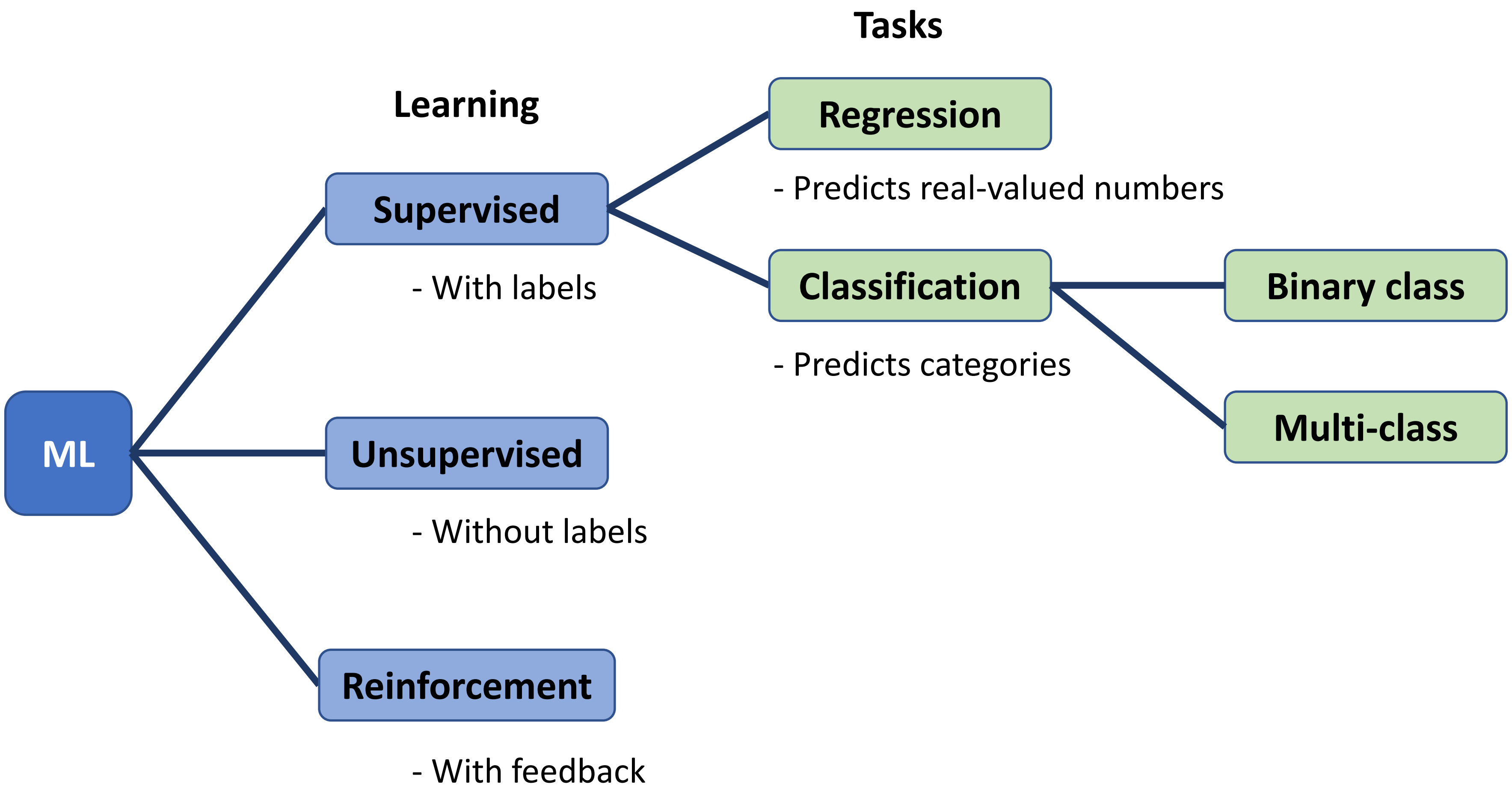
University of Colorado **Boulder**

Unsupervised Learning

Geena Kim

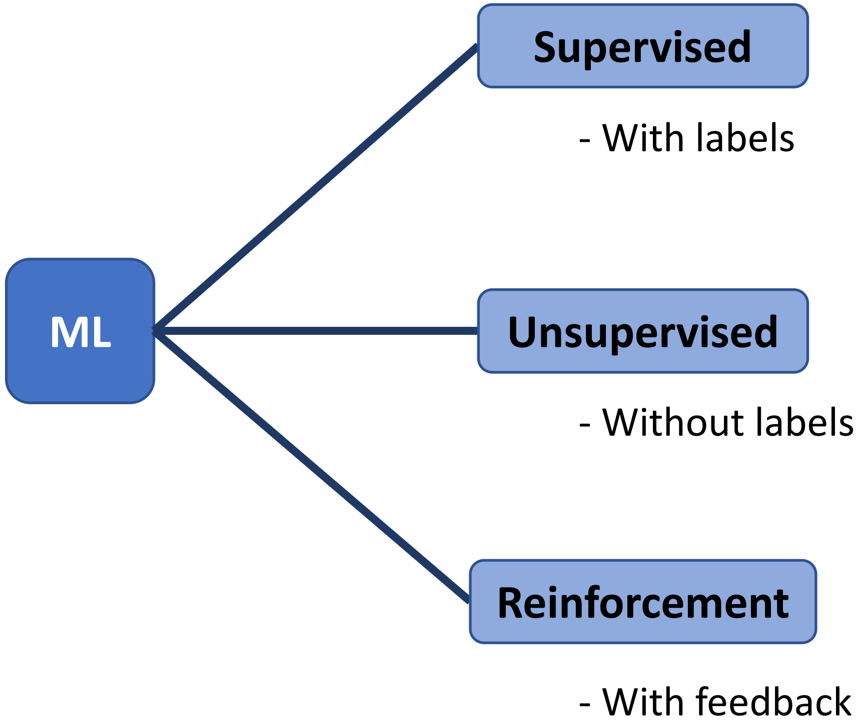
Introduction

Types of machine learning problems



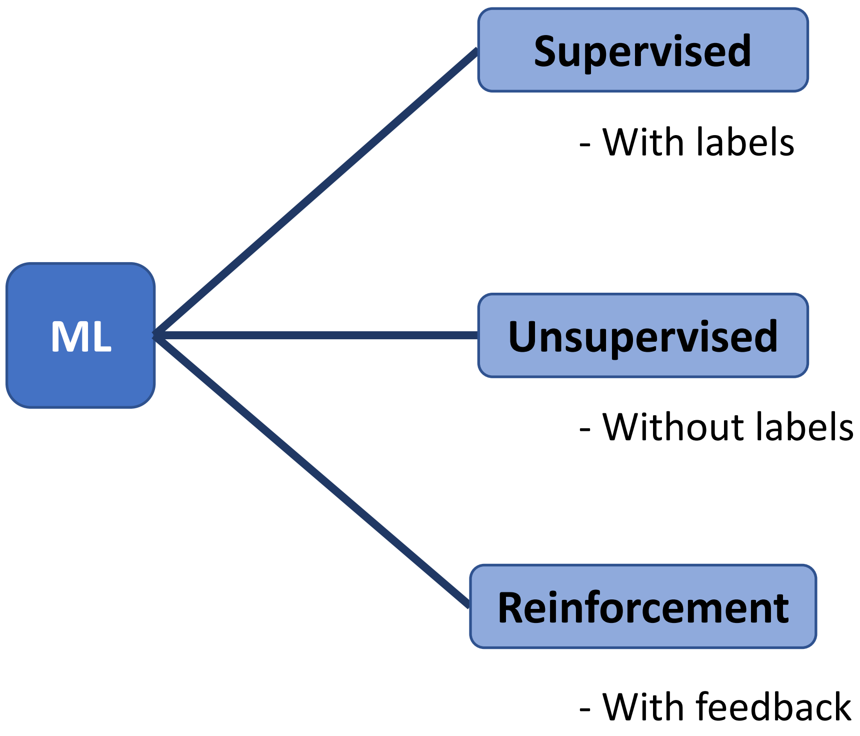
Why Unsupervised Learning

Learning



Why Unsupervised Learning

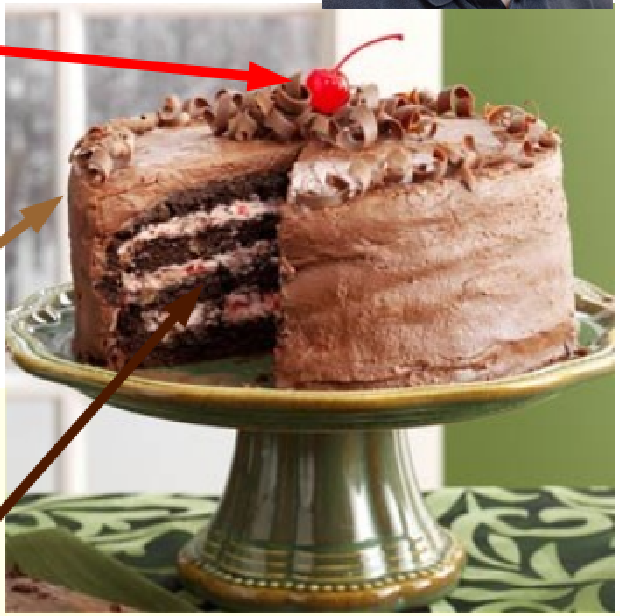
Learning



Yann LeCun says about Unsupervised Learning…

in terms of data availability

- "Pure" Reinforcement Learning (cherry)
  - The machine predicts a scalar reward given once in a while.
  - **A few bits for some samples**
- Supervised Learning (icing)
  - The machine predicts a category or a few numbers for each input
  - Predicting human-supplied data
  - **10→10,000 bits per sample**
- Unsupervised/Predictive Learning (cake)
  - The machine predicts any part of its input for any observed part.
  - Predicts future frames in videos
  - **Millions of bits per sample**



