CSCI4622 Machine Learning

2020 Spring

Unsupervised Learning

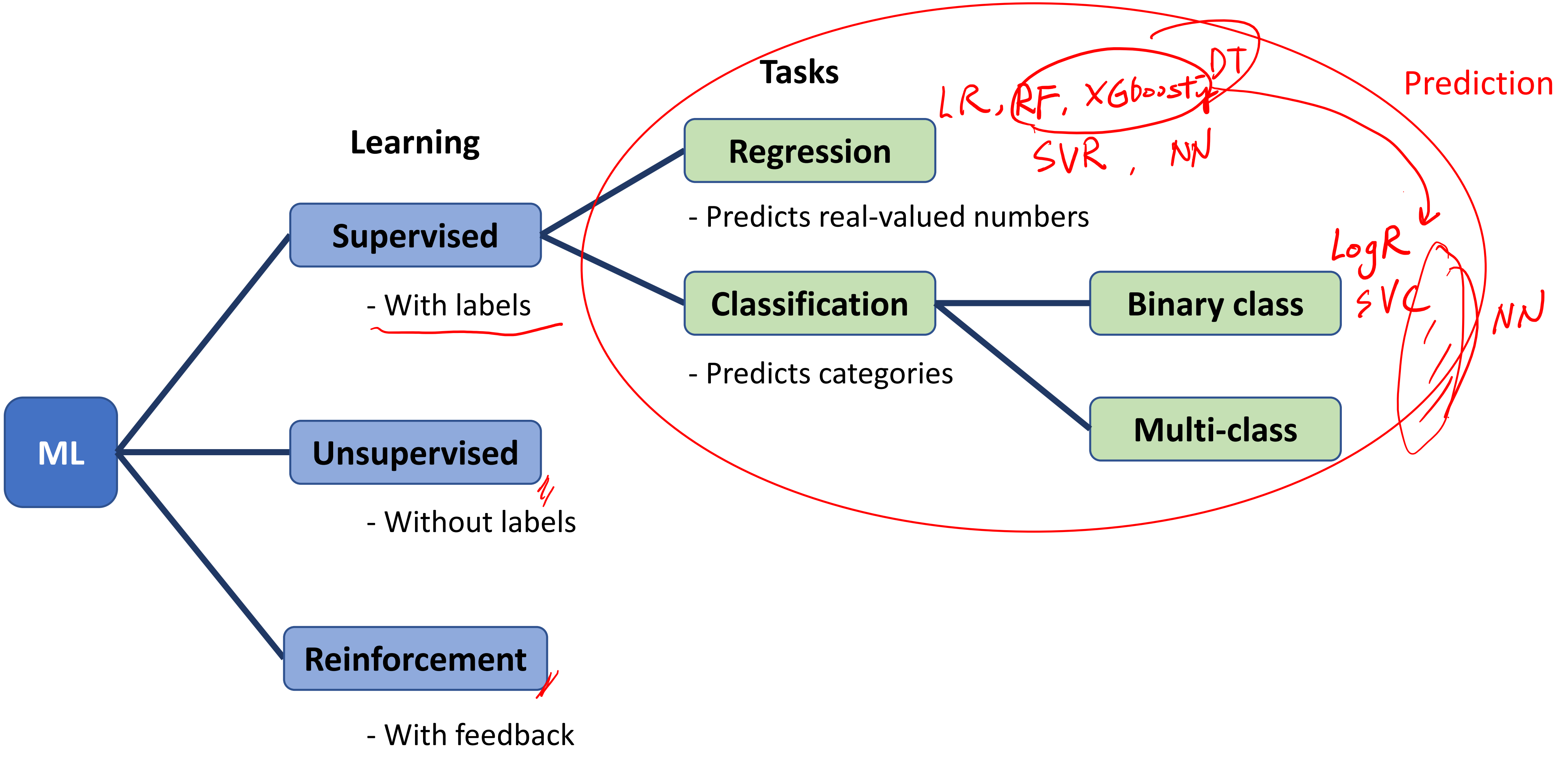
Geena Kim

*Some of the slide/diagram adopted from CMU deep learning course

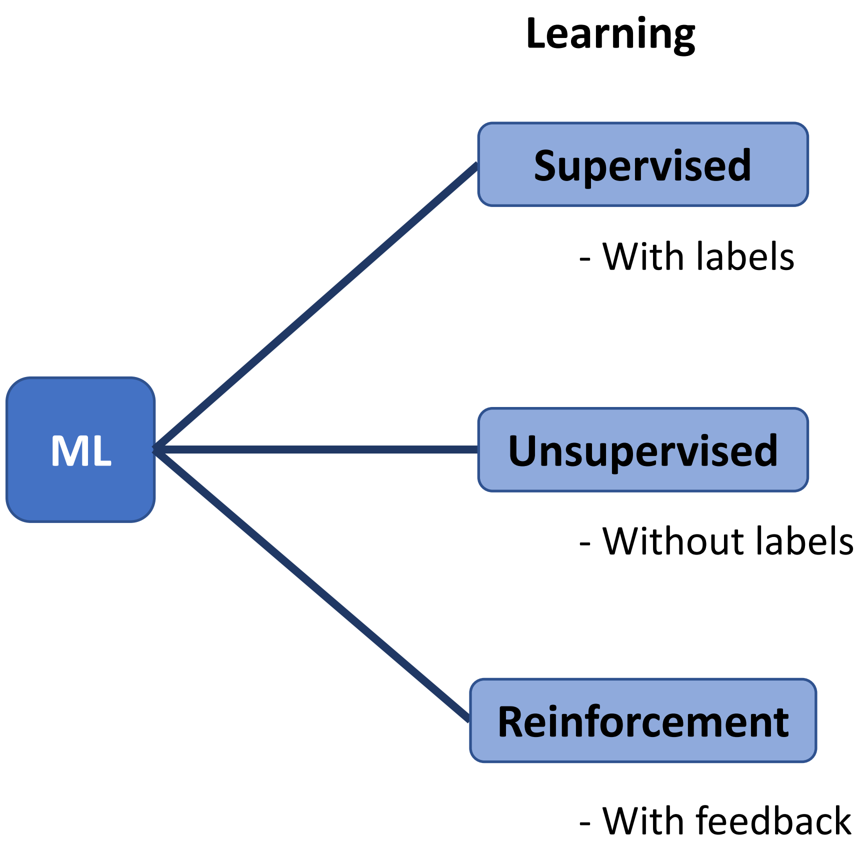
Unsupervised Learning

*Slide contents adopted from ISLR material

