DASC 4113 Machine Learning

Lecture 10

Ukash Nakarmi

Unsupervised Learning

UNIVERSITY OF ARKANSAS

Learning Objectives

In this class, we will learn:

- What Principal Components are.
- How to use PCA for dimension reduction, Visualization and Clustering.
- Other Clustering tools such as K –Means.

Unsupervised Learning

Supervised Learning

- n measurements (observations)
- Each measurement has an associated response variable.
- Task: Well defined
- Prediction

Unsupervised Learning

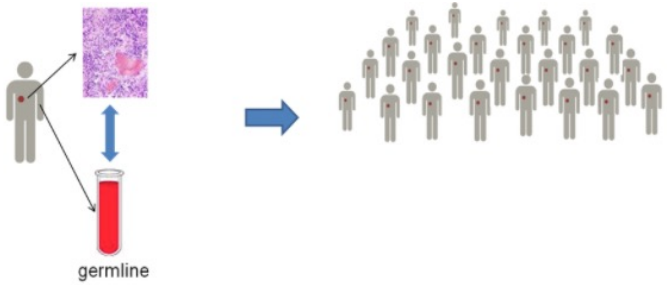
- n measurements (observations)
- Measurements have NO associated response variable.
- Task: Not so well defined
- Mostly Inference
- What can we discover infer from the data

- Visualization, Clustering,

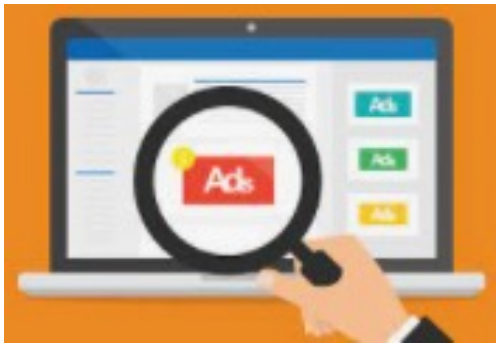
Unsupervised Learning

Some Examples:

- Cancer Gene Analysis



- Personalized Ads and Recommendation



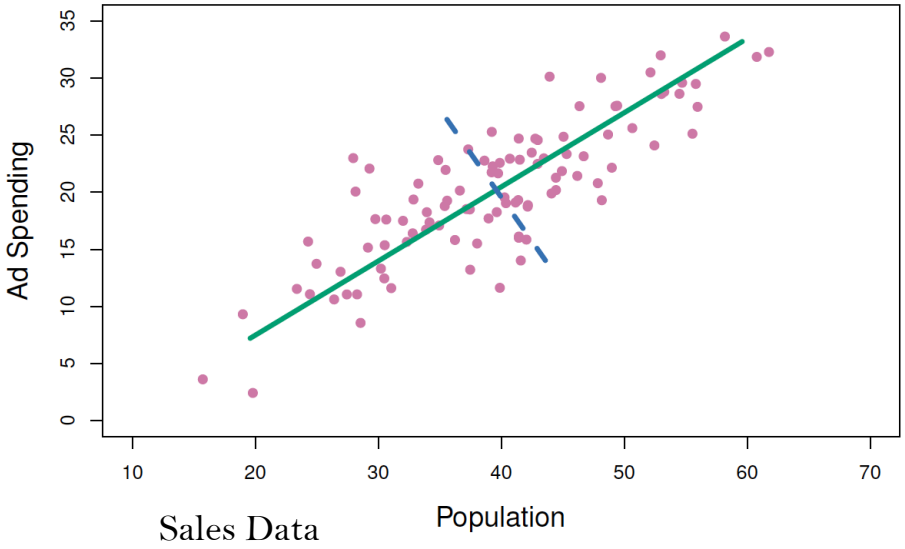
- Personalized Query and Search Optimization

Principal Component Analysis

Three Approaches:

- Subset Selection : Uses subset of p predictors that we believe to be more relevant, and model is fitted on reduced p.
- **Shrinkage:** Coefficients (β) of each input are shrunken towards zero. (Regularization)
- **Dimension Reduction:** Involves projecting p predictors into M-dimension where M < p. Then use M projections as predictors instead of original p predictors.

Recall:
Lecture 6, Model Selection and Regularization



Principal Component Used to :

- Reduce the dimension of the Input Features

PCs can have two interpretations:

1. Captures the direction of Variance
2. Line closest to the data (In a 2D case)

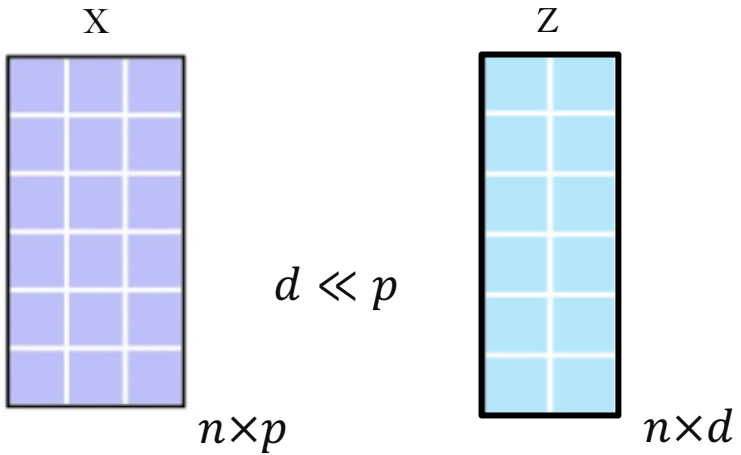
UNIVERSITY OF ARKANSAS

Principal Component Analysis

Some Applications:

- Dimension Reduction
- Data Visualization
- Data Imputation (Used in many fields: Recommendation, Image Inpainting, …)

Principal Component Analysis



Scenario:

- p features
- n measurements
- $n \times p$ data matrix X

Objective:

For each p dimensional measurement x_i ,

Find a d dimensional representation: $z_i,\ i = 1, 2, \ldots n$

Principal Component Analysis does this by:

- Finding each new dimension as a linear combination of p features.

Principal Component Analysis

Principal Component Analysis does this by:

- Finding each new dimension as a linear combination of p features.

The First Principal Component:

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

Example in a 2D: $p = 2$

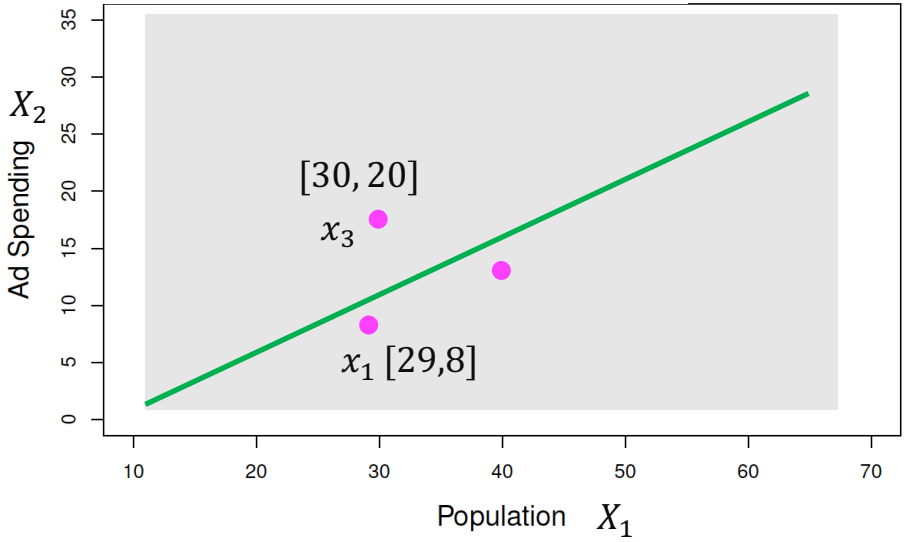
Slope of the line $= 5/10$

$\phi_{11} = 10, \qquad \phi_{21} = 5$

New representation of x_3:

$z_{13} = \phi_{11}x_{31} + \phi_{21}x_{32}$

$z_{13} = 10 \times 30 + 5 * 20$



Principal Component Analysis

The First Principal Component: $Z_1 = \phi_{11}X_1 + \phi_{21}X_2 + \cdots + \phi_{p1}X_p$

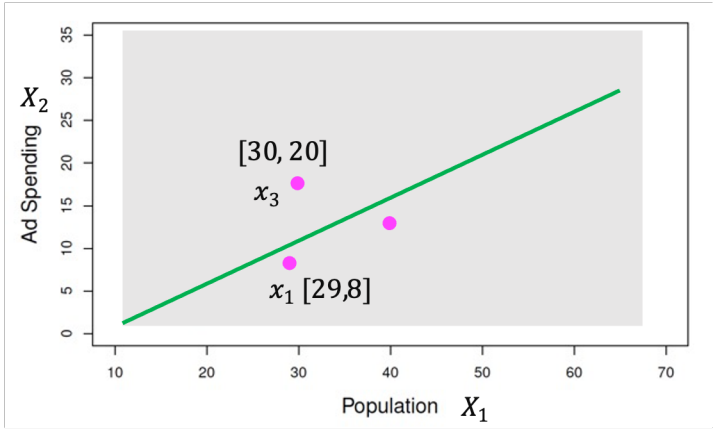
Slope of the line $= 5/10$

$\phi_{11} = 10, \qquad \phi_{21} = 5$

New representation of x_3:

$z_{13} = \phi_{11}x_{31} + \phi_{21}x_{32}$

$z_{13} = 10 \times 30 + 5 * 20$



New representation as a linear combination of original features.