- (Yes, I know, this picture is slightly offensive to RL folks. But I'll make it up)

Goals of Unsupervised Learning

Not interested in prediction but to discover interesting things about the data

Informative visualization

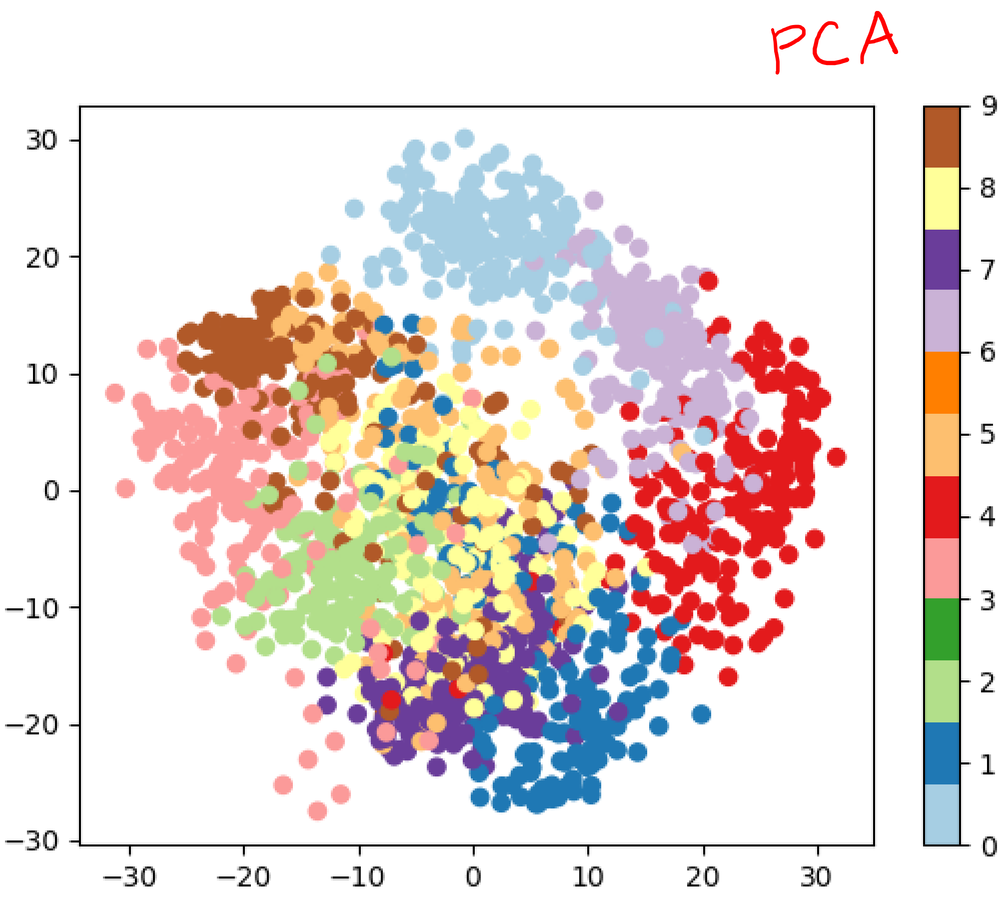
Finding subgroups Clustering

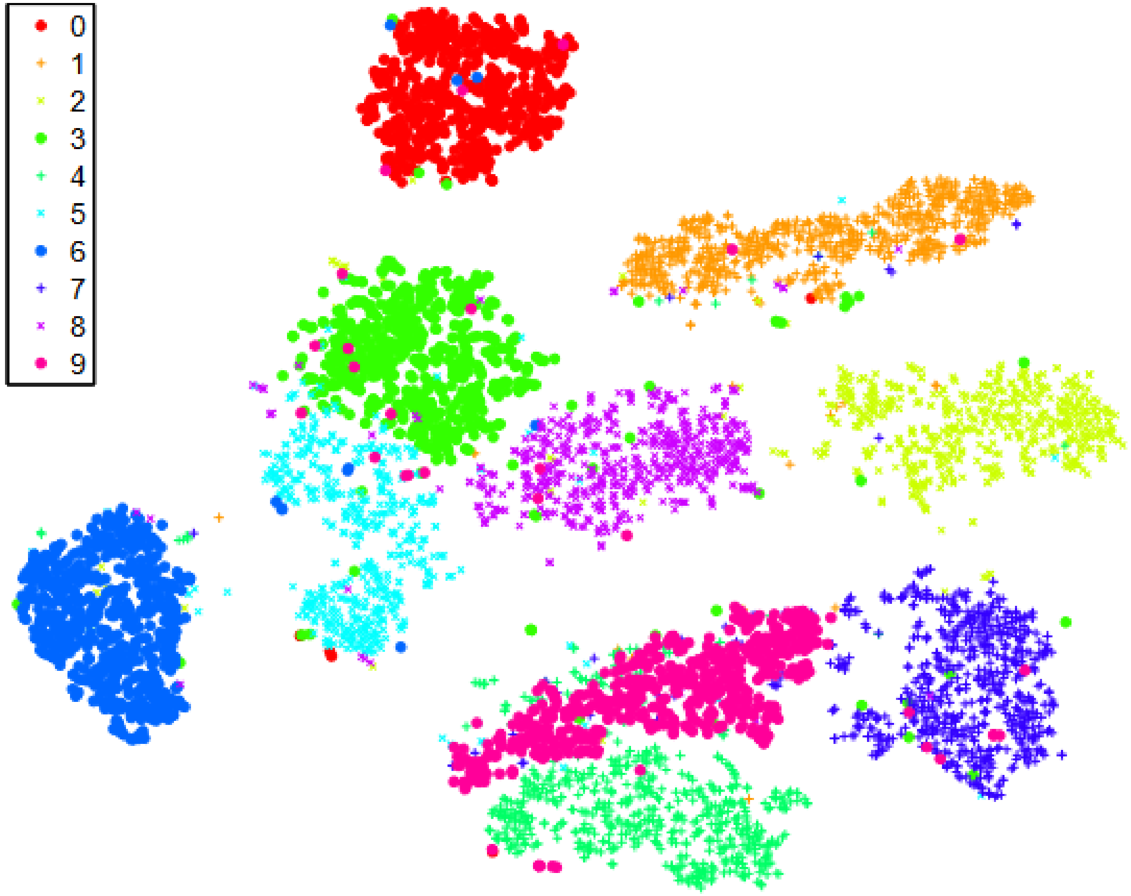
Dimensionality Reduction

Preprocessing

Data synthesis

Visualization by unsupervised learning



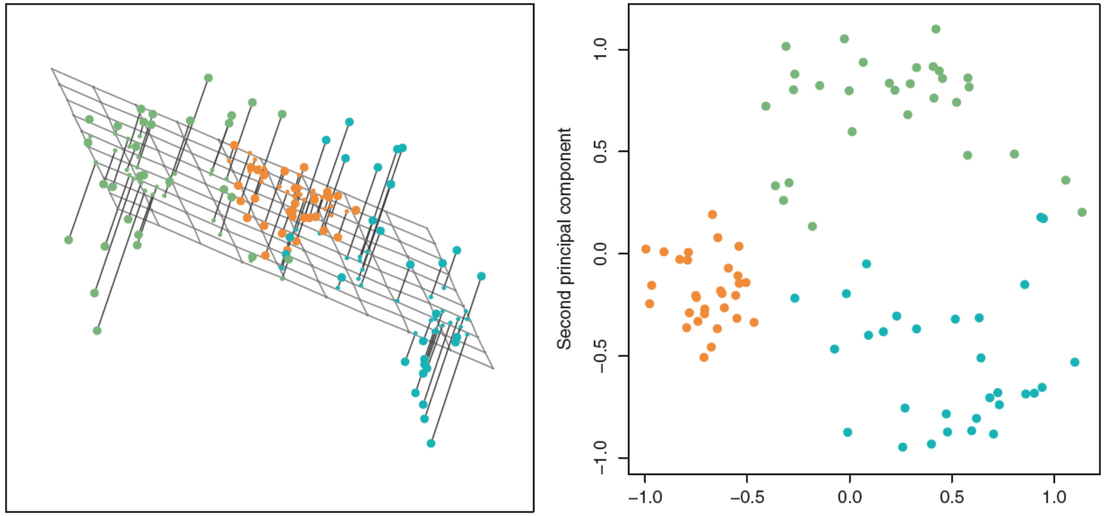


(a) Visualization by t-SNE.

