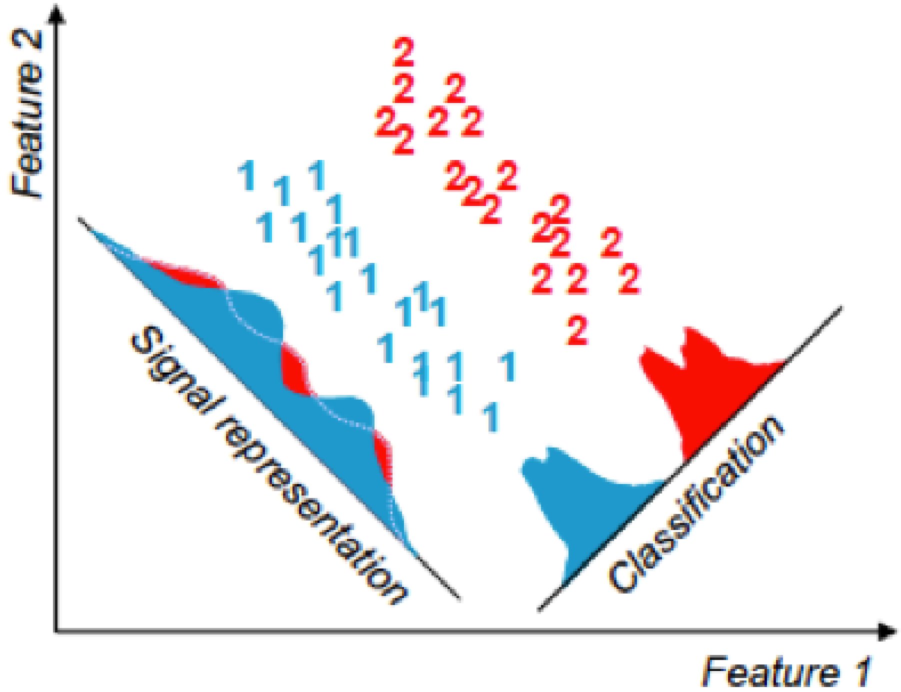
Motivation

- *Curse of Dimensionality*
- "For a given sample size, there is a maximum number of features above which the performance of out classifier will degrade rather than improve."



PCA v.s. LDA

- PCA uses a signal *representation* criterion
- LDA uses a signal *classification* criterion



Linear Discriminant Analysis

- *Fish-sorting*
- Objective: To perform dimensionality reduction while preserving as much of the class discriminatory information as possible.

Principal Component Analysis

- Objective: To perform dimensionality reduction while preserving as much of the variance in the high-dimensional space as possible

Canonical Component Analysis

- Two datasets X and Y, to find basis vectors such that correlation between the projection of the variables onto the basis vectors are mutually maximized

Independent Component Analysis

- *Cocktail party problem*
- How to solve for a_{ij}?