Types of machine learning problems



Types of machine learning problems



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

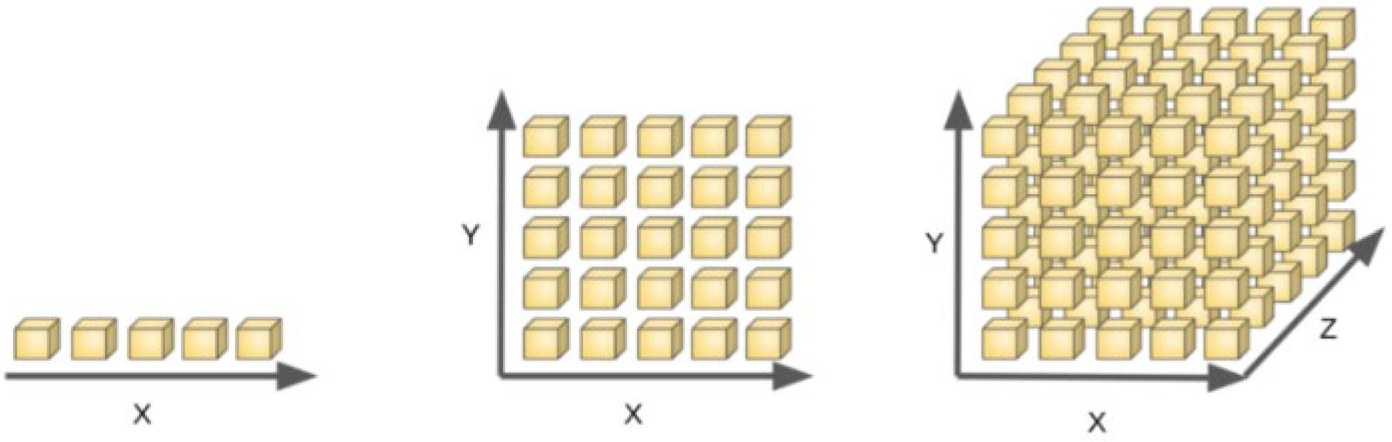
Dimensionality Reduction

Curse of dimensionality

Data become sparse

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