Image credit: scipy.org and L van der Maaten et al (2008)

Dimensionality Reduction

Projection to low-dimension



Manifold learning

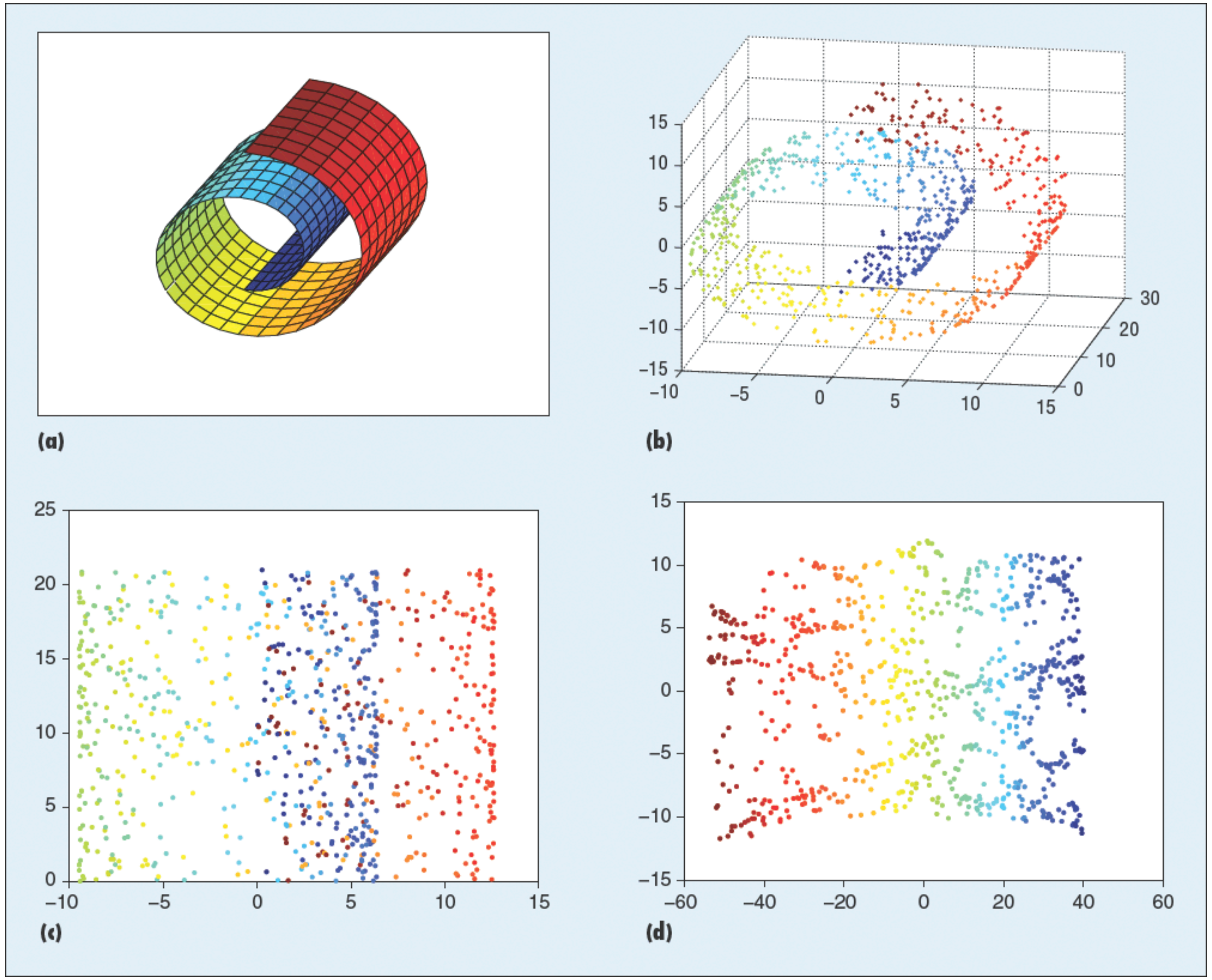


Image credit: ISLR textbook and Zhang et al (2010)

Clustering

- Marketing and sales
- Social network analysis
- Genomics, Oncology

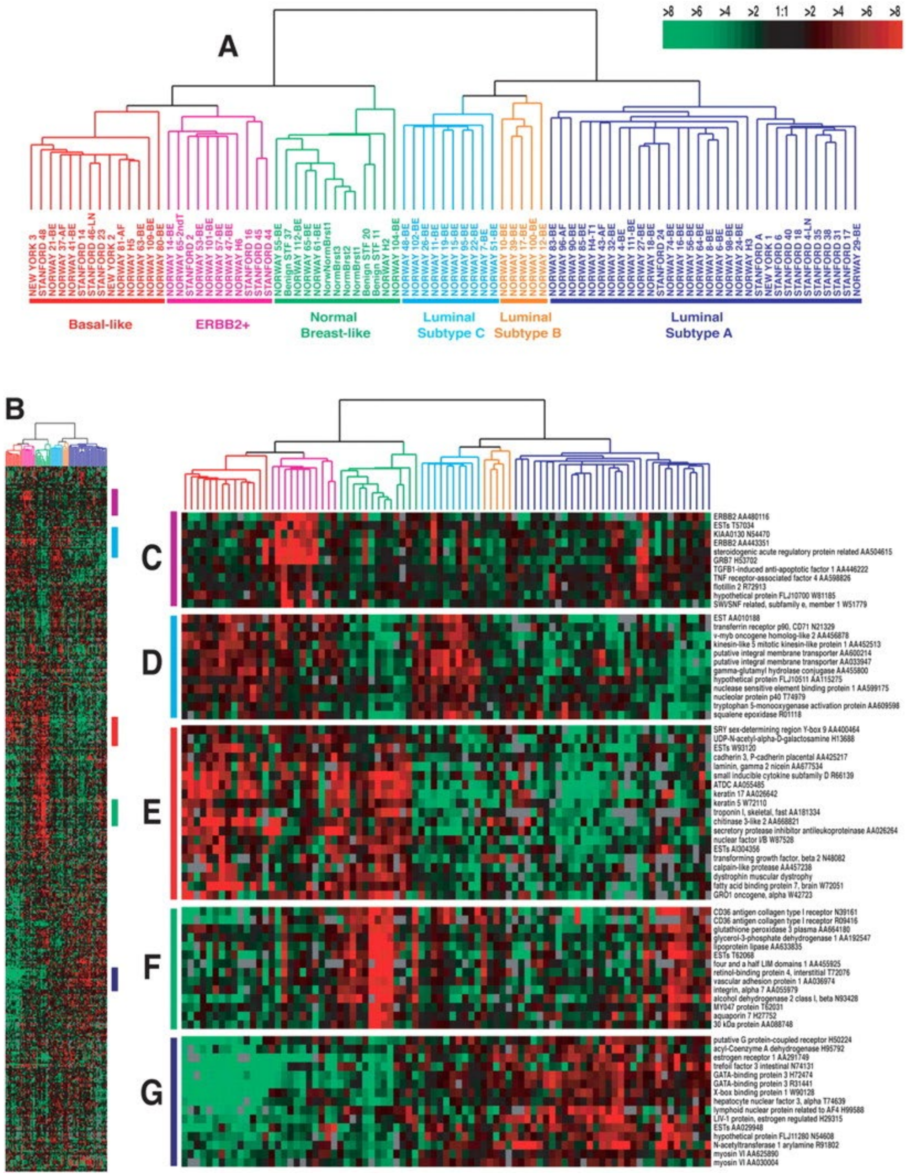
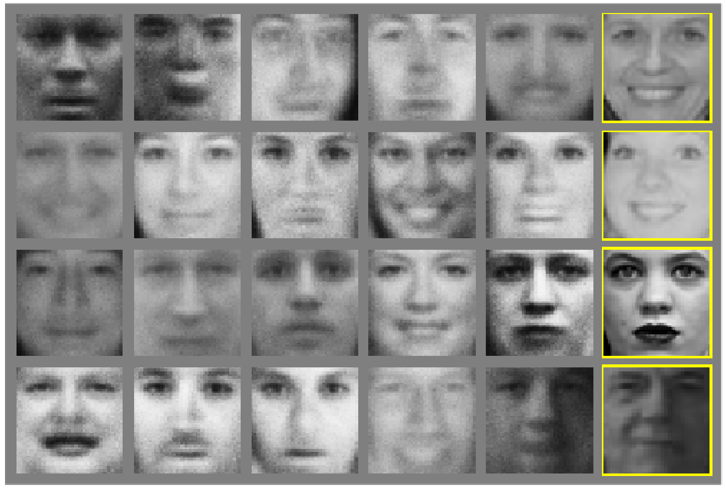


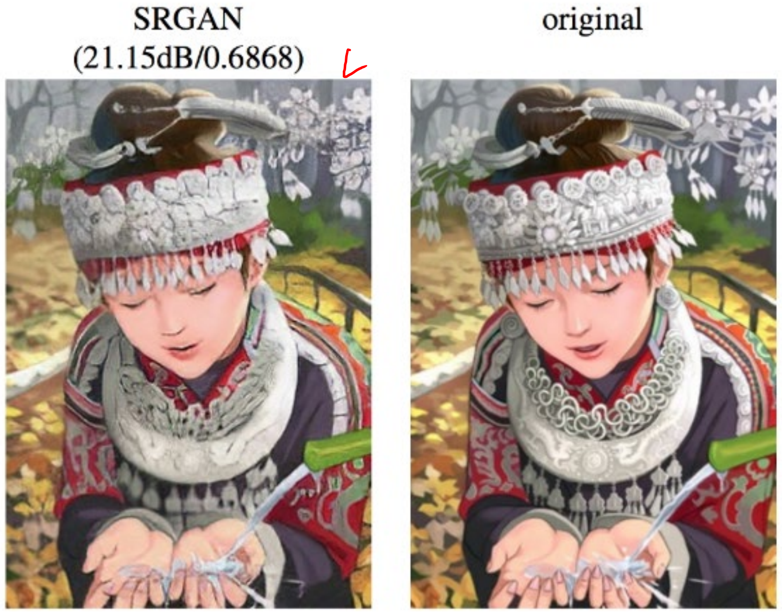
Image credit: T. Sørlie et al (2001)