Note: The feature along which data varied most is weighted higher $\phi_{11}(10) > \phi_{21}(5)$

ϕ_{ij}

i: index for input feature, $i = 1, 2, \ldots p$

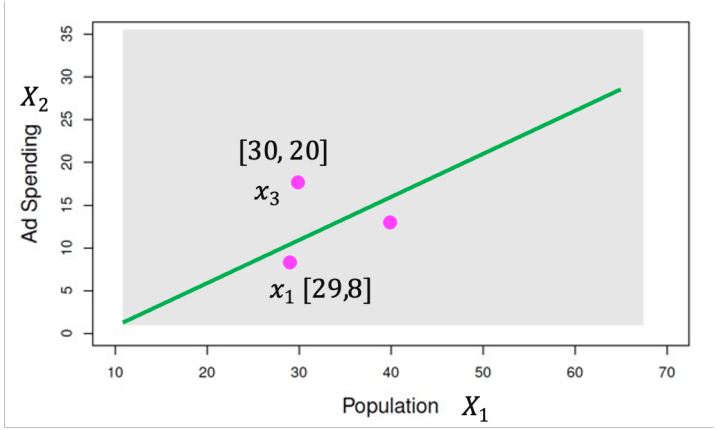
j: index for principal component , $j = 1, 2, \ldots d$

Principal Component Analysis

New representation of x_3:

$$z_{13} = \phi_{11}x_{31} + \phi_{21}x_{32}$$

$$z_{13} = 10\times30 + 5 * 20$$



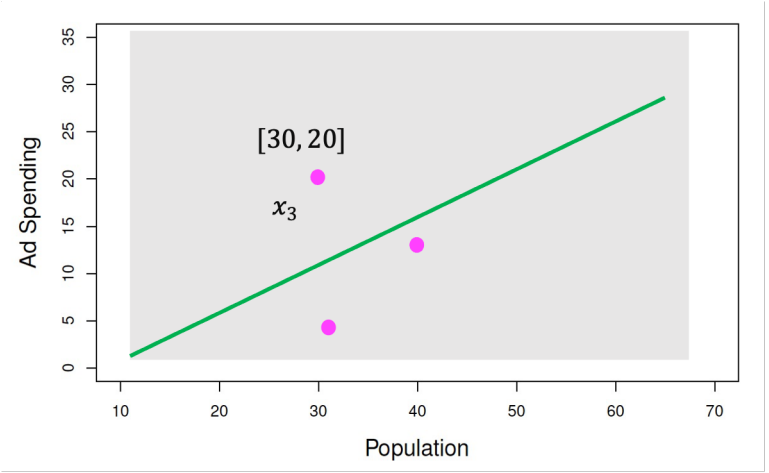
ϕ_{ij}

i: index for input feature, $i = 1, 2, \ldots p$

j: index for principal component , $j = 1, 2, \ldots d$

- How do we find ϕ ?
- What are its properties/Practical Consideration?

Principal Component Analysis

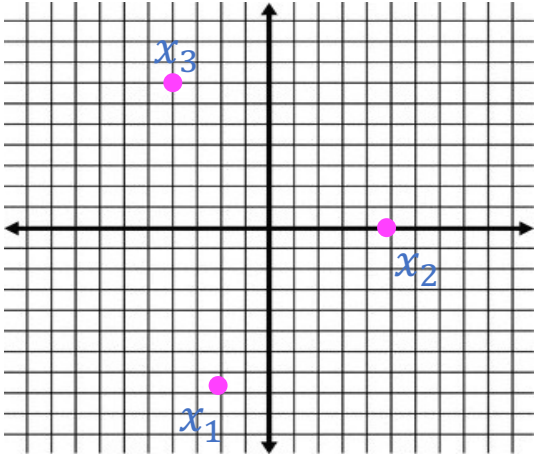


$X, n = 3, p = 2$

x_1	32	4
x_2	40	12
x_3	30	20

34 — Mean of Feature X_1

12 — Mean of Feature X_2



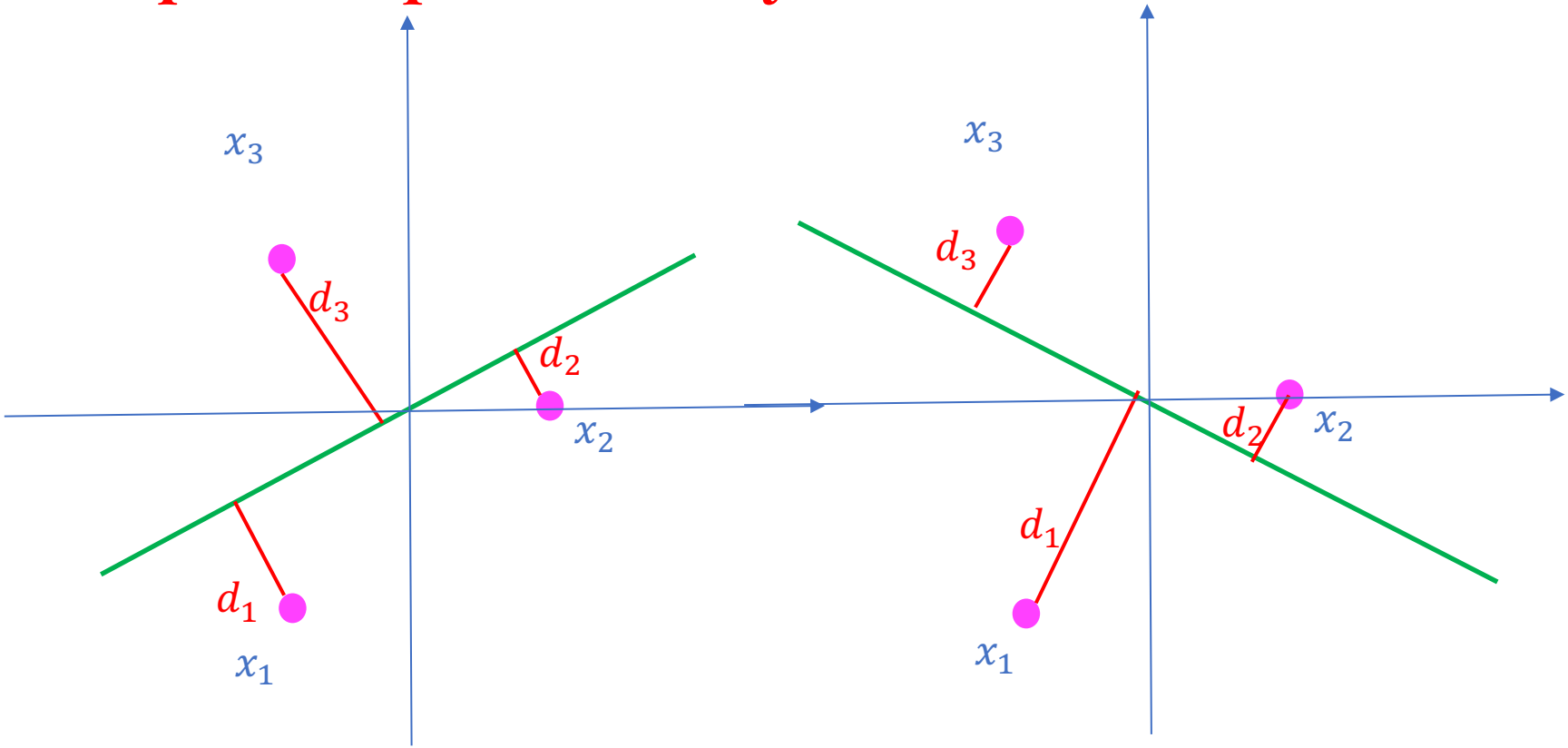
Note: Even after mean centering, the relation (relative position) of data points x_1, x_2, x_3 has not changed in new shifted representation

Mean centered for each feature

−2	−8
6	0
−4	8

- PC does not care about the intercept (only slope)

Principal Component Analysis

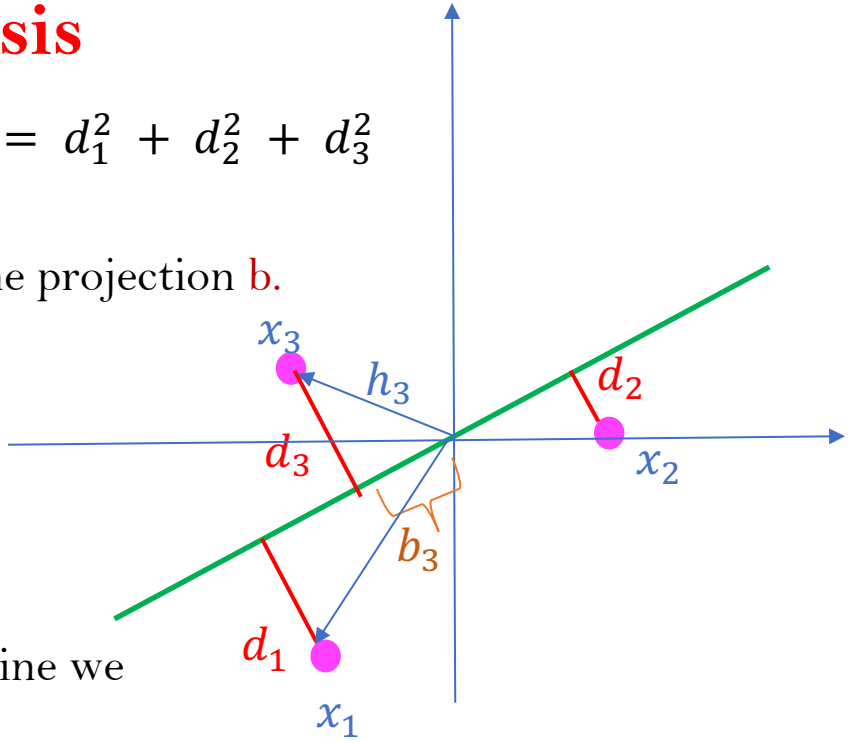


- Finding 1st PC ~ A line that minimizes $d_1^2 + d_2^2 + d_3^2$
- A line (subspace) closest to the data point : First Interpretation of PC

Principal Component Analysis

- Finding 1st PC ~ A line that minimizes $d^2 = d_1^2 + d_2^2 + d_3^2$

For any data point x
A line that minimizes distance d maximizes the projection b.