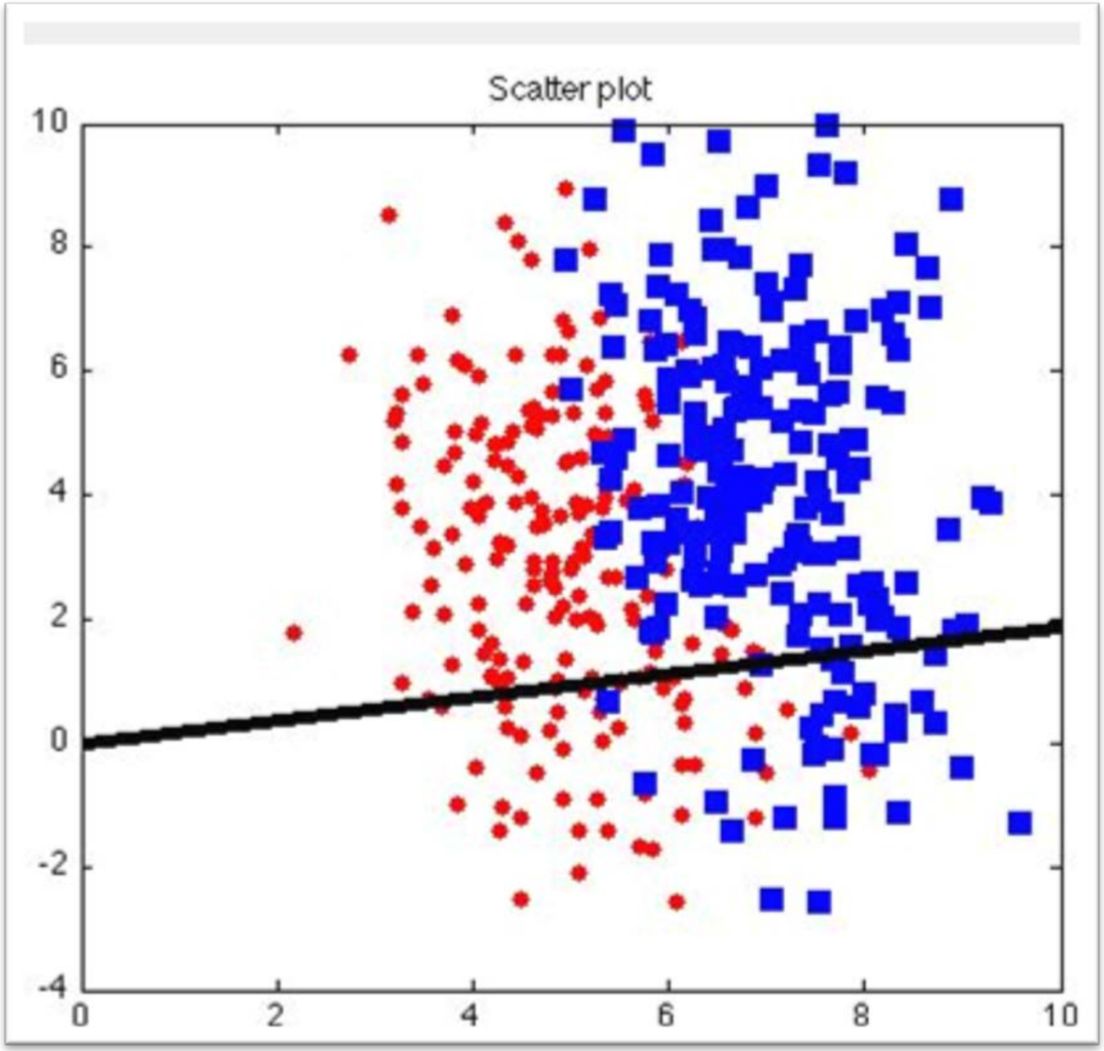
$$Sam(t) = a_{11}mic_1 + a_{12}mic_2$$
$$Alex(t) = a_{21}mic_1 + a_{22}mic_2$$