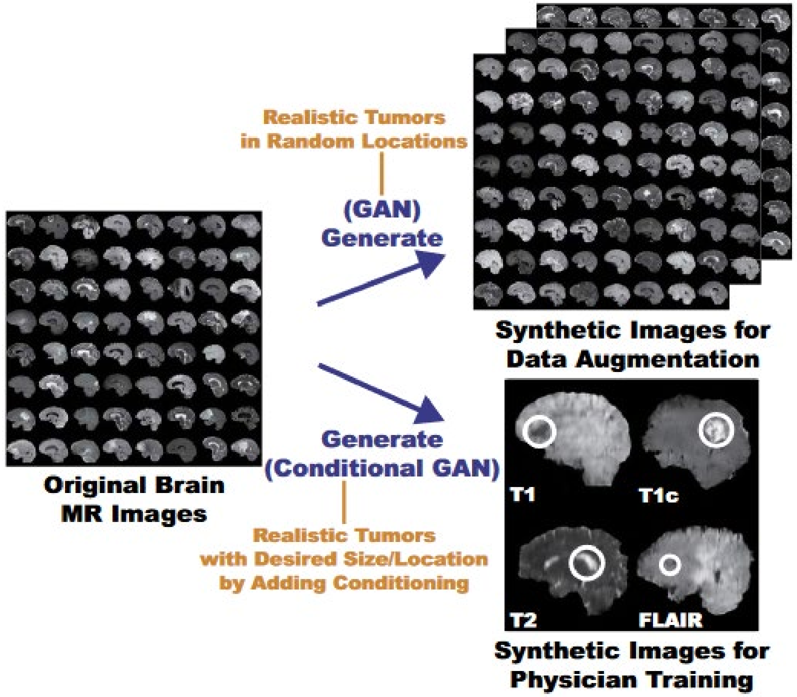
Applications: Recommender System

- Similarity based
- Learning latent features/Matrix Factorization
- Collaborative Filtering using Graph

Data Generation







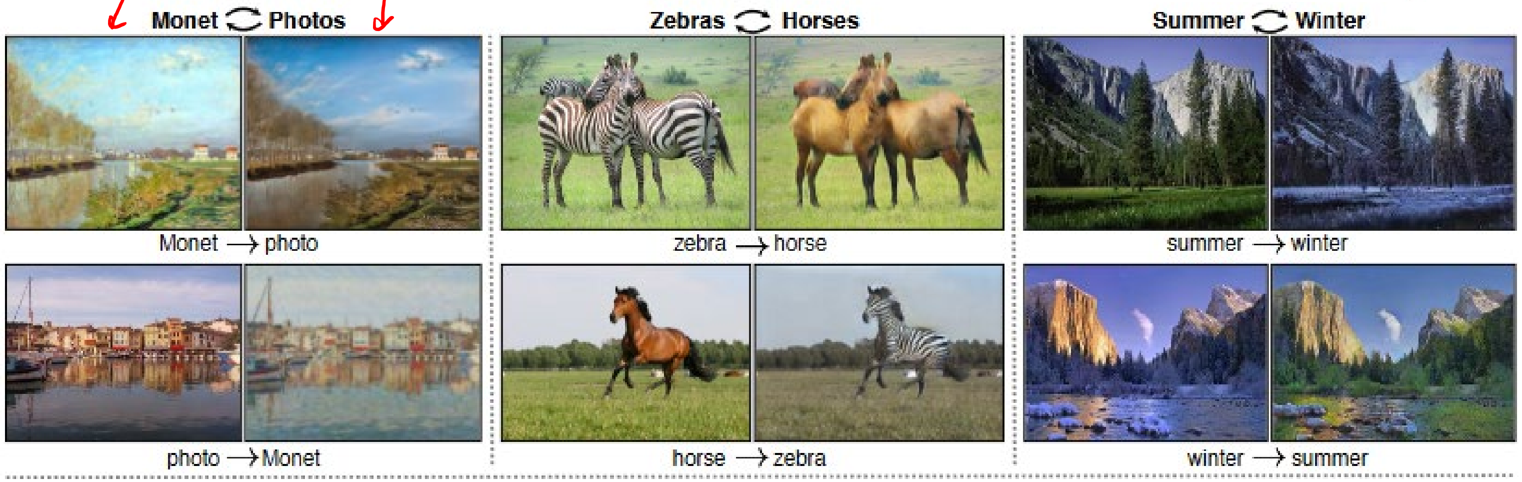
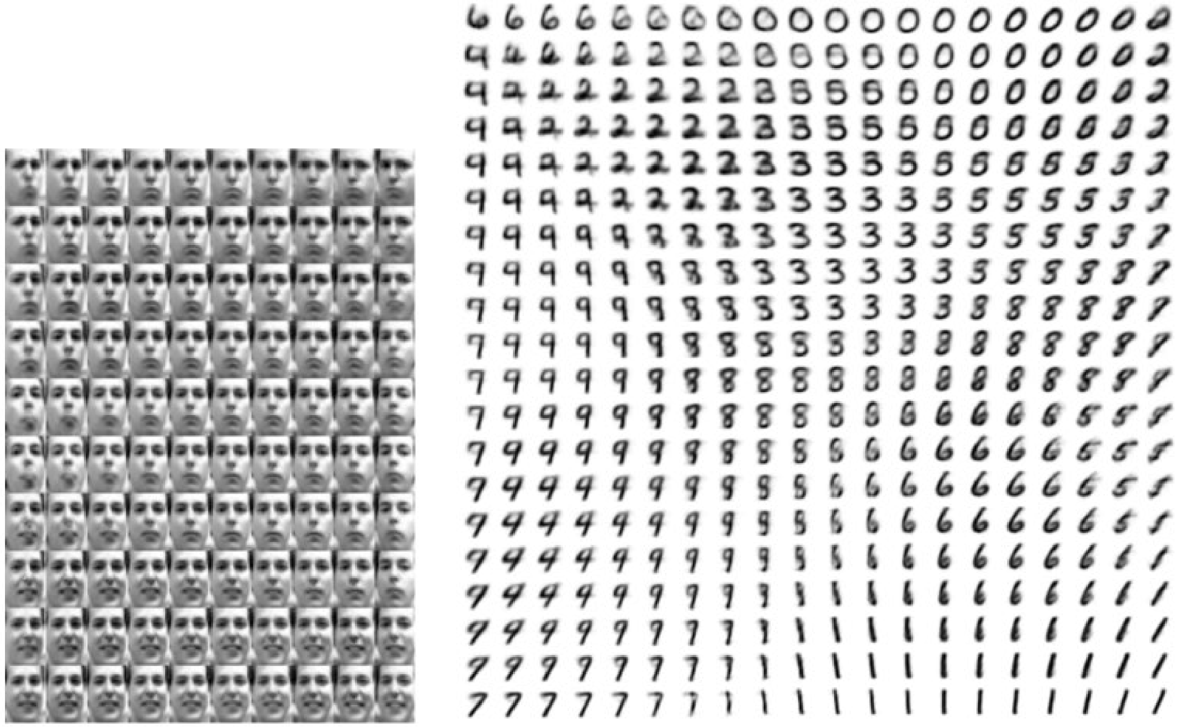


Image credit: Goodfellow et al(2014),
Ledig et al (2016),
Zhu et al (2017),
Han et al (2018)

Self-supervision

- A generative model to reconstruct inputs (Autoencoders)
- Surrogate tasks in vision tasks
- Using clustering for graphs
- Pretraining for NLP tasks (GPT)



(a) Learned Frey Face manifold (b) Learned MNIST manifold

Image credit: Kingma & Welling (2013)

Summary

- Unsupervised Learning
- Usage:
  - Dimensionality reduction, pre-training, visualization
  - Clustering (marketing, medicine, etc)
  - Data generation
  - Industrial applications such as Recommender systems