For example: point x_3
From Pythagorean:
$h_3^2 = d_3^2 + b_3^2$.

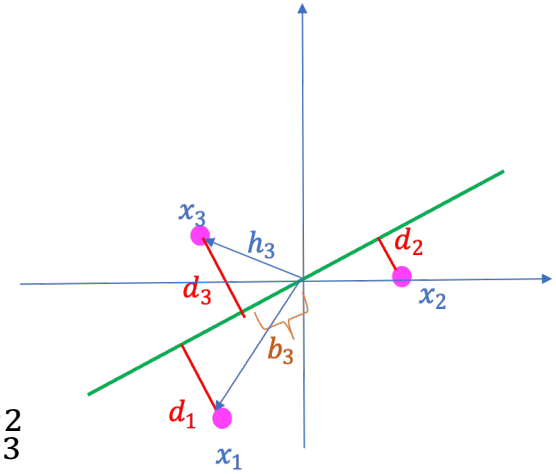
- Note for all data points, no matter which line we choose, h remains same.

- This means if a line we choose minimizes d^2 , it maximizes the projection b^2

Principal Component Analysis

- Finding 1st PC ~ A line that minimizes $d_1^2 + d_2^2 + d_3^2$

For any data point x
A line that minimizes distance d maximizes the projection b.



- Finding 1st PC ~ A line that minimizes $d_1^2 + d_2^2 + d_3^2$

Is equivalent to

- A line that maximizes $b_1^2 + b_2^2 + b_3^2$

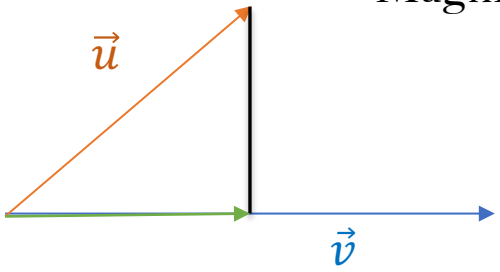
2nd Interpretation of PC: Captures the direction of maximum variance

$b_1, b_2, b_3 \ldots b_n$ are nothing but projection of data points into the 1st PC.

$\phi_{11}x_{11} + \phi_{21}x_{12}$

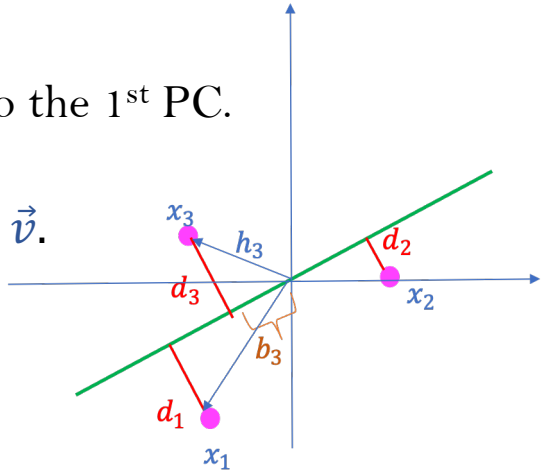
Principal Component Analysis

$b_1, b_2, b_3 \ldots b_n$ are nothing but projection of data points into the 1st PC.



Magnitude of Projection of vector $\vec{u}$ *onto* $\vec{v}$.

$$|proj_{\vec{v}}\ \vec{u}| = |\left(\frac{\vec{u}.\vec{v}}{|\vec{v}|^2}\right)|$$



If $\vec{v}$ is a unit vector than projection can be approximated by a dot product.

So, ϕ is constrained to have magnitude of 1.

$$\phi_{11}^2 + \phi_{21}^2 + \cdots \phi_{p1}^2 = 1$$

Principal Component Analysis

Principal Component Analysis does this by:

- Finding each new dimension as a linear combination of p features.

The First Principal Component:

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

Example in a 2D: $p = 2$

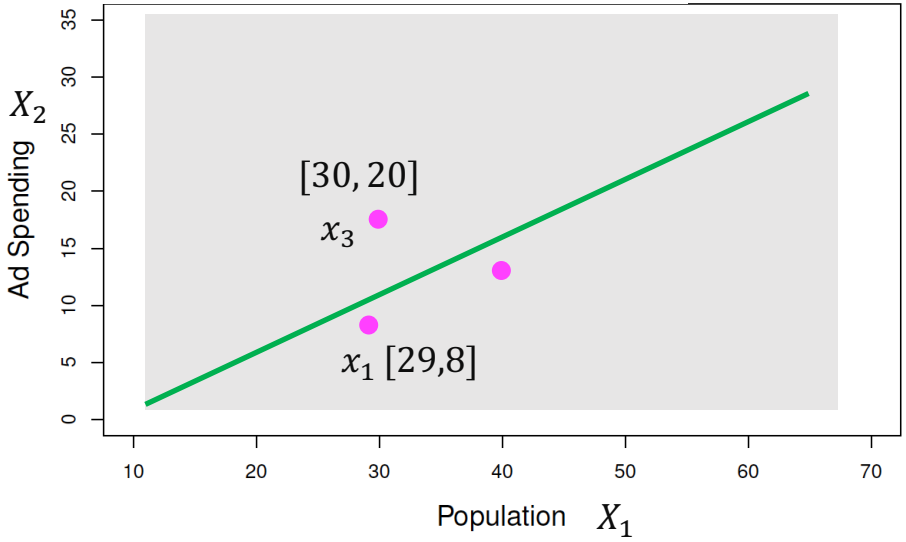
Slope of the line $= 5/10$

$\phi_{11} = 10, \qquad \phi_{21} = 5$

Normalized Φ

$$\phi_{11} = \frac{10}{\sqrt{10^2 + 5^2}}, \phi_{21} = \frac{5}{\sqrt{10^2 + 5^2}}$$

$\phi_{11} = 0.89, \qquad \phi_{21} = 0.45$



Principal Component Analysis

Principal Component Analysis does this by:

- Finding each new dimension as a linear combination of p features.

The First Principal Component:

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

Example in a 2D: $p = 2$

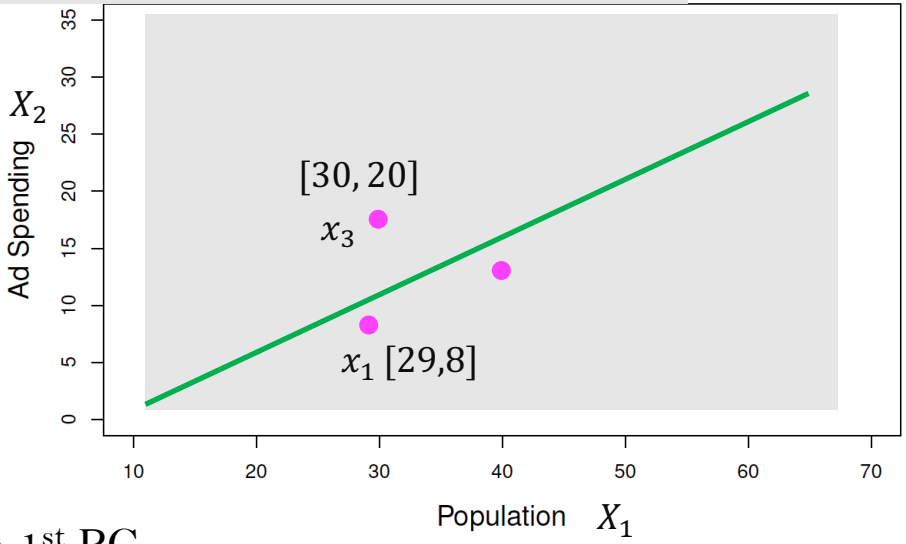
Slope of the line $= 5/10$

Normalized:

$\phi_{11} = 0.89, \qquad \phi_{21} = 0.45$

New representation of x_3:

$z_{13} = \phi_{11}x_{31} + \phi_{21}x_{32}$



$[\phi_{11}\ \phi_{21}]$ -> Loading Vectors

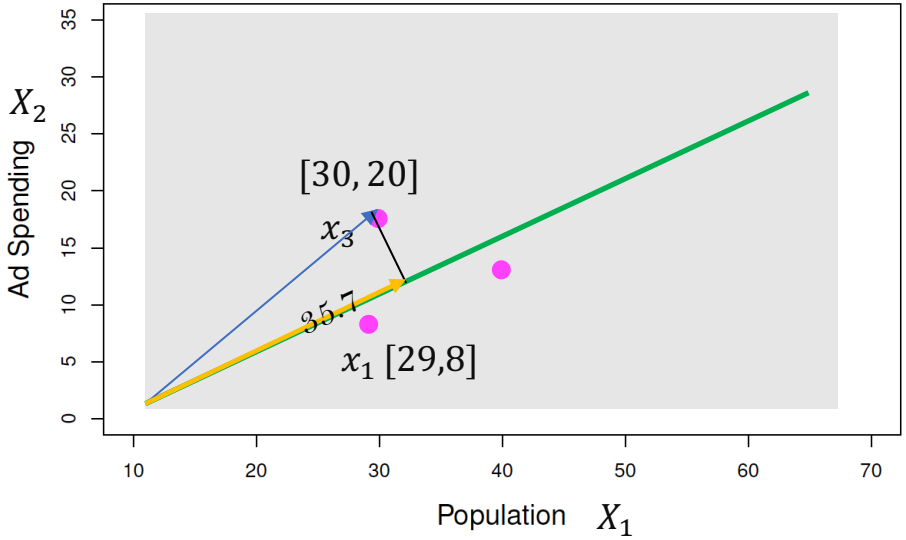
z_{13} -> Score of data Point on 1st PC

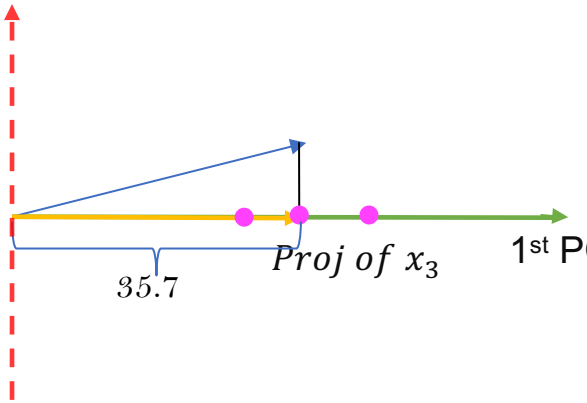
Principal Component Analysis

New representation of x_3:

$$z_{13} = \phi_{11}x_{31} + \phi_{21}x_{32}$$

$$z_{13} = 0.89 * 30 + 0.45*20$$
$$= 35.7$$



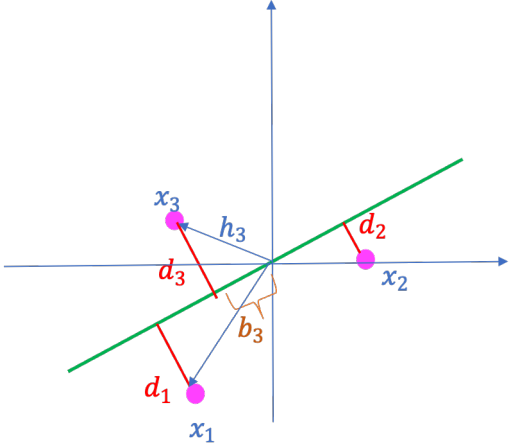


1st PC: $0.89 * population + 0.45 * Ad\ Spending$

Principal Component Analysis

The First Principal Component:

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



Summary : Finding 1st Principal component

$$\underset{\phi_{11},\ldots,\phi_{p1}}{\text{maximize}} \left\{ \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{j=1}^{p}\phi_{j1}x_{ij}\right)^2 \right\} \text{ subject to } \sum_{j=1}^{p}\phi_{j1}^2 = 1.$$

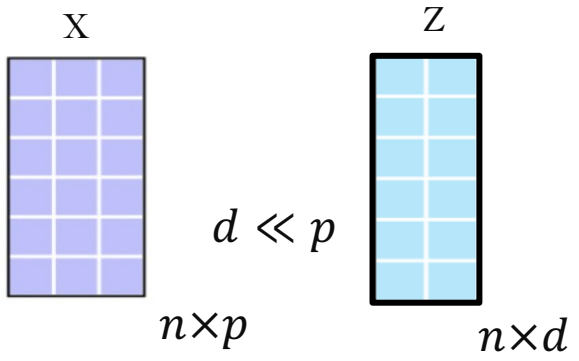
Similarly, 2nd Principal component captures the direction 2nd direction along which data varies most, and so and so forth……

Solution is given by : Either Eigen Decomposition of a Covariance of a data matrix or Singular Value Decomposition of a Data Matrix.

Principal Component Analysis

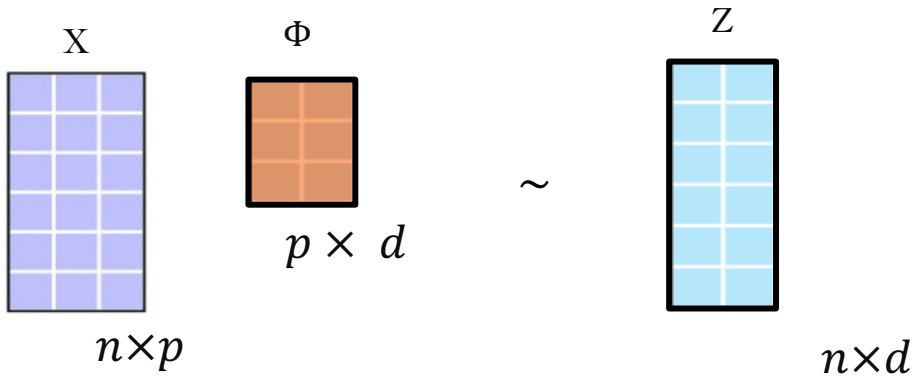
Scenario:

- p features
- n measurements
- $n \times p$ data matrix X



Principal Component Analysis does this by:

- Finding each new dimension as a linear combination of p features.



$d \ll p$

Principal Component Analysis: Example

Dataset: *USArrest:* Numbers of arrest per 100,000 resident related to 3 different crimes and urban population for or each 50 US states.

p = 4, i.e., 4 input features: Assault, Murder, Rape, UrbanPopulation

Crime

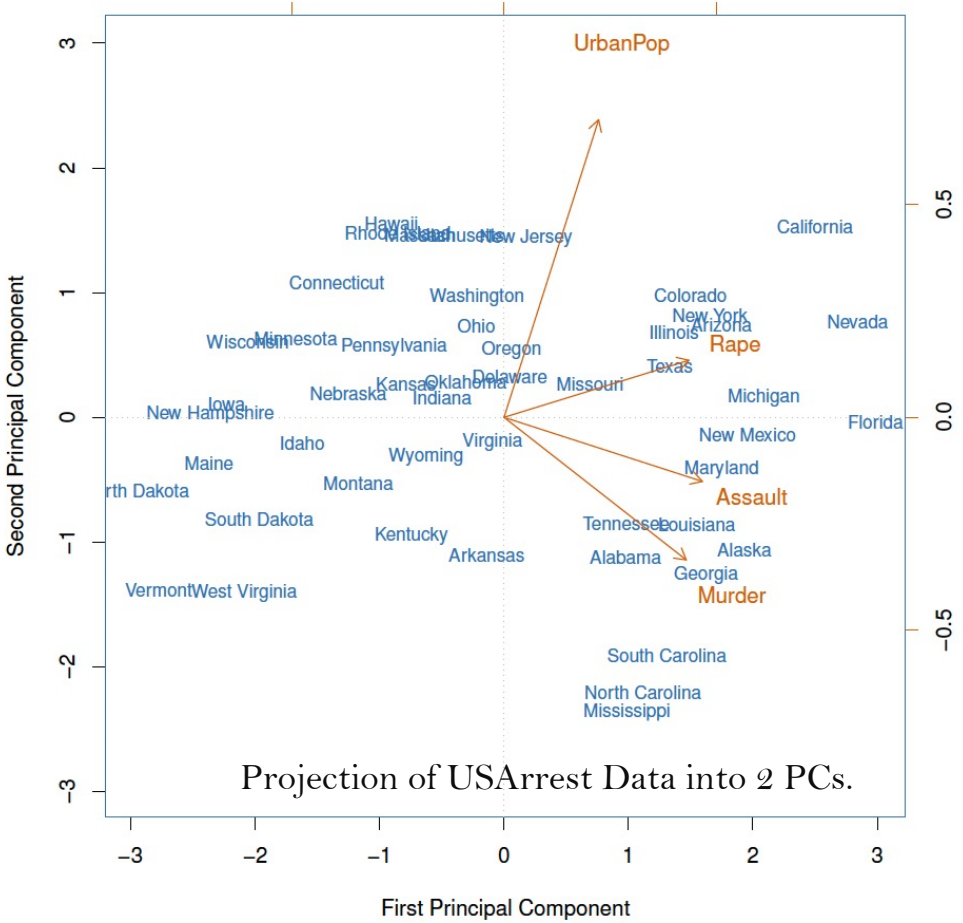
Urbanization

n = 50, Data for 50 US states

d = 2, compute 2 principal components

	PC1		PC2	
Murder	0.5358995	(ϕ_{11})	−0.4181809	(ϕ_{12})
Assault	0.5831836	(ϕ_{21})	−0.1879856	(ϕ_{22})
UrbanPop	0.2781909	(ϕ_{31})	0.8728062	(ϕ_{32})
Rape	0.5434321	(ϕ_{41})	0.1673186	(ϕ_{42})

Principal Component Analysis: Example



	PC1		PC2	
Murder	0.5358995	(ϕ_{11})	−0.4181809	(ϕ_{12})
Assault	0.5831836	(ϕ_{21})	−0.1879856	(ϕ_{22})
UrbanPop	0.2781909	(ϕ_{31})	0.8728062	(ϕ_{32})
Rape	0.5434321	(ϕ_{41})	0.1673186	(ϕ_{42})

- 1st PC Almost equally load 3 crime variables: Murder, Assault, Rape.
- Three crime variable are correlated.
- 2nd PC highly loads UrbanPop.
- 2nd PC corresponds to the urbanization.
- Data Points (States) which have higher score(component) in first direction: Michigan, Florida, Nevada, California have high crime rates and vice versa.
- Data Points (States) which have higher score(component) in 2nd direction: Hawaii, New Jersey, California have high urbanization and vice versa.

Principal Component Analysis: Proportion of Variance

- Principal Components capture the direction of maximum variance.
- How much variance is captured by each PC ?