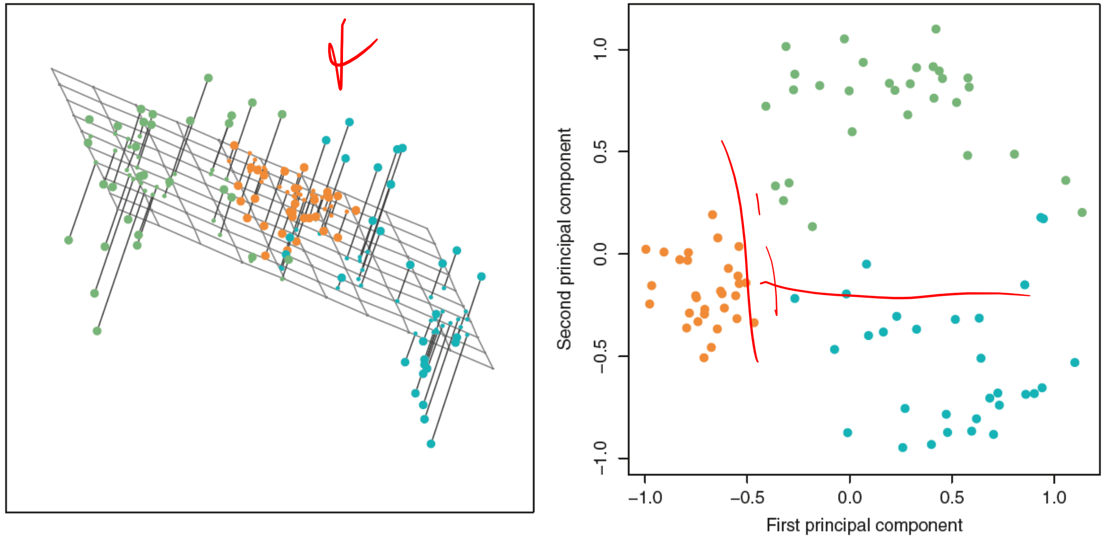
Likely to overfit



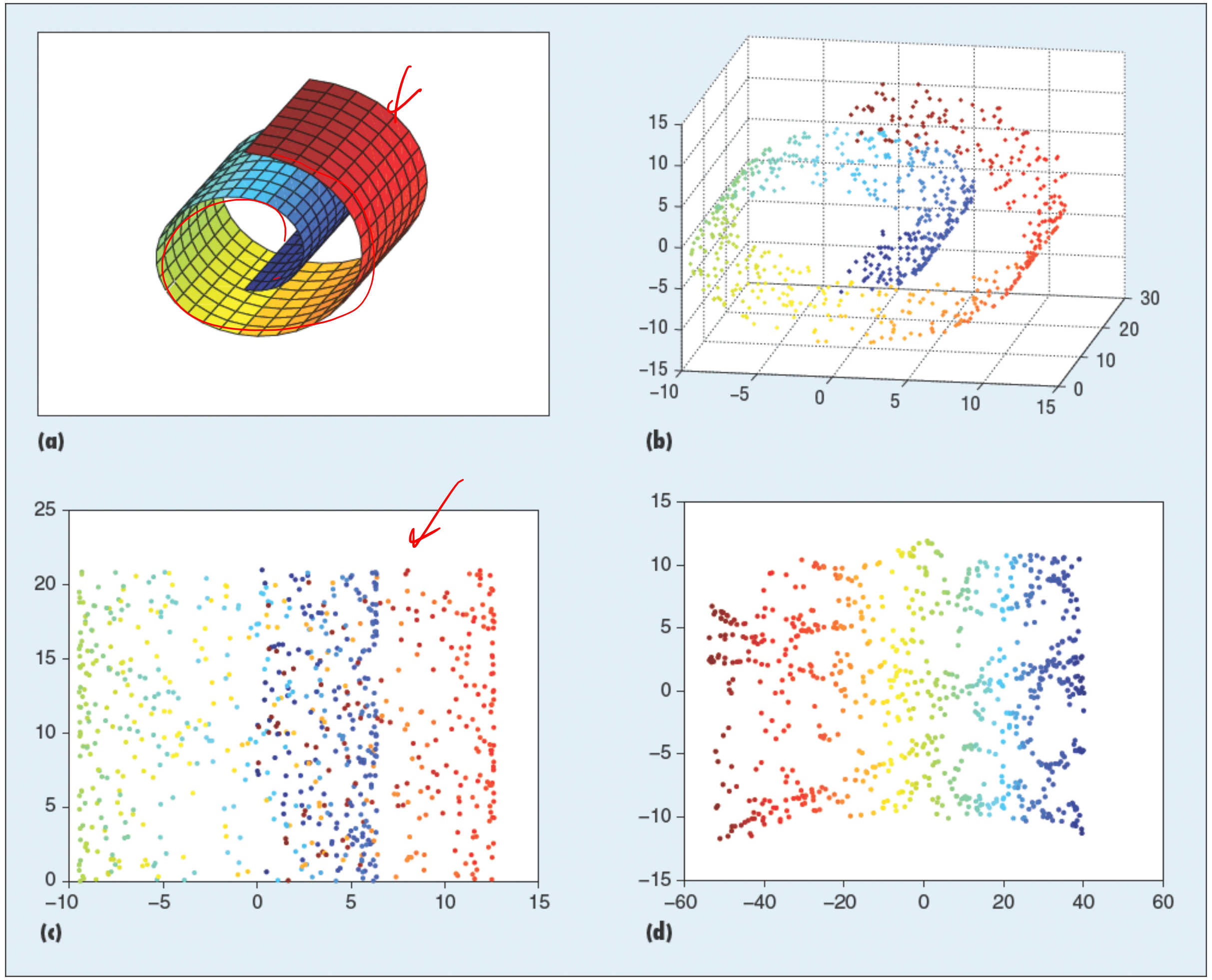
Dimensionality Reduction

t SNE

Projection to low-dimension



Manifold learning



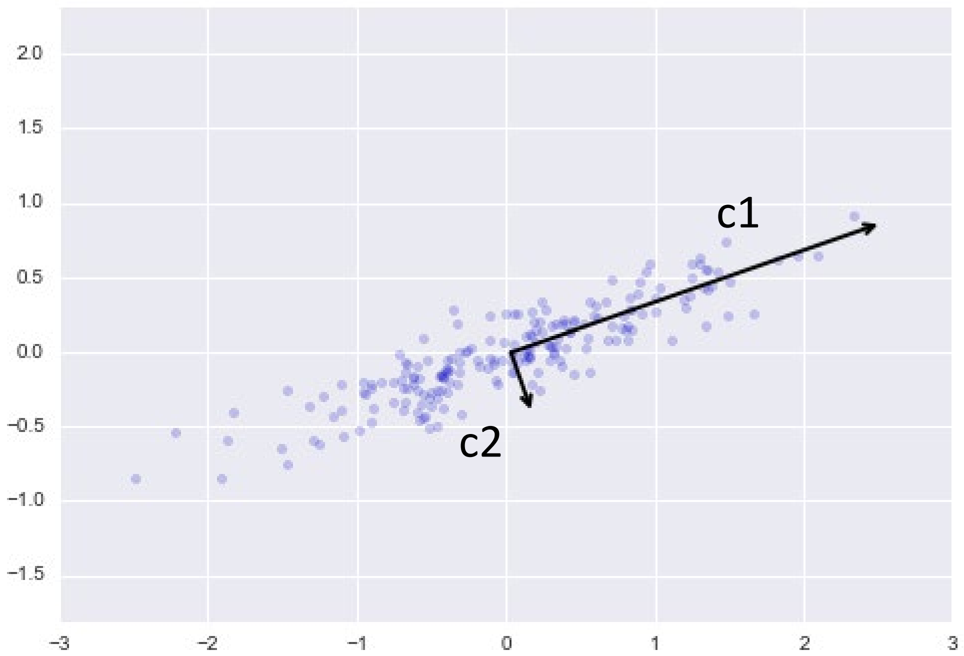
Principal Component Analysis (PCA)