Principal Component Analysis

Dimensionality Reduction

Curse of dimensionality

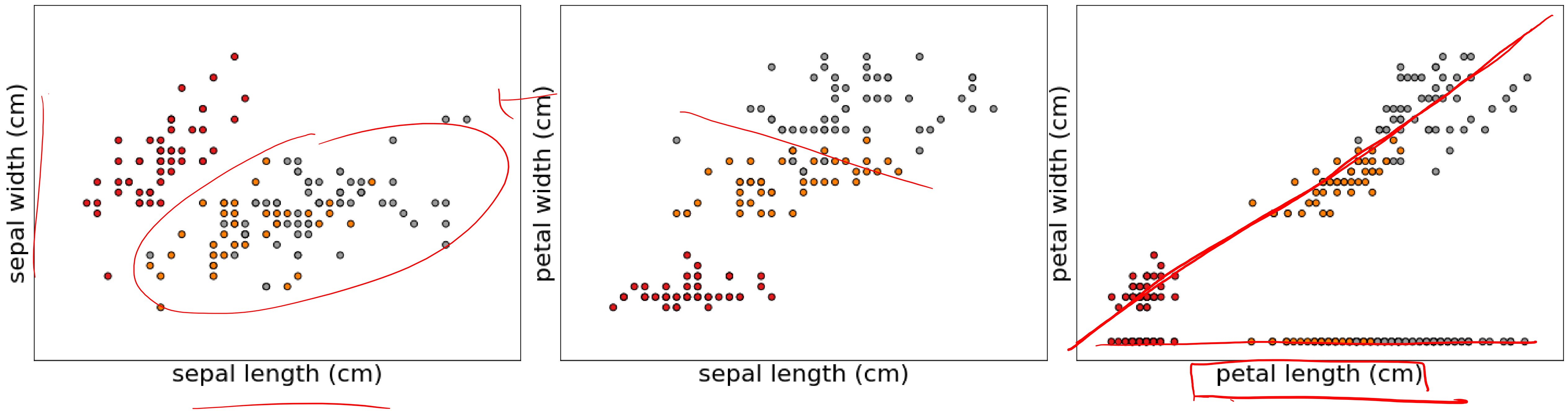
Data become sparse

Features in high dimension tend to be redundant (and correlated)

Likely to overfit

Principal Component Analysis (PCA)

PCA is a popular dimensionality reduction technique



Principal Component Analysis (PCA)

Principal Component Analysis (PCA)

PCA is a popular dimensionality reduction technique

Principal components

$$Z_1 = \phi_{11} X_1 + \phi_{21} X_2 + \ldots + \phi_{p1} X_p$$

Normalized loading vectors

$$\sum_{j=1}^{p} \phi_{j1}^2 = 1$$

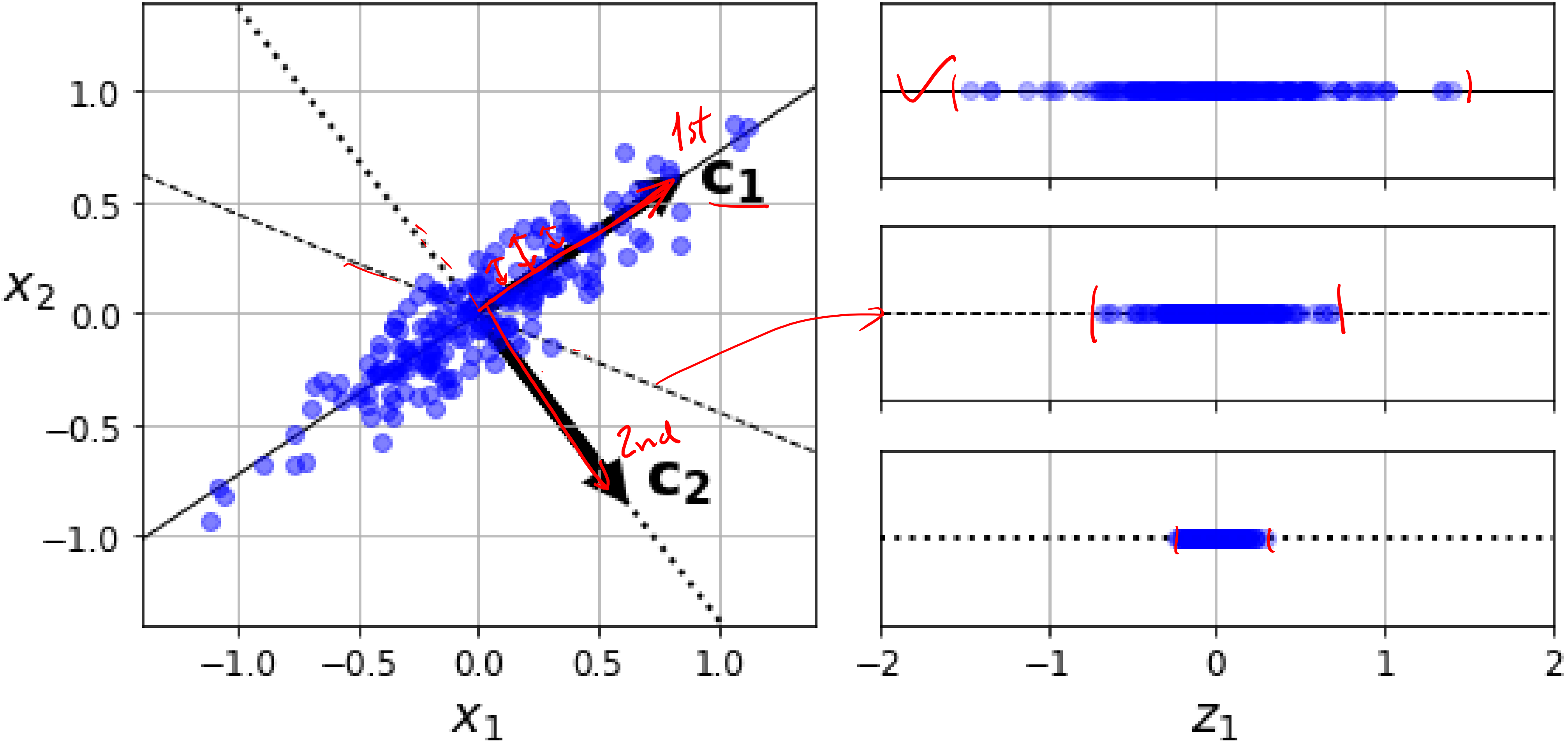
Principal Component Analysis (PCA)

How to choose the principal components?

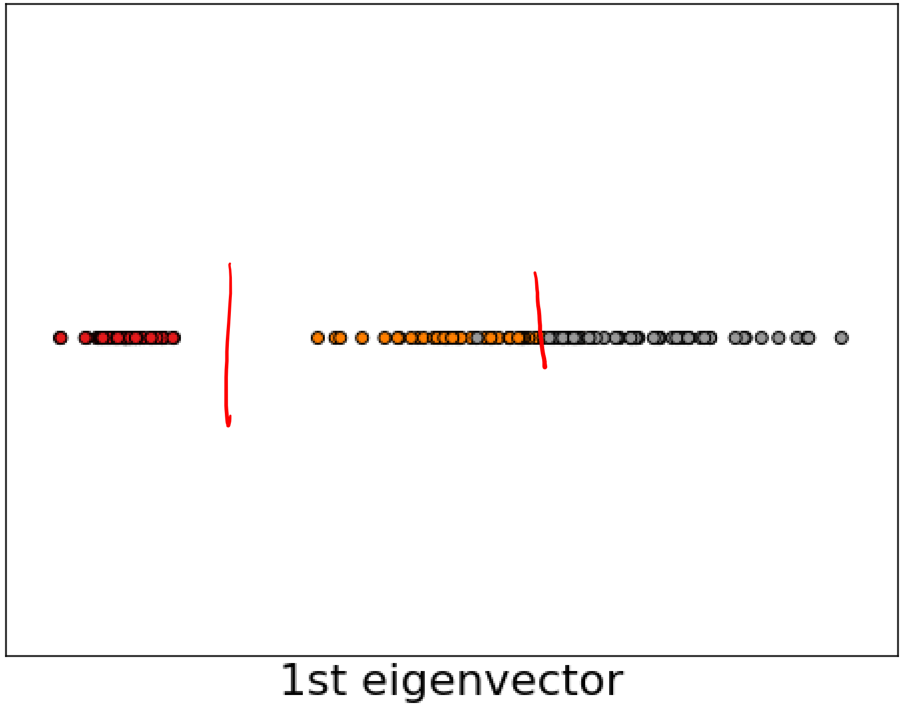
Method 1. Preserve the maximum variance

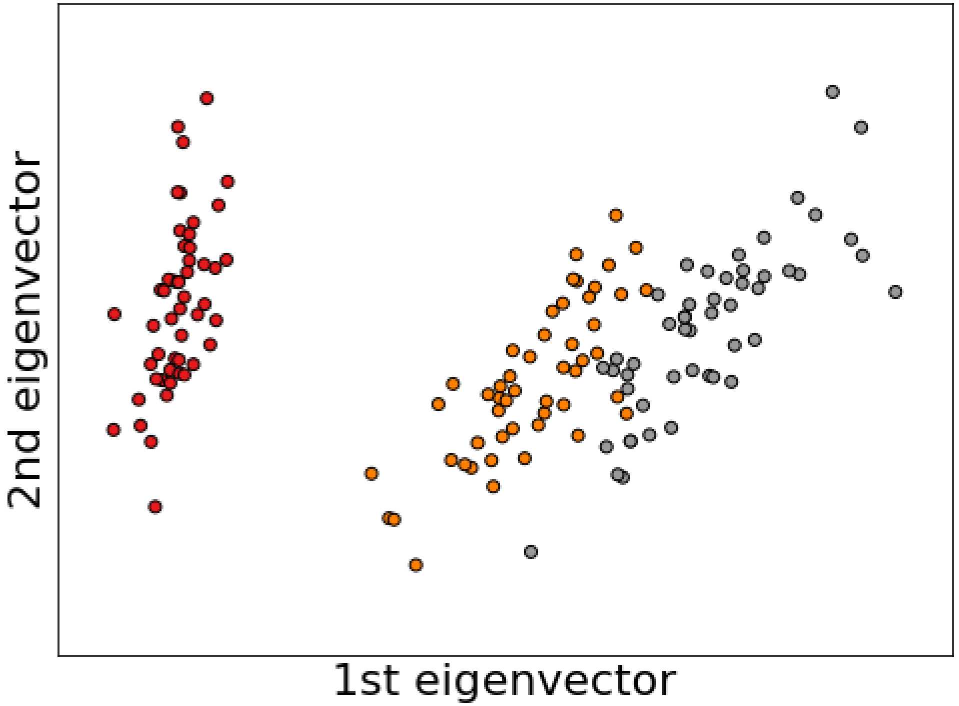
Method 2. Choose axis that minimizes the mean squared distance between the original dataset and its projection onto the axis