The total Variance present in the data: (assuming the data is mean centered)

$$\sum_{j=1}^{p} \mathrm{Var}(X_j) = \sum_{j=1}^{p} \frac{1}{n} \sum_{i=1}^{n} x_{ij}^2$$

The Variance captured by mth Principal component:

$$\frac{1}{n} \sum_{i=1}^{n} z_{im}^2 = \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{j=1}^{p} \phi_{jm} x_{ij} \right)^2$$

Sum of squares of all scores on mth PC

Principal Component Analysis: Proportion of Variance

- Proportion of Variance (PVE):

Variance captured by m[th] Principal component

Total Variance

$$\frac{\sum_{i=1}^{n} z_{im}^2}{\sum_{j=1}^{p}\sum_{i=1}^{n} x_{ij}^2} = \frac{\sum_{i=1}^{n}\left(\sum_{j=1}^{p}\phi_{jm}x_{ij}\right)^2}{\sum_{j=1}^{p}\sum_{i=1}^{n} x_{ij}^2}$$

If we use first M principal Components

$$\underbrace{\sum_{m=1}^{M}\frac{1}{n}\sum_{i=1}^{n} z_{im}^2}_{\text{Var. of first } M \text{ PCs}}$$

Principal Component Analysis: Proportion of Variance

If we use first M principal Components:

Total Variance in Data = Variance Represented by M PCs + Error of Representation

$$\underbrace{\sum_{j=1}^{p}\frac{1}{n}\sum_{i=1}^{n}x_{ij}^2}_{\text{Var. of data}} = \underbrace{\sum_{m=1}^{M}\frac{1}{n}\sum_{i=1}^{n}z_{im}^2}_{\text{Var. of first } M \text{ PCs}} + \underbrace{\frac{1}{n}\sum_{j=1}^{p}\sum_{i=1}^{n}\left(x_{ij}-\sum_{m=1}^{M}z_{im}\phi_{jm}\right)^2}_{\text{MSE of } M\text{-dimensional approximation}}$$

Fixed for Given Data

More the variance is captured by PCs → Smaller is the Error of Representation

Principal Component Analysis: Proportion of Variance

If we use first M principal Components:

Total Variance in Data = Variance Represented by M PCs + Error of Representation

$$\underbrace{\sum_{j=1}^{p}\frac{1}{n}\sum_{i=1}^{n}x_{ij}^2}_{\text{Var. of data}} = \underbrace{\sum_{m=1}^{M}\frac{1}{n}\sum_{i=1}^{n}z_{im}^2}_{\text{Var. of first } M \text{ PCs}} + \underbrace{\frac{1}{n}\sum_{j=1}^{p}\sum_{i=1}^{n}\left(x_{ij}-\sum_{m=1}^{M}z_{im}\phi_{jm}\right)^2}_{\text{MSE of } M\text{-dimensional approximation}}$$

Fixed for Given Data

More the variance is captured by PCs

Smaller is the Error of Representation

PCA as R^2 approximation of a data matrix

Principal Component Analysis: Proportion of Variance

$$\underbrace{\sum_{j=1}^{p} \frac{1}{n} \sum_{i=1}^{n} x_{ij}^2}_{\text{Var. of data}} = \underbrace{\sum_{m=1}^{M} \frac{1}{n} \sum_{i=1}^{n} z_{im}^2}_{\text{Var. of first } M \text{ PCs}} + \underbrace{\frac{1}{n} \sum_{j=1}^{p} \sum_{i=1}^{n} \left(x_{ij} - \sum_{m=1}^{M} z_{im} \phi_{jm} \right)^2}_{\text{MSE of } M\text{-dimensional approximation}}$$

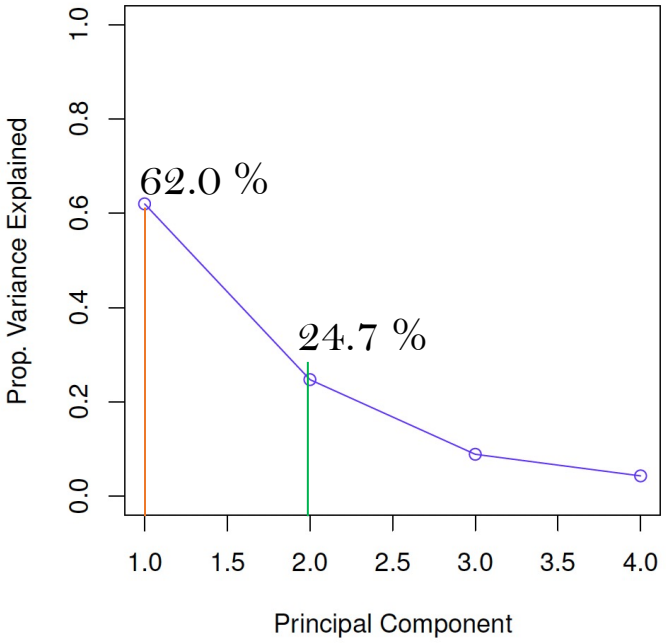
Proportion of Variance

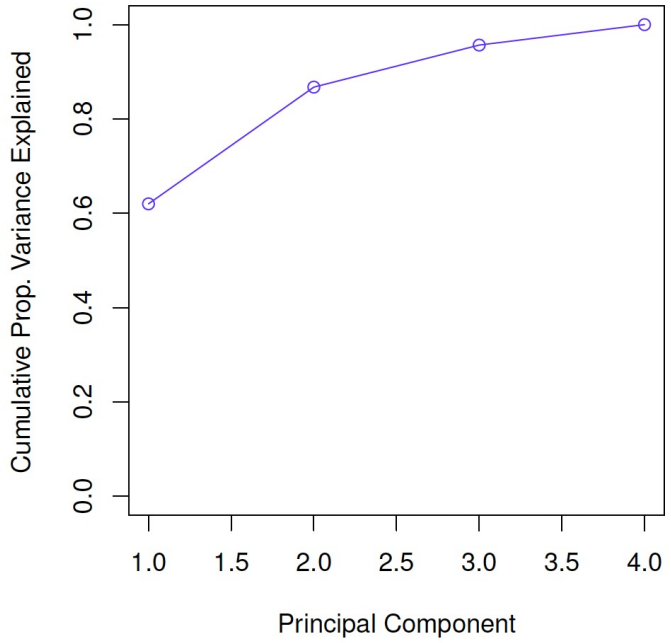
$$\textbf{PVE} = 1 - \frac{\sum_{j=1}^{p} \sum_{i=1}^{n} \left(x_{ij} - \sum_{m=1}^{M} z_{im} \phi_{jm} \right)^2}{\sum_{j=1}^{p} \sum_{i=1}^{n} x_{ij}^2} = 1 - \frac{\text{RSS}}{\text{TSS}}$$

PCA as an R^2 approximation of a data matrix

Proportion of Variance and Scree Plot

Scree Plot: Used to represent Proportion of Variance captured by Each Principal Component





First 2 PCs ~

62.0 + 24.7 =

~ 87% of Variance

Scree Plot for PCA of *USArrest* Data

Principal Component Analysis: Scaling of Variables

Practical Consideration:

- Data scaled to be centered at mean 0 and standard deviation (SD) of 1.
- Each variable should be scaled to have a SD of 1. Else, the representation would be dominated by the variable with maximum variance.

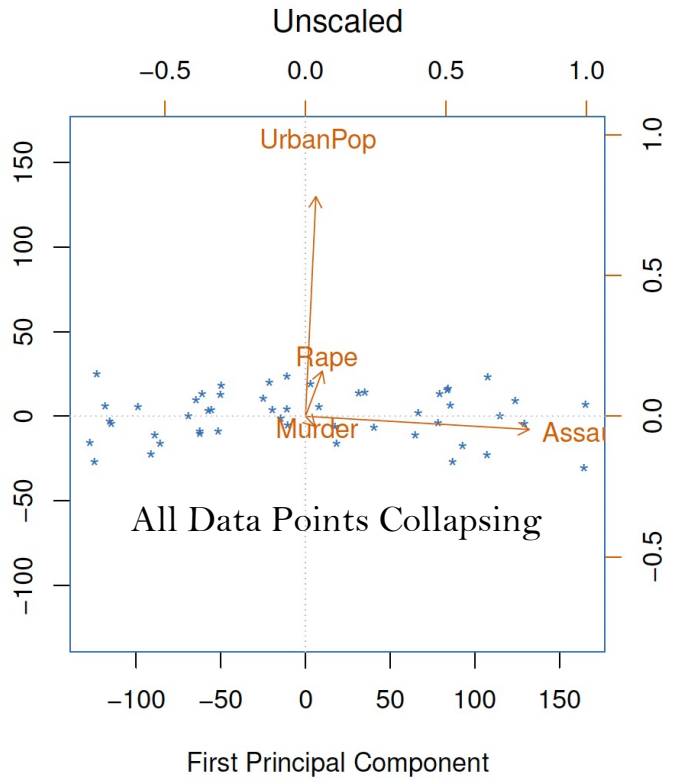
Example: In USArrest Data

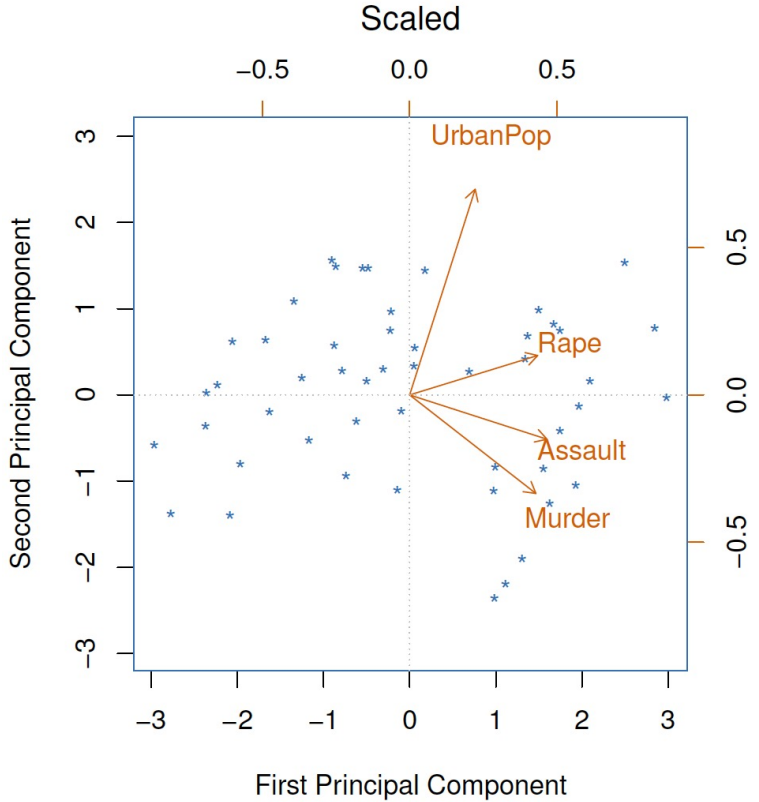
Four variables:	Murder	Rape	Asault	UrbanPop
Variance:	18.97	87.73	6945.16	209.5