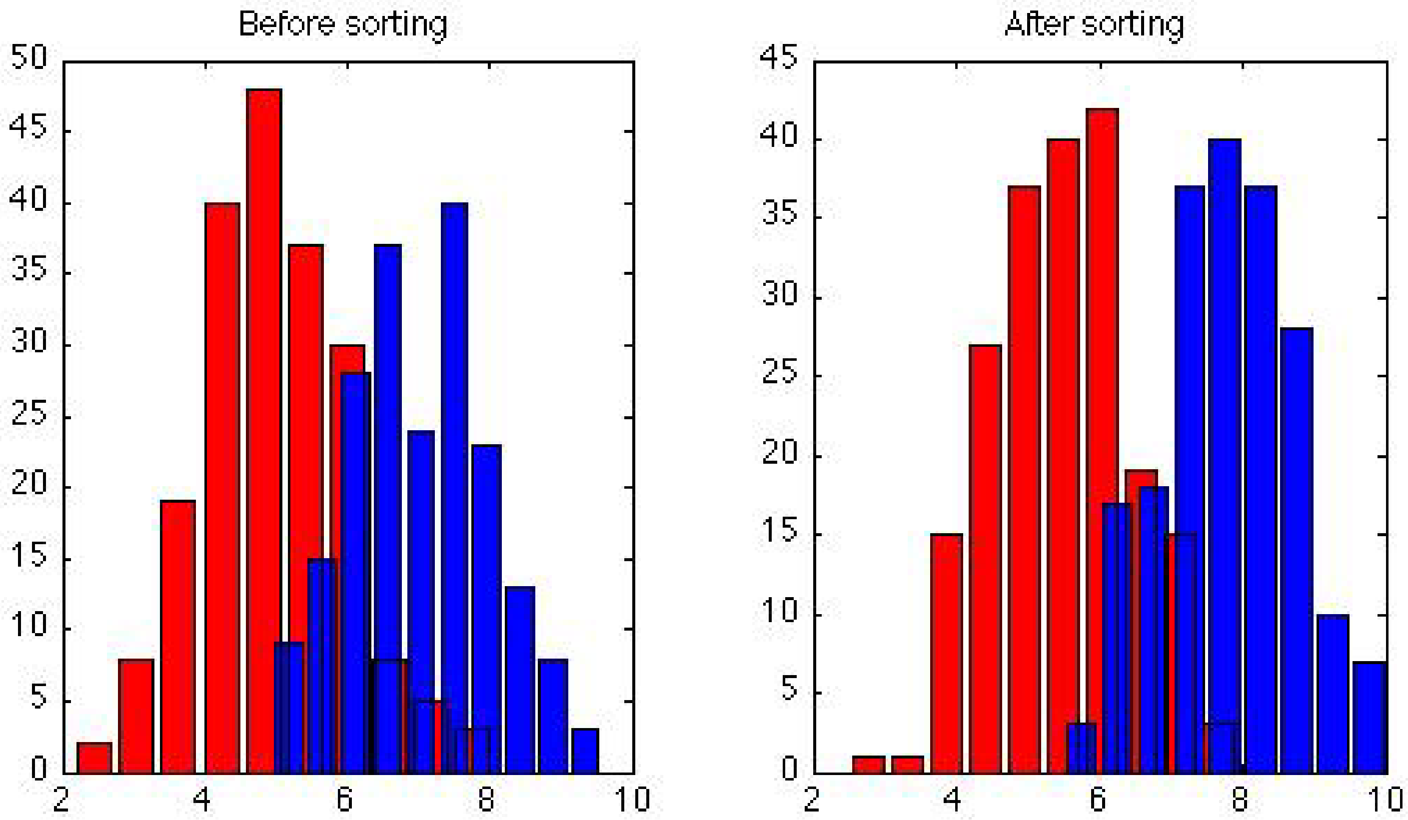
LDA in-depth

Demo Code

```
clear
clc
close all
% Simple example
gauss(1).mu = [5,3]'; gauss(1).cov=[1 -1; -1 7];
gauss(1).data=mvnrnd(gauss(1).mu,gauss(1).cov,200);

gauss(2).mu = [7,4]'; gauss(2).cov=[1 -1; -1 7];
gauss(2).data=mvnrnd(gauss(2).mu,gauss(2).cov,200);

x1=  gauss(1).data;
x2=  gauss(2).data;
clab = [ones(200,1),ones(200,1).*2];
x = [x1;x2]; % Full data

% Plot the data
```

+


Scatter plot
10
8
6
4
2
0
-2
-4
0
2
4
6
8
10


Before sorting
50
45
40
35
30
25
20
15
10
5
0
2
4
6
8
10
After sorting
45
40
35
30
25
20
15
10
5
0
2
4
6
8
10