Principal Component Analysis (PCA)



Principal Component Analysis (PCA)

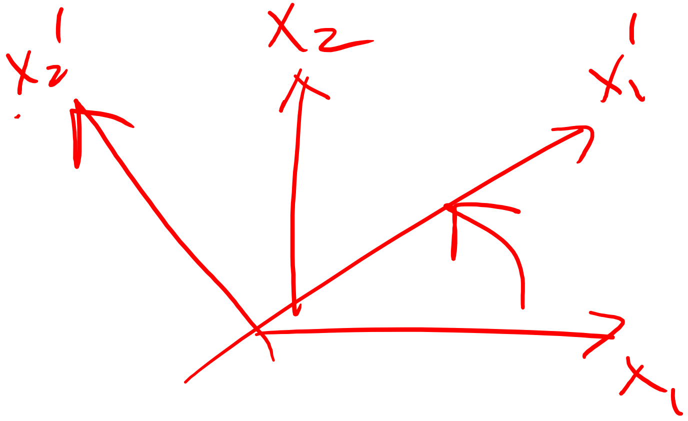




Principal Component Analysis (PCA)

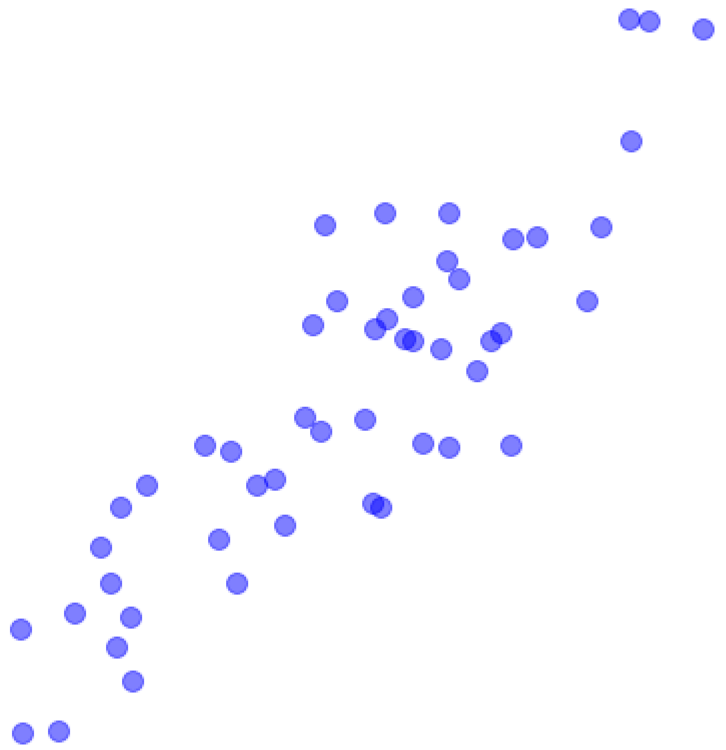
The best vector to project onto is called the **1st principal component**.
What properties should it have?

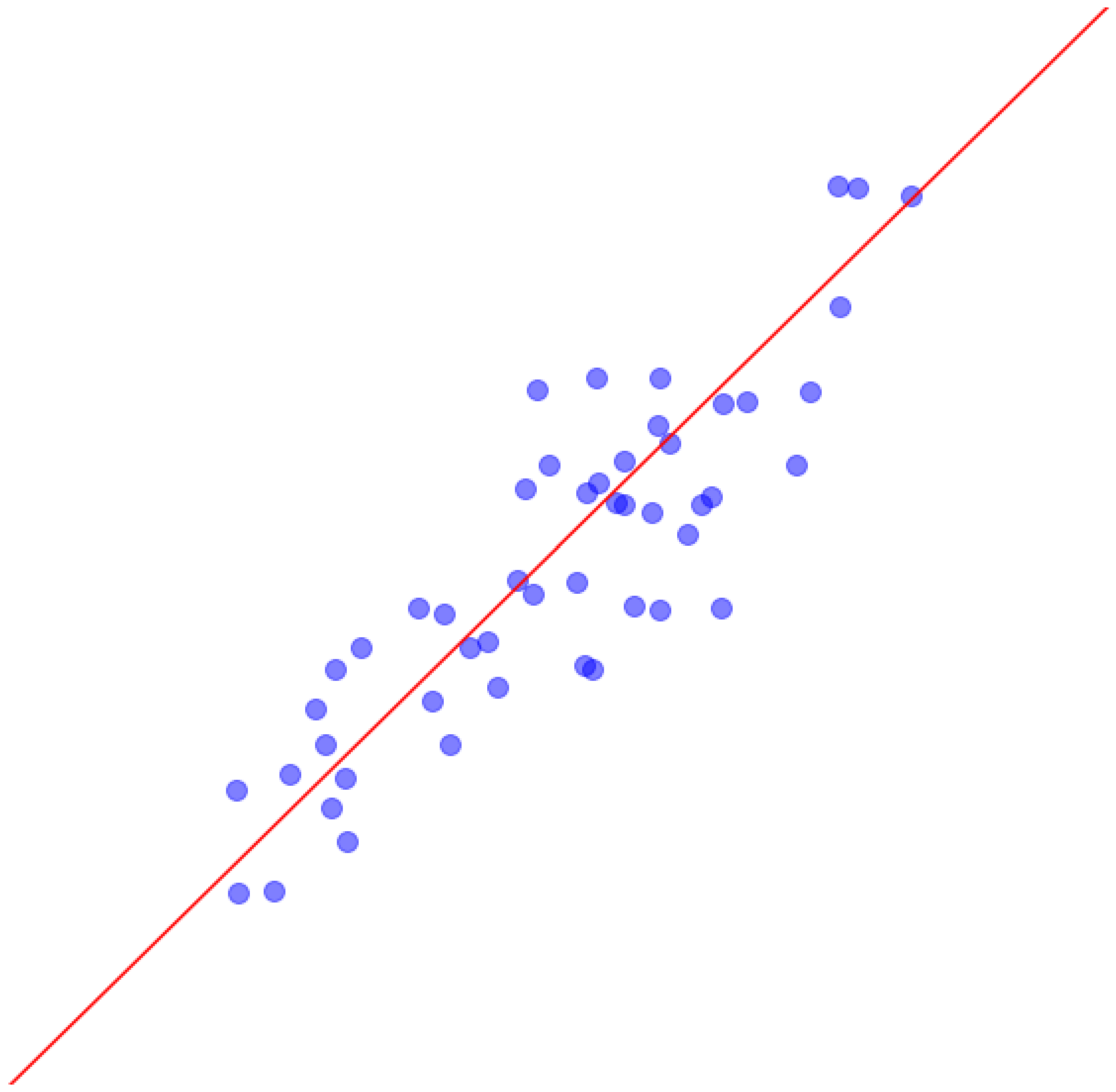
- Should capture largest variance in data
- Should probably be a unit vector