Principal Component Analysis: Scaling of Variables

Example: In USArrest Data

Four variables:	Murder	Rape	Asault	UrbanPop
Variance:	18.97	87.73	6945.16	209.5



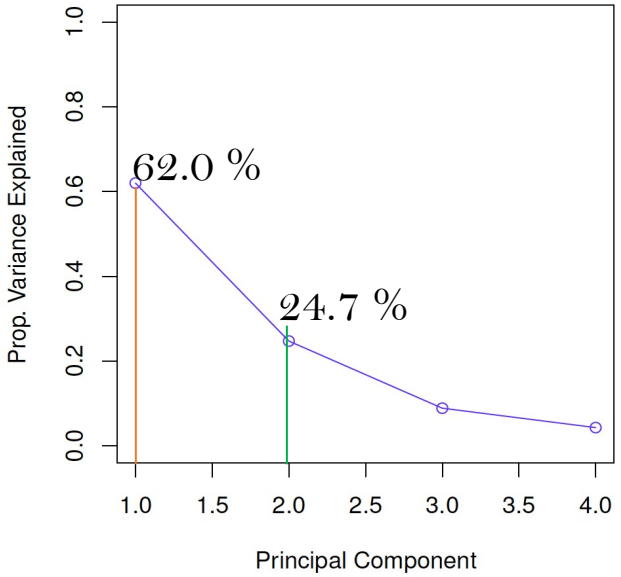


Principal Component Analysis: How Many Components?

For a $n \times p$ data matrix

- $Min(n-1, p)$ distinct Principal Components

  - No Definite way to say how many PCs should we choose.
  - Typically, we take help of scree plot



Example of *USArrest* Dataset

First 2 PCs ~

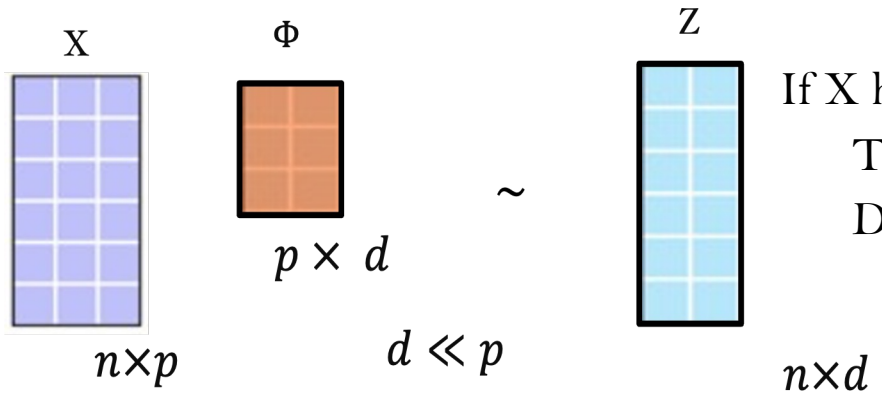
62.0 + 24.7 =

~ 87% of Variance

Principal Components: Other Applications

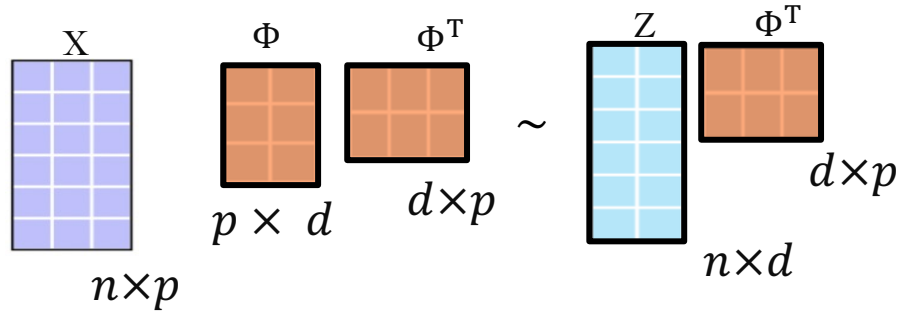
- Missing Data Imputation
- Recommender System

Missing Data Imputation



If X has no missing entries,

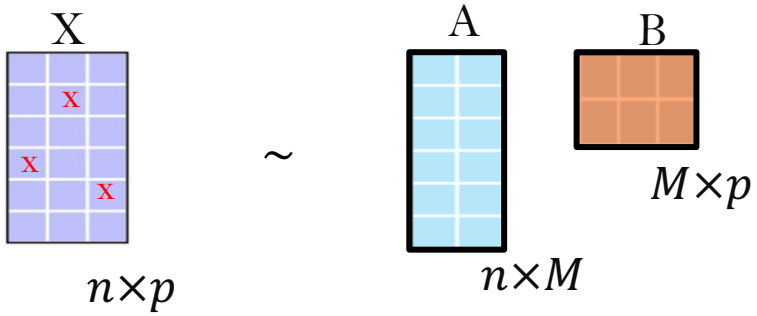
Then Φ can be easily found by Eigen Decomposition or SVD



X can be represented as matrix factorization

Missing Data Imputation

Lets Say we use M Principal Components



X has missing entries denoted by x

O is the set of all obvserved values i.e. indices (i,j) for which X has entries.

Impute Missing Values and Solve PC analysis jointly:

$$\underset{\mathbf{A}\in\mathbb{R}^{n\times M},\mathbf{B}\in\mathbb{R}^{p\times M}}{\text{minimize}} \left\{ \sum_{(i,j)\in\mathcal{O}} \left(x_{ij} - \sum_{m=1}^{M} a_{im} b_{jm} \right)^2 \right\}$$

Missing Data Imputation

$$\underset{\mathbf{A}\in\mathbb{R}^{n\times M},\mathbf{B}\in\mathbb{R}^{p\times M}}{\text{minimize}} \left\{ \sum_{(i,j)\in\mathcal{O}} \left(x_{ij} - \sum_{m=1}^{M} a_{im} b_{jm} \right)^2 \right\}$$

Impute Missing Values and Solve PC analysis jointly ~

- Approximating A and B
- Iterative Solution

Missing Data Imputation

Algorithm 12.1 *Iterative Algorithm for Matrix Completion*

1. Create a complete data matrix $\tilde{\mathbf{X}}$ of dimension $n \times p$ of which the (i, j) element equals

Step 1

$$\tilde{x}_{ij} = \begin{cases} x_{ij} & \text{if } (i,j) \in \mathcal{O} \\ \bar{x}_j & \text{if } (i,j) \notin \mathcal{O}, \end{cases}$$

where $\bar{x}_j$ is the average of the observed values for the jth variable in the incomplete data matrix $\mathbf{X}$. Here, $\mathcal{O}$ indexes the observations that are observed in $\mathbf{X}$.

Set the missing observation as mean of all observation for that feature

Missing Data Imputation

Algorithm 12.1 *Iterative Algorithm for Matrix Completion*

Repeat steps (a)–(c) until the objective fails to decrease:

(a) Solve

Step 2

$$\underset{\mathbf{A}\in\mathbb{R}^{n\times M},\mathbf{B}\in\mathbb{R}^{p\times M}}{\text{minimize}} \left\{ \sum_{j=1}^{p}\sum_{i=1}^{n}\left(\tilde{x}_{ij}-\sum_{m=1}^{M} a_{im}b_{jm}\right)^2 \right\}$$

by computing the principal components of $\tilde{\mathbf{X}}$. Compute Φ or B, Then A

(b) For each element $(i,j)\notin\mathcal{O}$, set $\tilde{x}_{ij}\leftarrow\sum_{m=1}^{M}\hat{a}_{im}\hat{b}_{jm}$.

Update the missing values

(c) Compute the objective

$$\sum_{(i,j)\in\mathcal{O}}\left(x_{ij}-\sum_{m=1}^{M}\hat{a}_{im}\hat{b}_{jm}\right)^2.$$

Compute the objective

Missing Data Imputation

Algorithm 12.1 *Iterative Algorithm for Matrix Completion*

Step 3

Return the estimated missing entries $\tilde{x}_{ij}$, $(i,j) \notin \mathcal{O}$.