$x_1 \; ... \; x_n$

PCA is a popular dimensionality reduction technique

$f_1 \; ... \; f_n$

Principal components



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

Normalized loading vectors

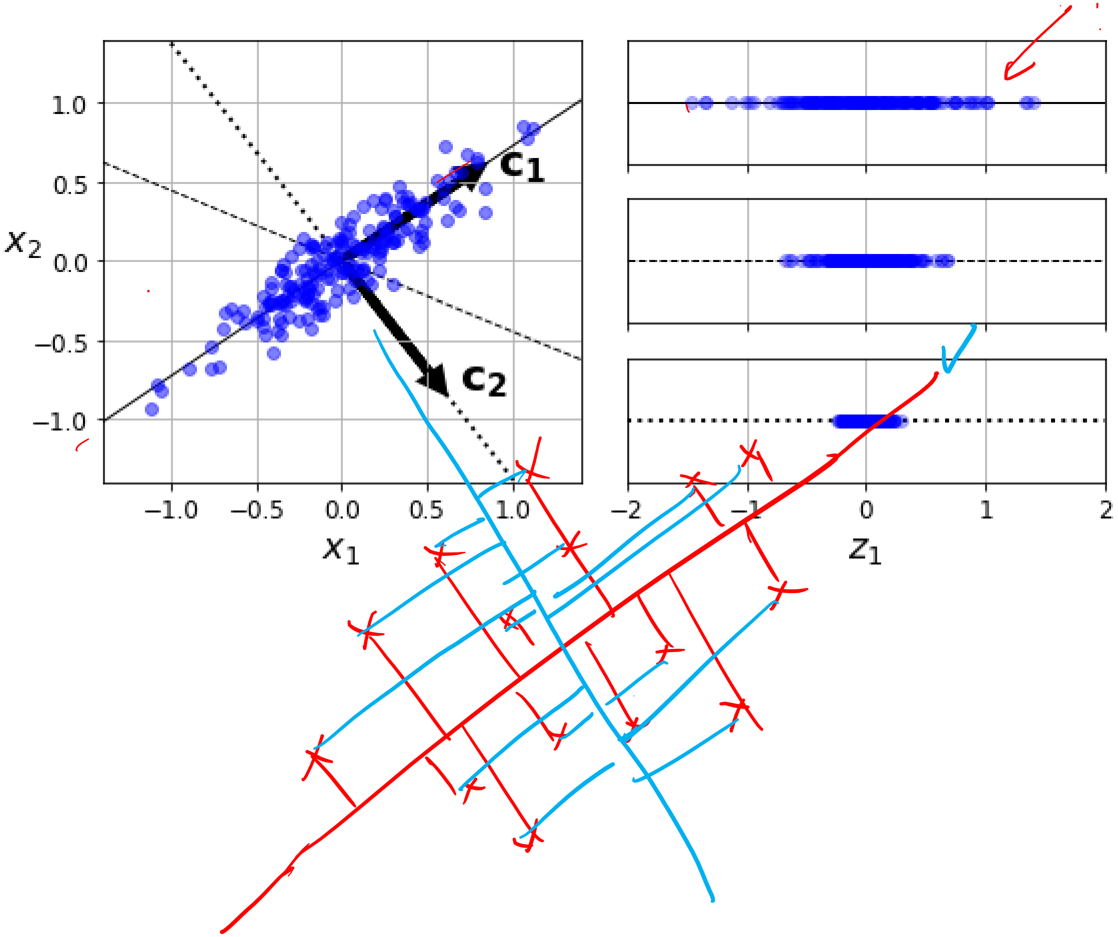
$$\sum_{j=1}^{p} \phi_{j1}^2 = 1$$

$$R^T R = I$$

Principal Component Analysis (PCA)

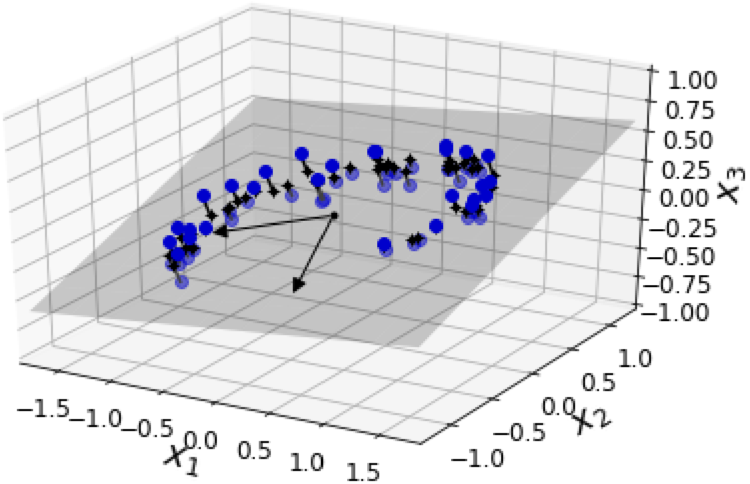
c_1

How to choose the principal components?