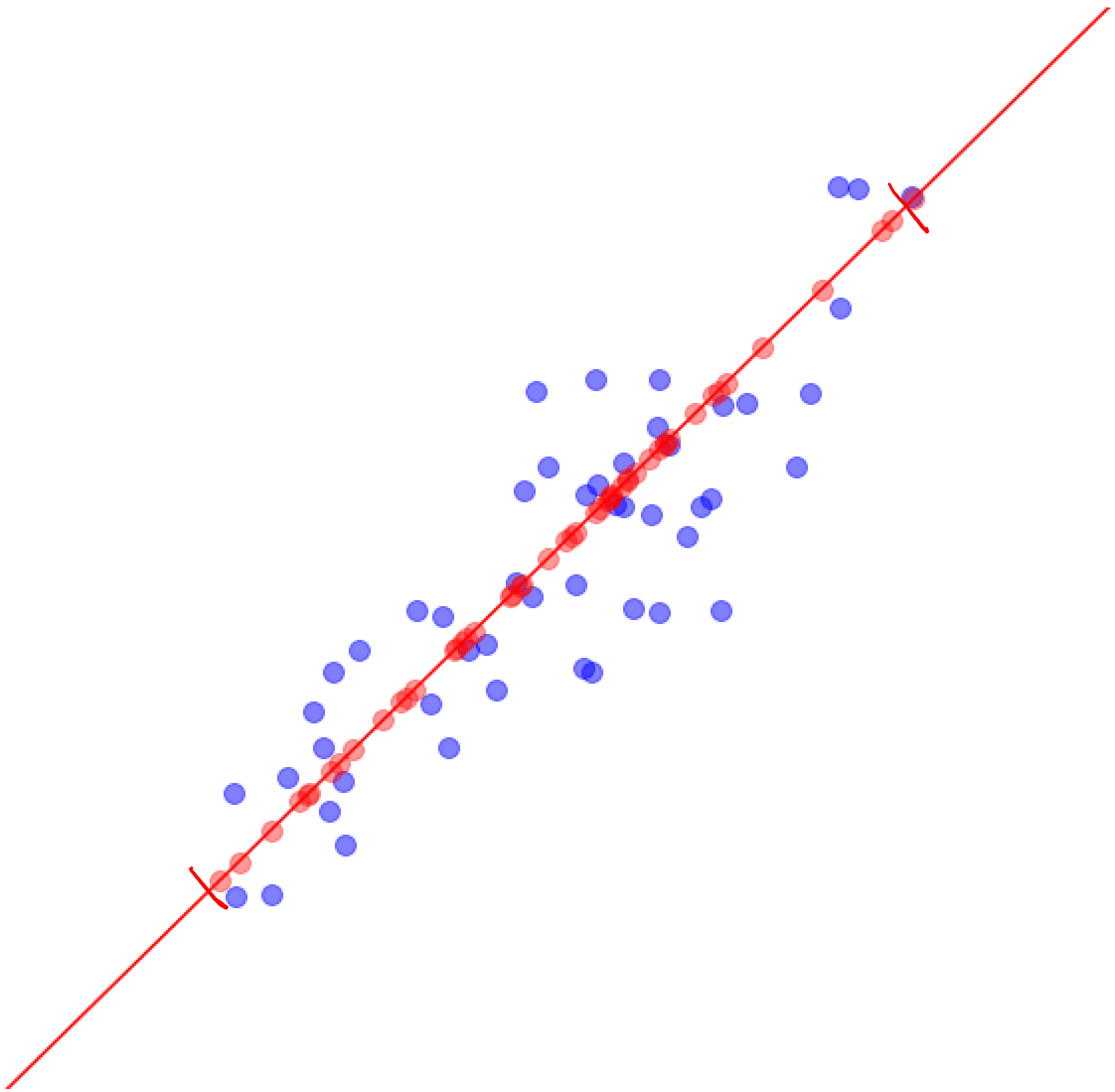


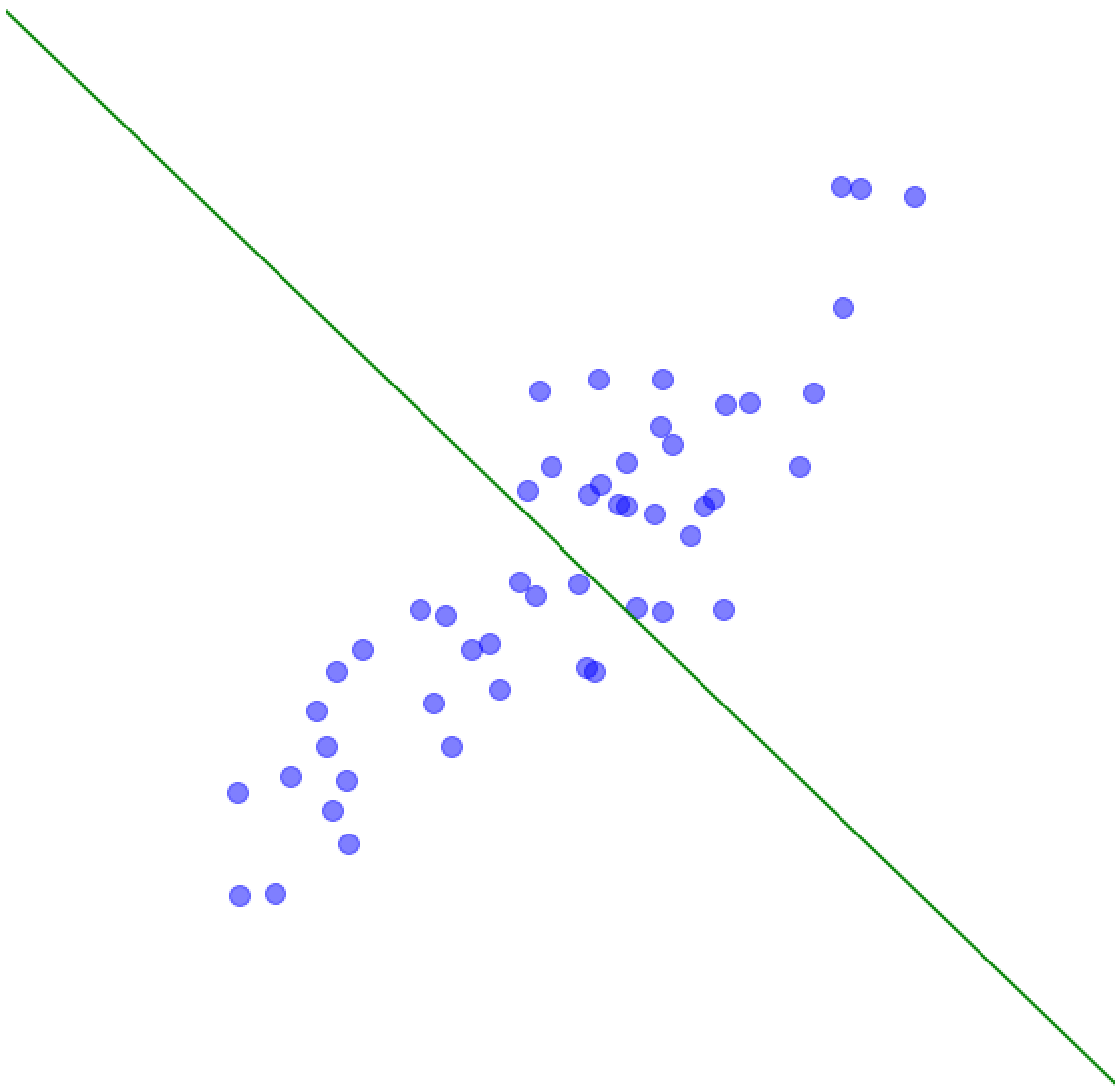
After we've found the first, look the second which:

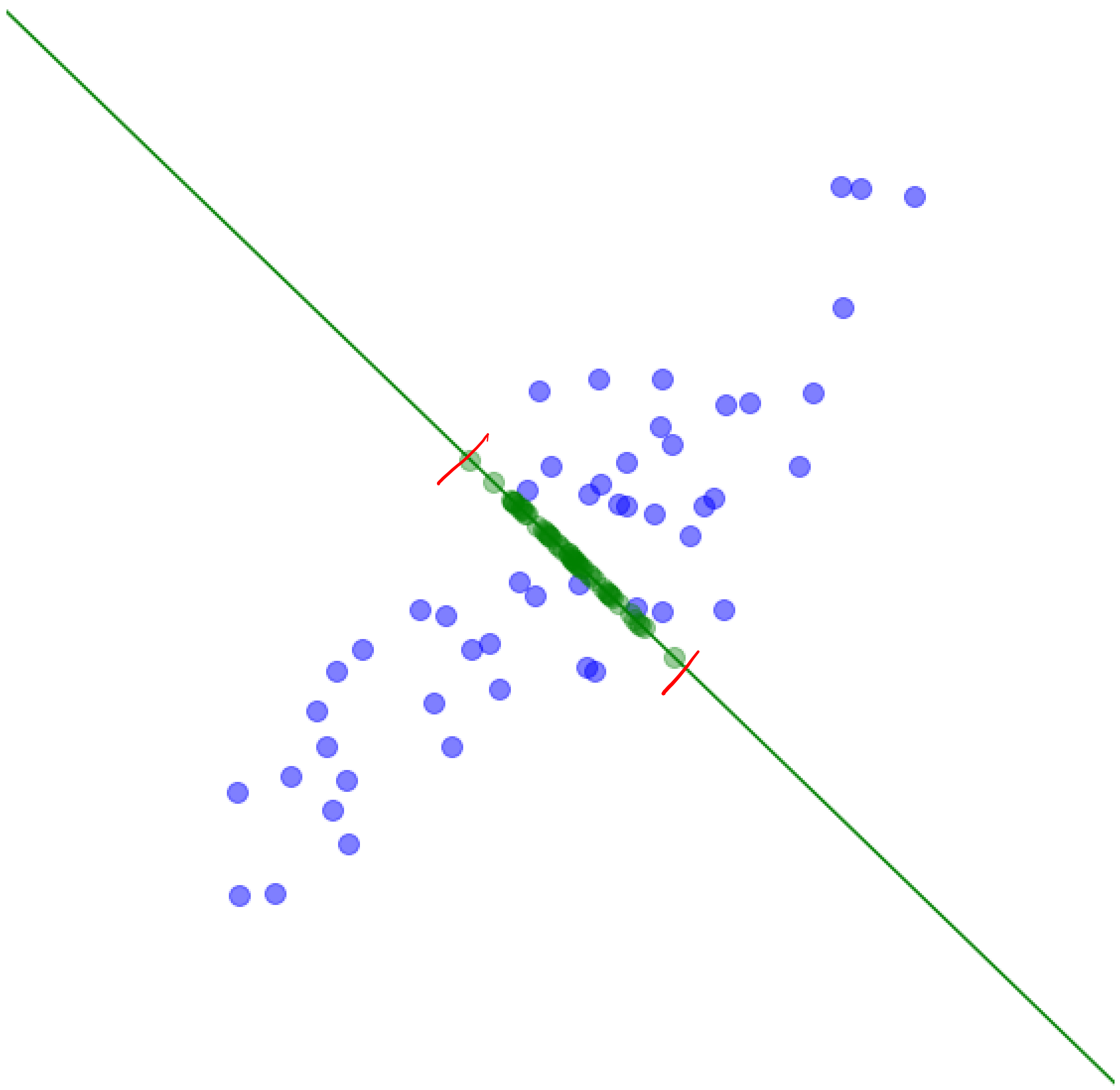
- Captures largest amount of leftover variance
- Should probably be a unit vector
- Should be orthogonal to the one that came before it