Missing Data Imputation: Movie Rating Example

	Jerry Maguire	Oceans	Road to Perdition	A Fortunate Man	Catch Me If You Can	Driving Miss Daisy	The Two Popes	The Laundromat	Code 8	The Social Network	...
Customer 1	•	•	•	•	4	•	•	•	•	•	...
Customer 2	•	•	3	•	•	•	3	•	•	3	...
Customer 3	•	2	•	4	•	•	•	•	2	•	...
Customer 4	3	•	•	•	•	•	•	•	•	•	...
Customer 5	5	1	•	•	4	•	•	•	•	•	...
Customer 6	•	•	•	•	•	2	4	•	•	•	...
Customer 7	•	•	5	•	•	•	•	3	•	•	...
Customer 8	•	•	•	•	•	•	•	•	•	•	...
Customer 9	3	•	•	•	5	•	•	1	•	•	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱

Clustering

K-Means Clustering

- Simple approach to partition data into K different non-overlapping groups
- Specify K in the beginning.

Given n data observation each indexed as $1, 2, 3, \cdots \ n$
Let, $C_1, C_2 \cdots C_K$ denote sets containing indices of observations in each cluster

Then these sets satisfy two properties:

1. $C_1 \cup C_2 \cup \ldots \cup C_K = \{1, \ldots, n\}$. In other words, each observation belongs to at least one of the K clusters.

2. $C_k \cap C_{k'} = \emptyset$ for all $k \neq k'$. In other words, the clusters are non-overlapping: no observation belongs to more than one cluster.

K-Means Clustering

The sets satisfy two properties:

1. $C_1 \cup C_2 \cup \ldots \cup C_K = \{1, \ldots, n\}$. In other words, each observation belongs to at least one of the K clusters.

2. $C_k \cap C_{k'} = \emptyset$ for all $k \neq k'$. In other words, the clusters are non-overlapping: no observation belongs to more than one cluster.

If i^{th} observation is on k^{th} cluster, Then: $i \in C_k$

- K- Means clustering minimizes the *within cluster variation*
- If we represent within cluster variation for cluster C_k as: $W(C_k)$

K-Means Clustering is equivalent to solving:

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} W(C_k) \right\}.$$

K-Means Clustering

K-Means Clustering is equivalent to solving:

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} W(C_k) \right\}.$$

Q. What is the definition of the function $W(\cdot)$

- One most common definition of within cluster variation is *squared Euclidean Distance*

$$W(C_k) = \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2$$

K-Means Clustering is equivalent to solving:

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}$$

K-Means Clustering: Algorithm

Algorithm 12.2 *K-Means Clustering*

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 does it work?

Example:

Measuring tumor size in cancer study

Consider: numbers of observation (n) = 8

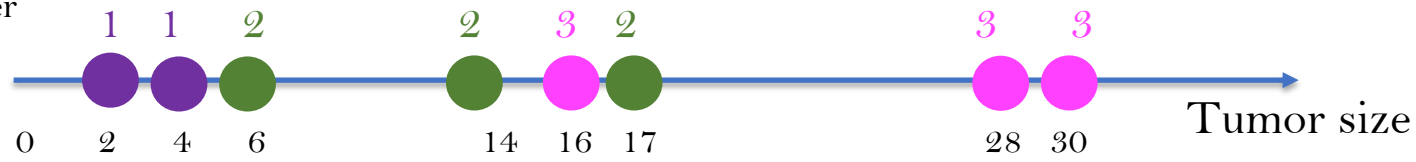
Numbers of features (p) = 1. (Tumor size)

K-Means Clustering: How does it work?

1. Randomly assign a number, from 1 to K, to each of the observations.

Let's consider numbers of clusters K = 3

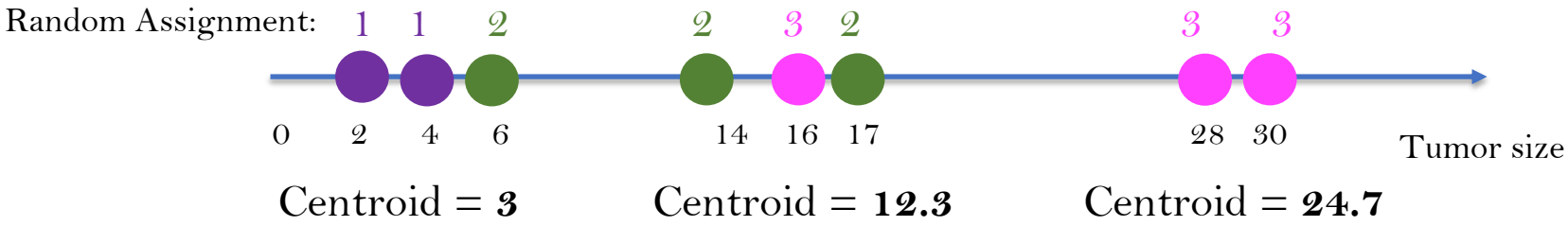
Random Cluster Assignment:



K-Means Clustering: How does it work?

Step 2. a

(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.

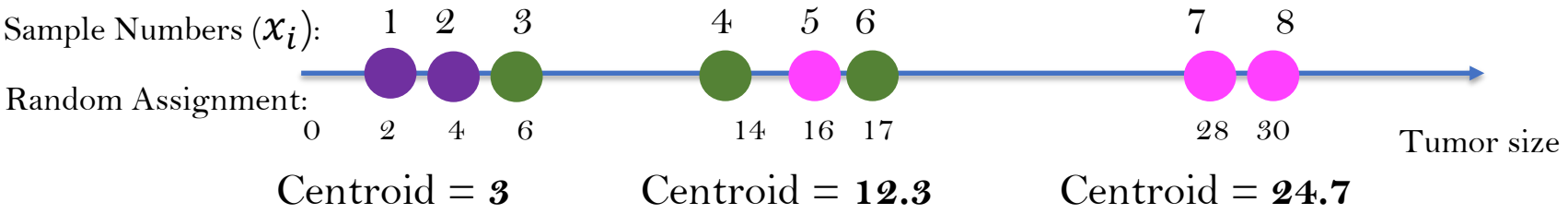


Correction in the video: The centroid calculation error in the video recordings has been corrected here on the slides.

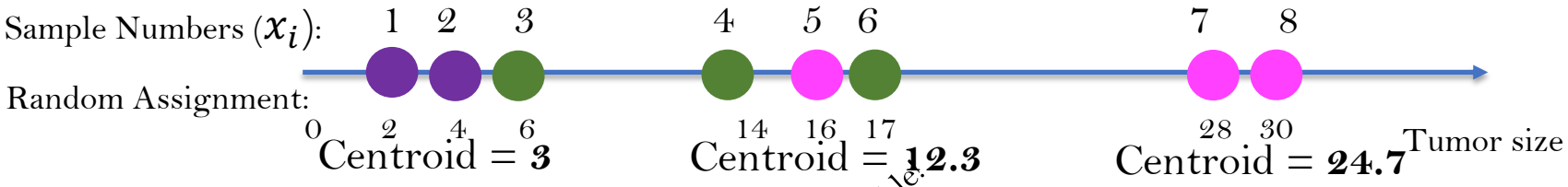
K-Means Clustering: How does it work?

Step 2. b

(b) Assign each observation to the cluster whose centroid is closest (where *closest* is defined using Euclidean distance).



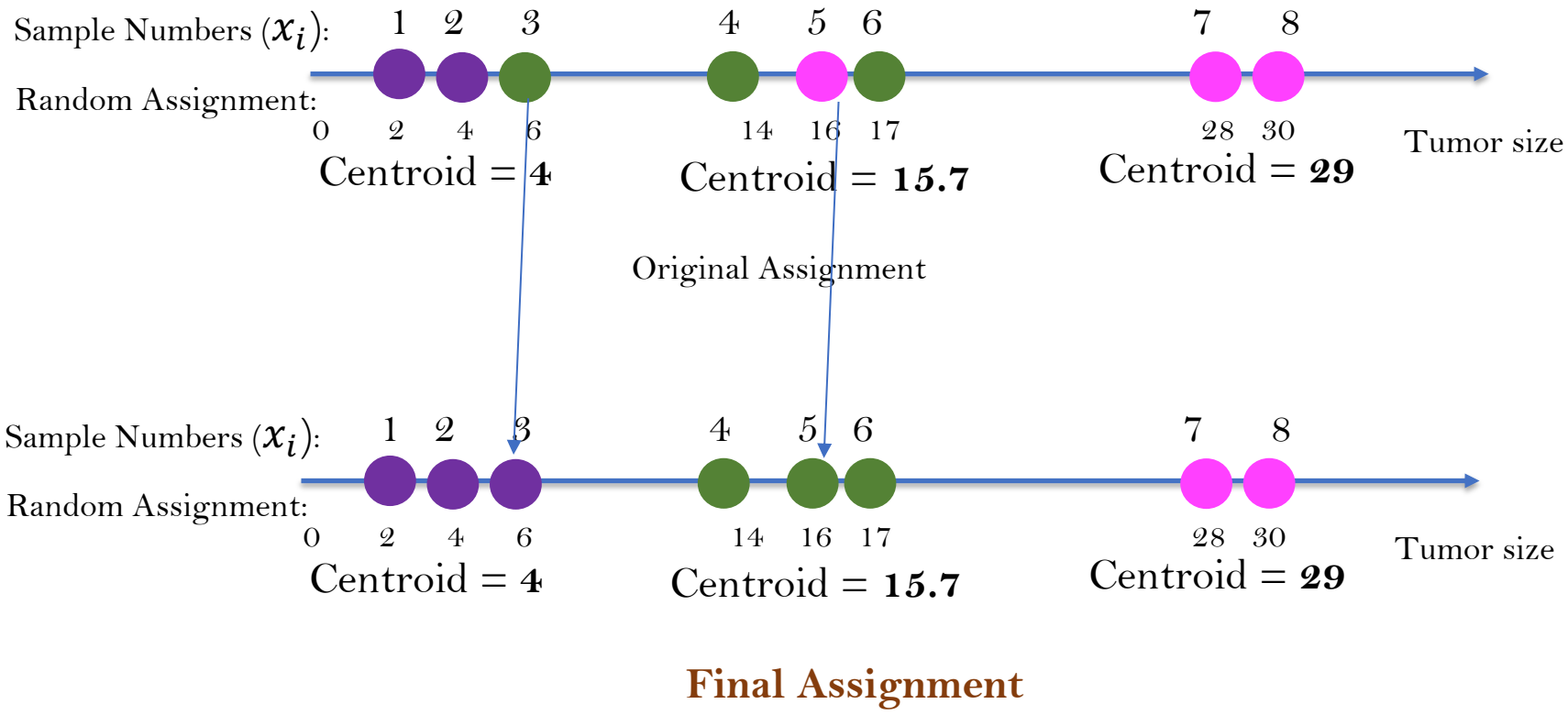
K-Means Clustering: How does it work?