Method 1. Preserve the maximum variance

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



Principal Component Analysis (PCA)

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

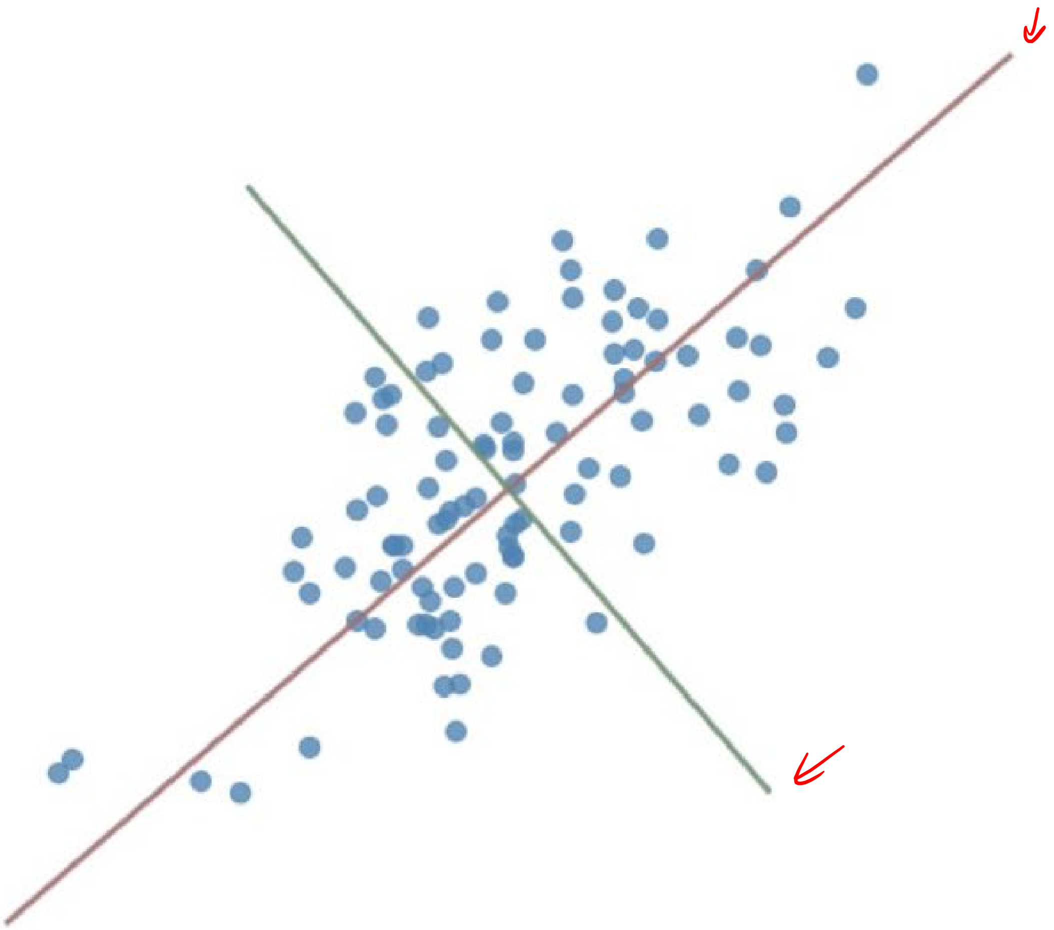
- Should capture largest variance in data
- Should probably be a unit vector $\|v\| = 1$

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

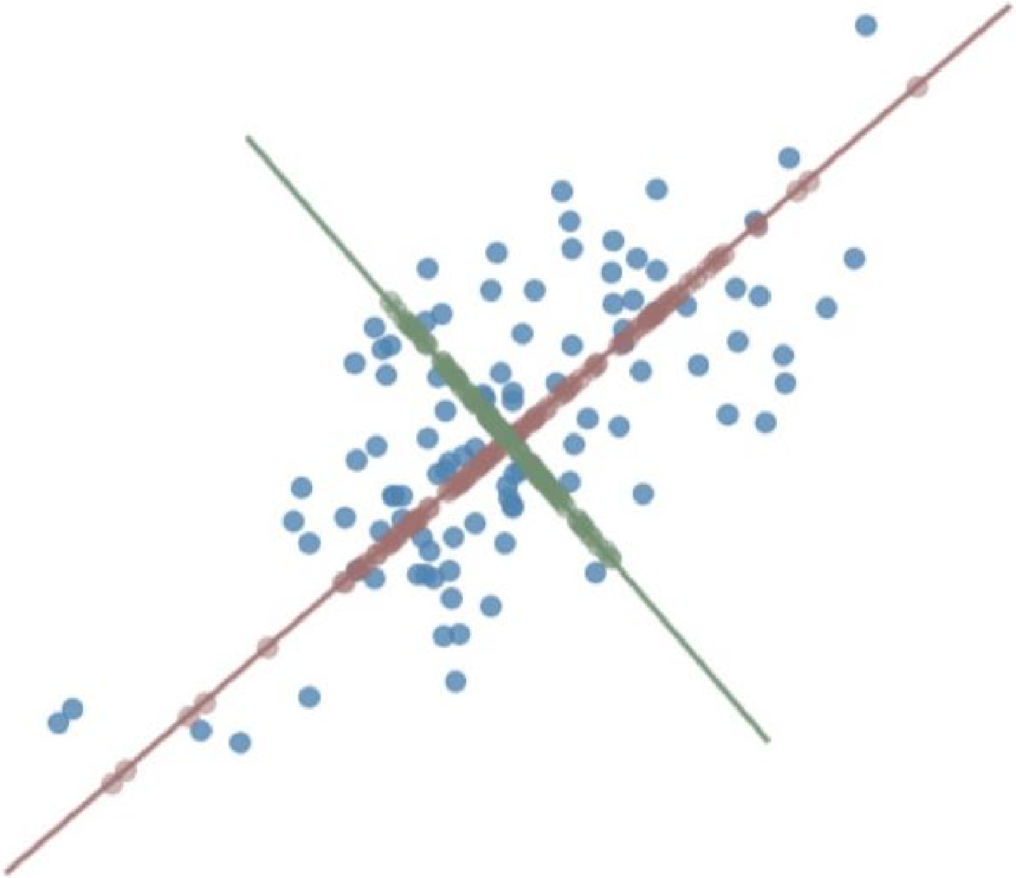
Principal Component Analysis (PCA)