How to find principal components

Define covariance matrix

$$C = \boxed{\frac{1}{N-1} X^T X}$$

$x_1 \quad x_2 \quad \cdots \quad x_p$

$x_1 \quad Var(x_1 x_1) \quad Cov(x_1, x_2) \quad \cdots$

$Cov(x_2, x_1)$

X has zero mean

Eigenvectors of the covariance matrix are the principal components

$$Av = \lambda v$$

$$\frac{1}{N-1} X^T X = v \Lambda v^T$$

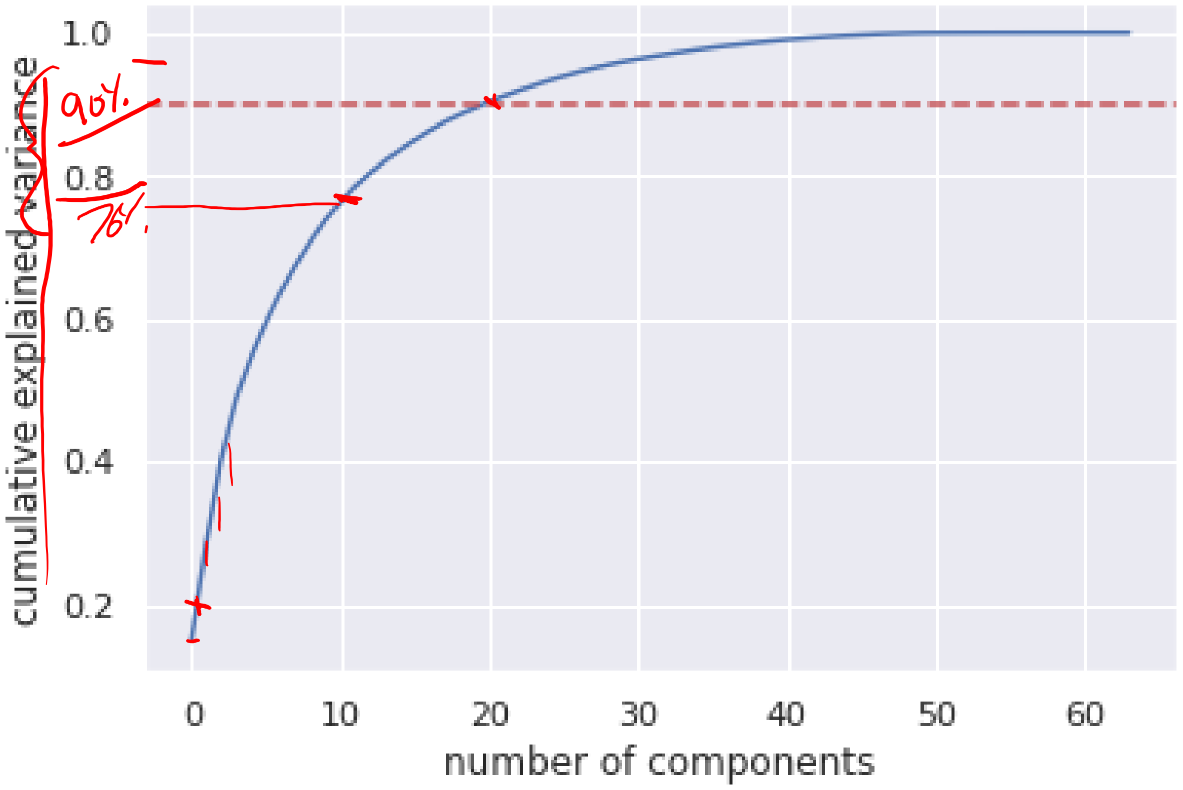
$$C = v \Lambda v^T$$

v_1

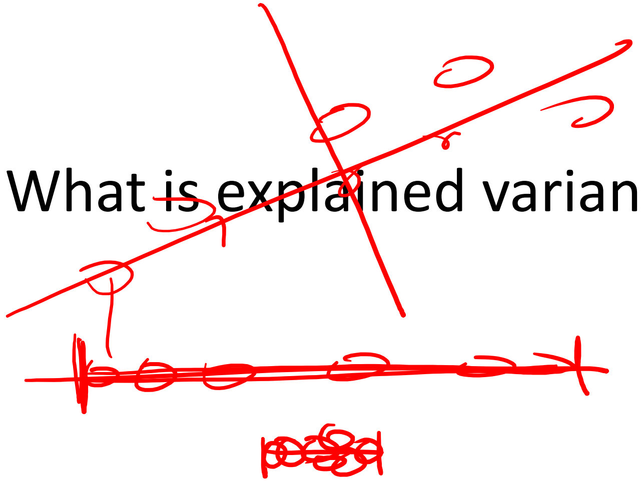
$$\begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_p \end{pmatrix}$$

Explained Variance Ratio

How many dimensions should we choose to use?



What is explained variance?



1st → Var1

2nd → Var2

What is explained variance ratio?

$$\frac{Var(m)}{tot\ Var}$$

PCA in sklearn

sklearn.decomposition.PCA

```
pca =  PCA(n_components=2).fit(X)
x_reduced = PCA(n_components=2).fit_transform(X)
pca.components_
pca.explained_variance_ratio_
```

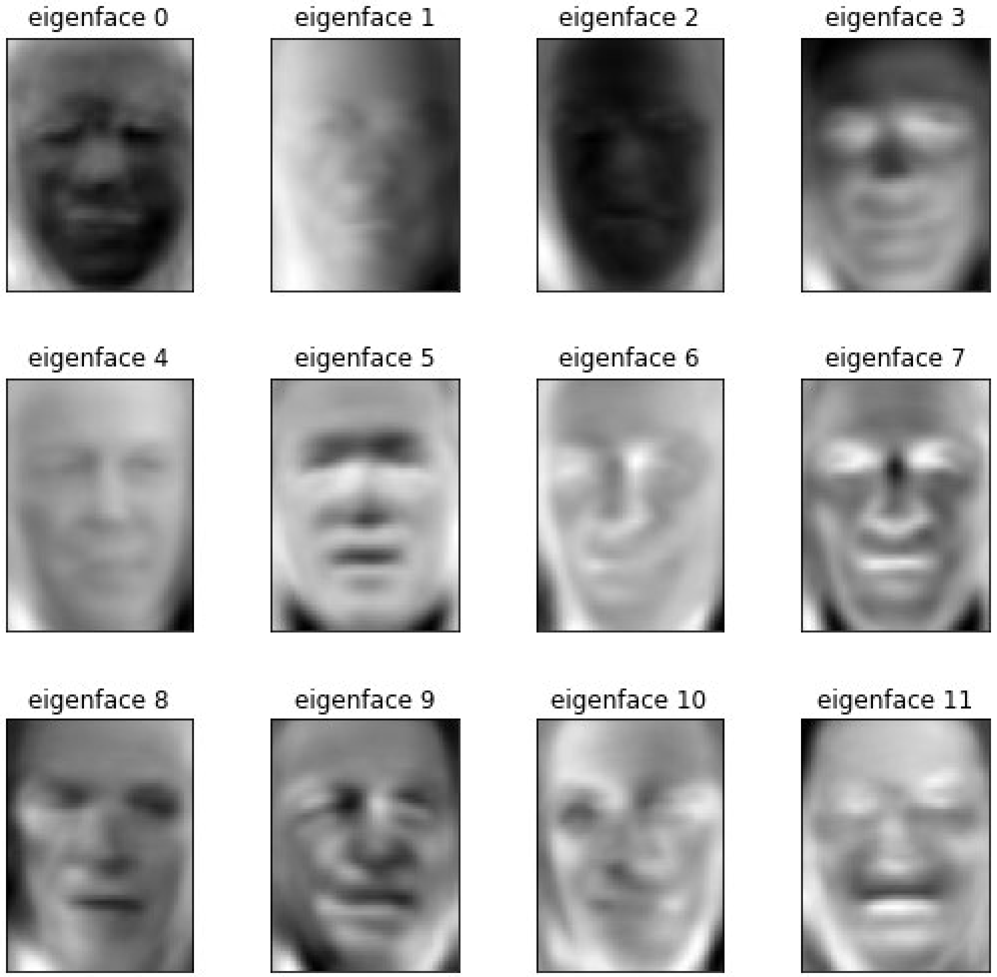
PCA Applications

Principal Component Regression (PCR)

- Use transformed features
- The transformed features are uncorrelated
- Lower dimension helps
- Difficult to interpret

PCA Applications

Eigenfaces, Face recognition



Turk, Matthew A; Pentland, Alex P (1991). [Face recognition using eigenfaces]()

Summary

- PCA as Dimensionality Reduction Techniques
- Finds axes that maximize the variance
- Explained variance ratio
- Feature selection
- Applications to PCR and face detection