Principal components of the previous example



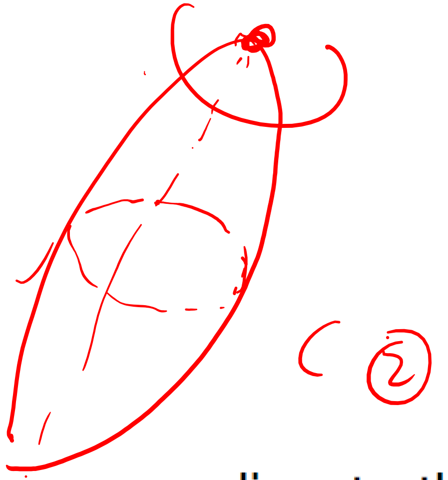
Principal Component Analysis (PCA)

Principal components of the previous example



Principal Component Analysis (PCA)

①



OK, so how do we find the first principle component?
Store data in an $m \times D$ matrix X (where $\mathbf{x}_i$ are rows)
Define covariance matrix $C^X = \frac{1}{m-1} X^T X$
Claim: First principle component $\mathbf{v}_1$ is the eigenvector of C^X corresponding to the largest eigenvalue
Recall: $\mathbf{v}$ is an eigenvector of A with associated eigenvalue λ if

$$A\mathbf{v} = \lambda \mathbf{v},$$

$$\begin{pmatrix} & & \\ & R & \\ & & \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \end{pmatrix} \Rightarrow \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix}$$

$$x^T R x = \lambda I$$

Principal Component Analysis (PCA)

$$\begin{pmatrix} \lambda_1 & & & & \\ & \lambda_2 & & & \\ & & \lambda_3 & & \\ & & & \ddots & \\ & & & & \lambda_n \end{pmatrix}$$

Facts about $C^X = \frac{1}{m-1} X^T X$

- Symmetric
- All eigenvalues are real (b/c symmetric)
- All eigenvalues are nonnegative (because it is positive semidefinite)
- C^X has $\cancel{D}$ n mutually orthogonal eigenvectors (which can be scaled to unit length)

$$C^X = \sum_{i=1}^{D} \lambda_i \boldsymbol{v}_i \boldsymbol{v}_i^T ,$$

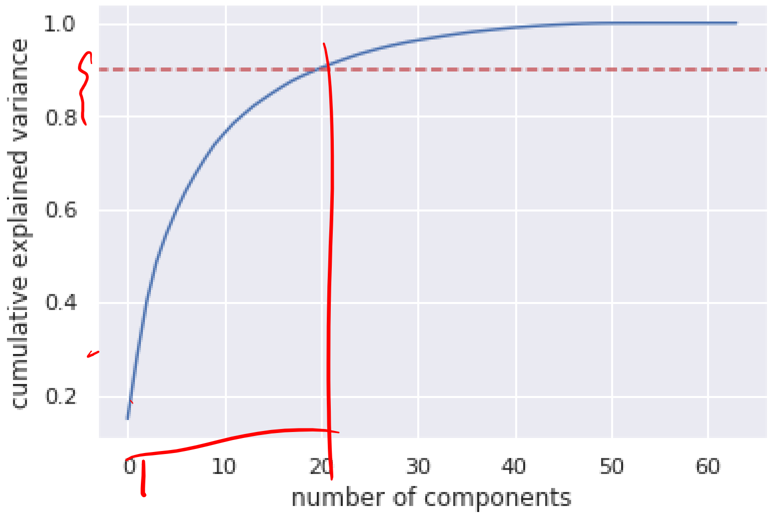
where λ_i are the eigenvalues $(\lambda_1 \geq \lambda_2 \geq \ldots, \geq \lambda_D)$, $\boldsymbol{v}_i$ is the eigenvector associated with λ_i.

Principal Component Analysis (PCA)

How many dimensions should we choose to use?

$$= \frac{V_j}{V(j=1 \dots N)}$$

"elbow" plot



1

What is explained variance?

What is explained variance ratio?

$x\%$ 0.0000001

$$\boxed{f_1} = \phi_1 x_1 + \phi_2 x_2 \dots \phi_n \cancel{x_n}$$

$$1 = \sum \phi^2$$

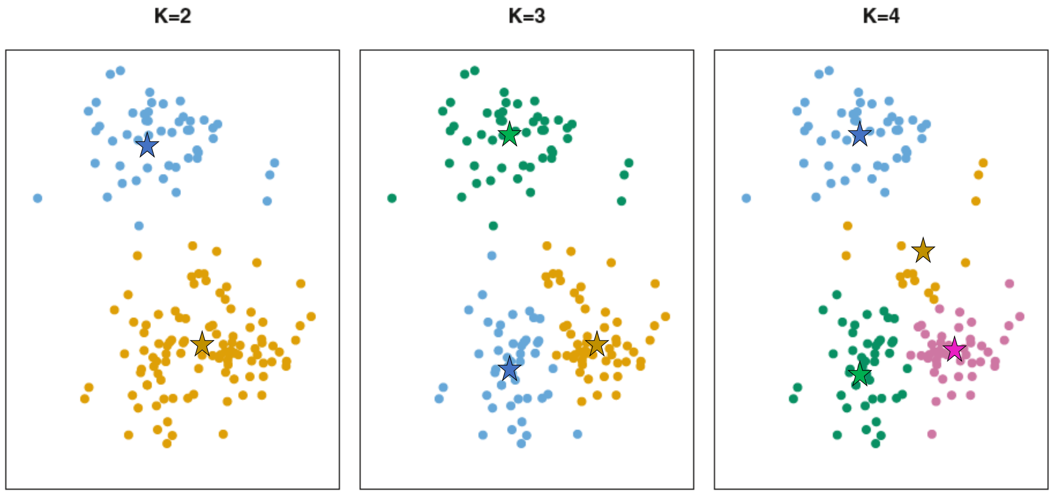
K-means Clustering

What is K-means clustering?

Cluster

Centroid

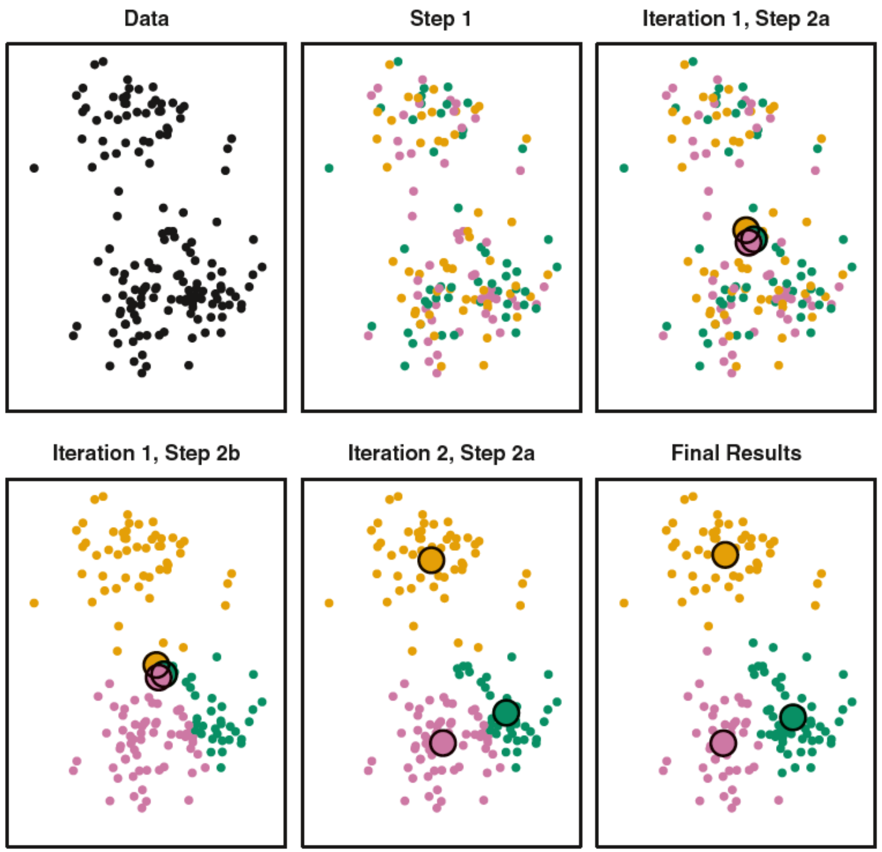
Euclidean distance



K-means Clustering

K-means Clustering Algorithm

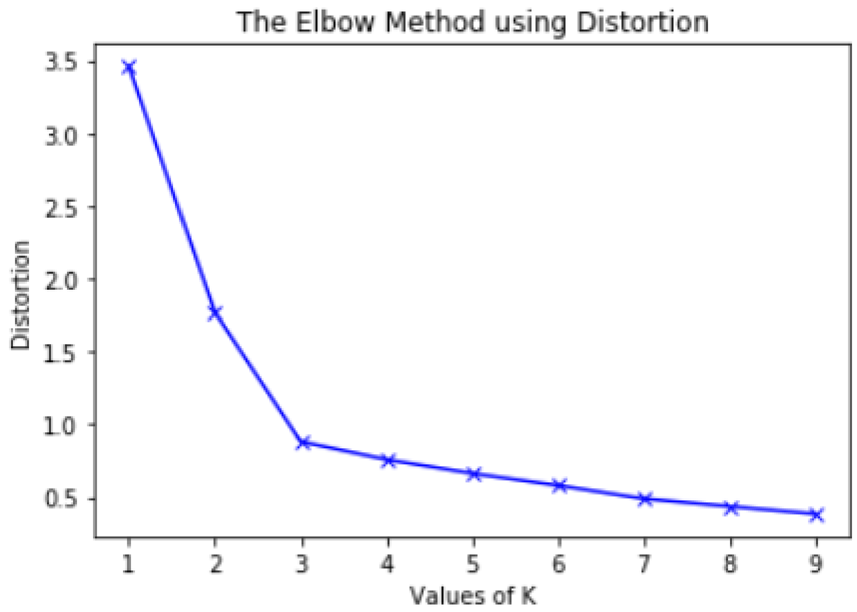
1. Randomly assign a number, from 1 to K, to each of the observations. These serve as initial cluster assignments for the observations.
2. Iterate until the cluster assignments stop changing:
   (a) For each of the K clusters, compute the cluster *centroid*. The kth cluster centroid is the vector of the p feature means for the observations in the kth cluster.
   (b) Assign each observation to the cluster whose centroid is closest (where *closest* is defined using Euclidean distance).

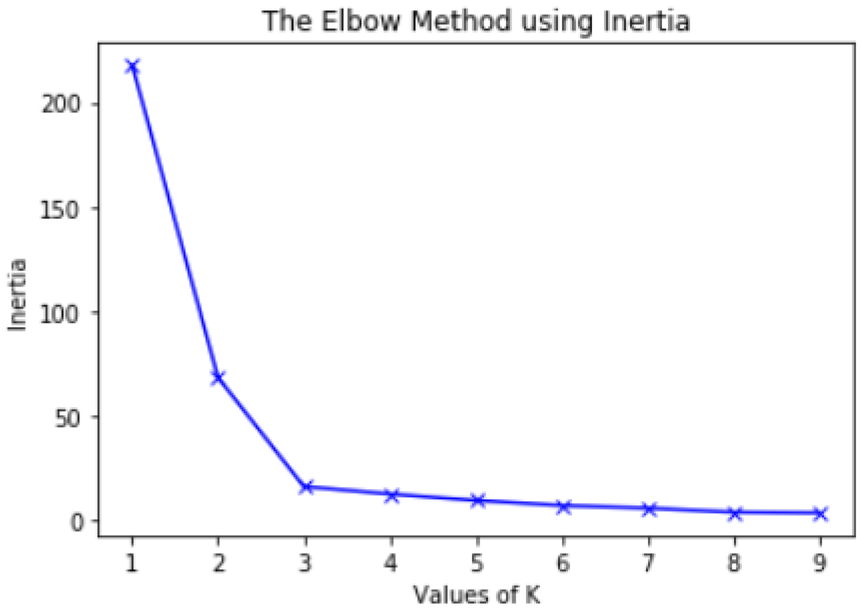


K-means Clustering

How to choose K?

Metric:
Distortion (the mean of square distance within a cluster)
Inertia (the sum of square distance within a cluster)





K-means Clustering

K-means Clustering

Need to decide how many clusters (K) before trying

Vulnerable to curse of dimensionality PCA preprocessing helps

Given enough time, K-means will always converge

Finds local minimum, not global minimum

The local minimum is highly dependent on the initialization of the centroids

sklearn's KMeans can initialize better if `init='k-means++'` is used

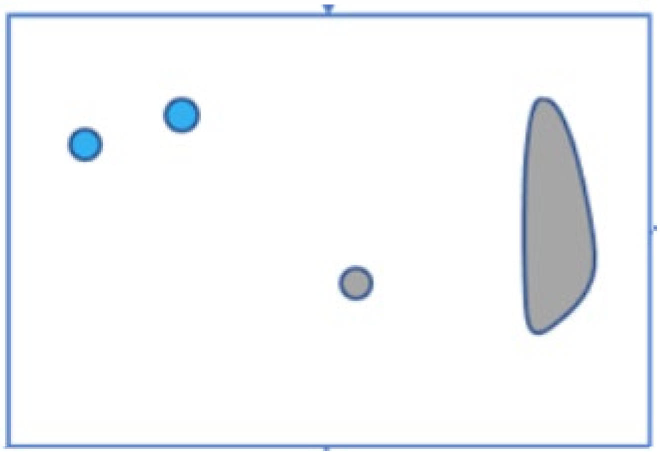
`MiniBatchKmeans` uses mini-batches to reduce the computation time

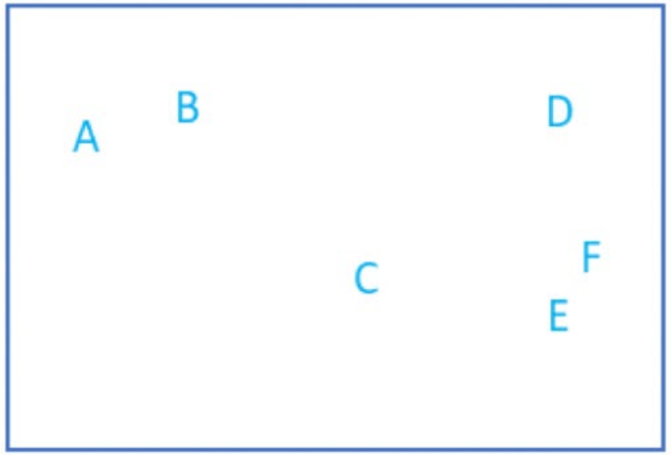
Hierarchical Clustering

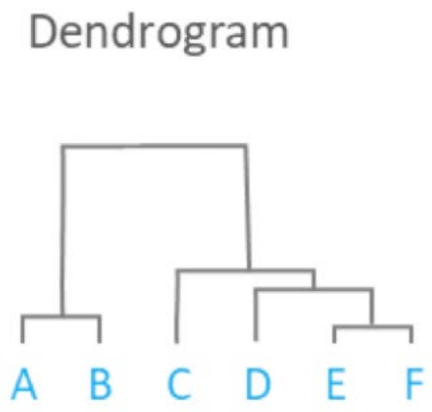
It does not need to know K in advance!

Dendrogram (upside down tree)

agglomerative hierarchical clustering







Distance: Euclidean, Correlation-based

Hierarchical Clustering

Finding clusters from the dendrogram