x_i (Tumor size)	Distance from (K = 1)	Distance from (K = 2)	Distance from (K = 3)	Old K	New K
1(2)	1	10.3	22.7	1	1
2(4)	1	8.3	20.7	1	1
3(6)	3	6.3	18.7	2 (wrong)	1 (corrected)
4(14)	11	2.3	10.7	2	2
5(16)	13	4.3	8.7	3(wrong)	2(corrected)
6(17)	13	4.3	7.7	2	2
7(28)	25	16.3	4.7	3	3
8(30)	27	18.7	6.7	3	3

Find two mistakes in the table.

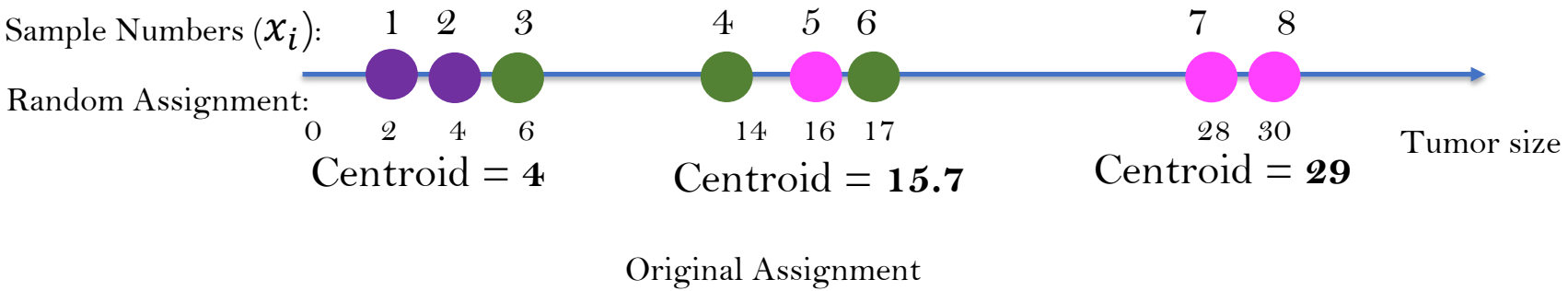
K-Means Clustering: How does it work?



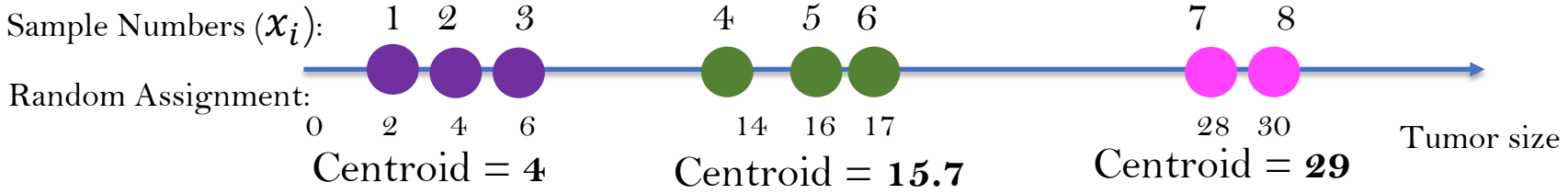
Final Assignment

K-Means Clustering: How does it work?

2. Iterate until the cluster assignments stop changing:



Original Assignment



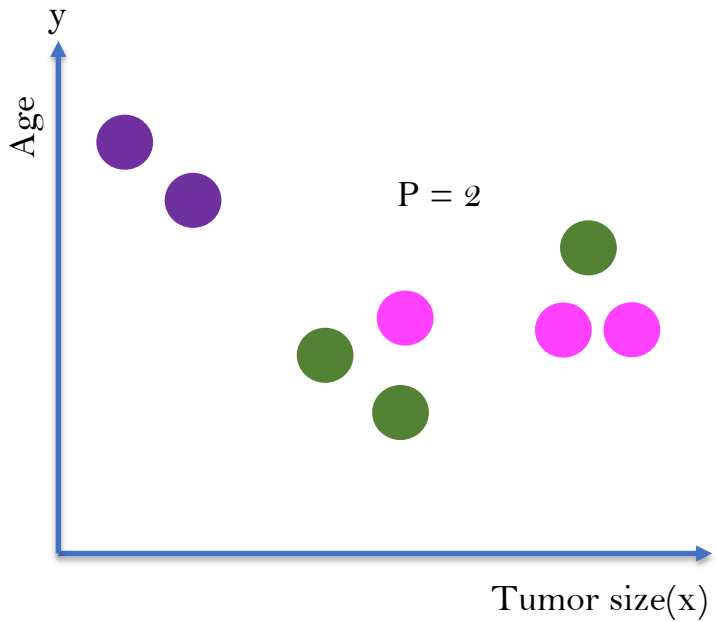
Final Assignment

Repeat 2.a and 2.b, you will see no any assignment of group changes, so stop iteration.

K-Means Clustering: How does it work?

When numbers of feature are large i.e., p is large,

Distance is calculated using Euclidean Distance in the feature space.
Everything else works similarly!



K-Means Clustering: Why the Algorithm works?

Recall:

K-Means Clustering is equivalent to solving:

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}$$

Q. Did we actually do this in the Algorithm?

Q. How assigning an observation to the group with smallest distance from the centroid solves the above problem?

K-Means Clustering: Why the Algorithm works?

Recall:

K-Means Clustering is equivalent to solving:

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}$$

$$\frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 = 2 \sum_{i \in C_k} \sum_{j=1}^{p} (x_{ij} - \bar{x}_{kj})^2$$

$$\bar{x}_{kj} = \frac{1}{|C_k|} \sum_{i \in C_k} x_{ij}$$

Mean of the cluster. That is what we computed to define the centroid.

K-Means Clustering: Choice for K

Objective

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}$$

$$\frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 = 2 \sum_{i \in C_k} \sum_{j=1}^{p} (x_{ij} - \bar{x}_{kj})^2$$

Q. What happens to the objective when K = n?

K-Means Clustering: Choice for K

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}$$

$$\frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 = 2 \sum_{i \in C_k} \sum_{j=1}^{p} (x_{ij} - \bar{x}_{kj})^2$$

0

Q. What happens to the objective when K = n?

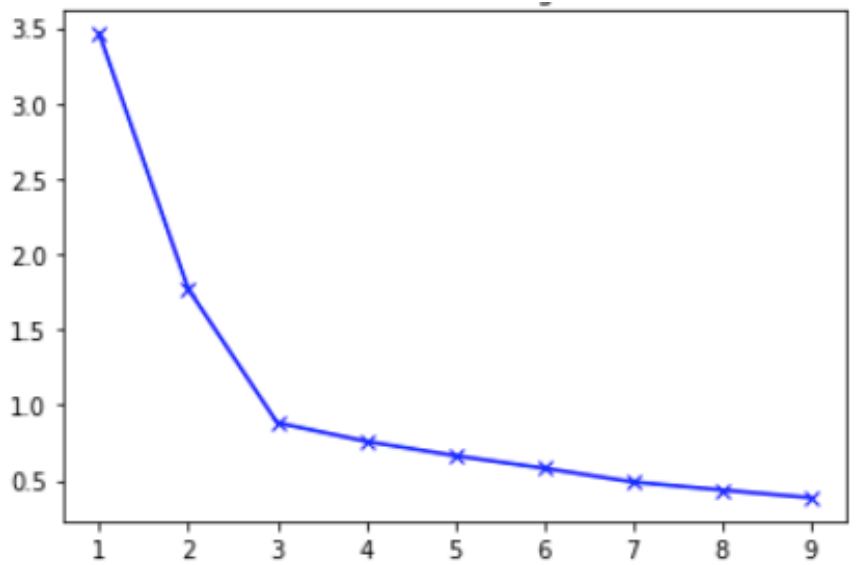
- The objective becomes 0.
- As we increase the K, the objective decreases.

K-Means Clustering: Choice for K

Q. What happens to the objective when K = n?

- The objective becomes 0.
- As we increase the K, the objective decreases.

- Plot K vs. Objective (Sum of *Within Cluster Variance*)
- Choose value of K where slope of curve changes rapidly. (*Elbow method*)



K-Means Clustering: Practical Considerations

- Standardize/Not Standardize the data????
- Solution depends on the initial random assignment. Typically, we do many experiments with random initial assignment, choose the one with minimum objective.
- Solution depends on the perturbation of the data.
  (If we take a subset of the original data and do the K mean clustering, we might get a very different clusters than the original.)
- Not very straightforward to interpret. (But very helpful to design new hypotheses and further investigations.)

K-Means Clustering: Some Examples

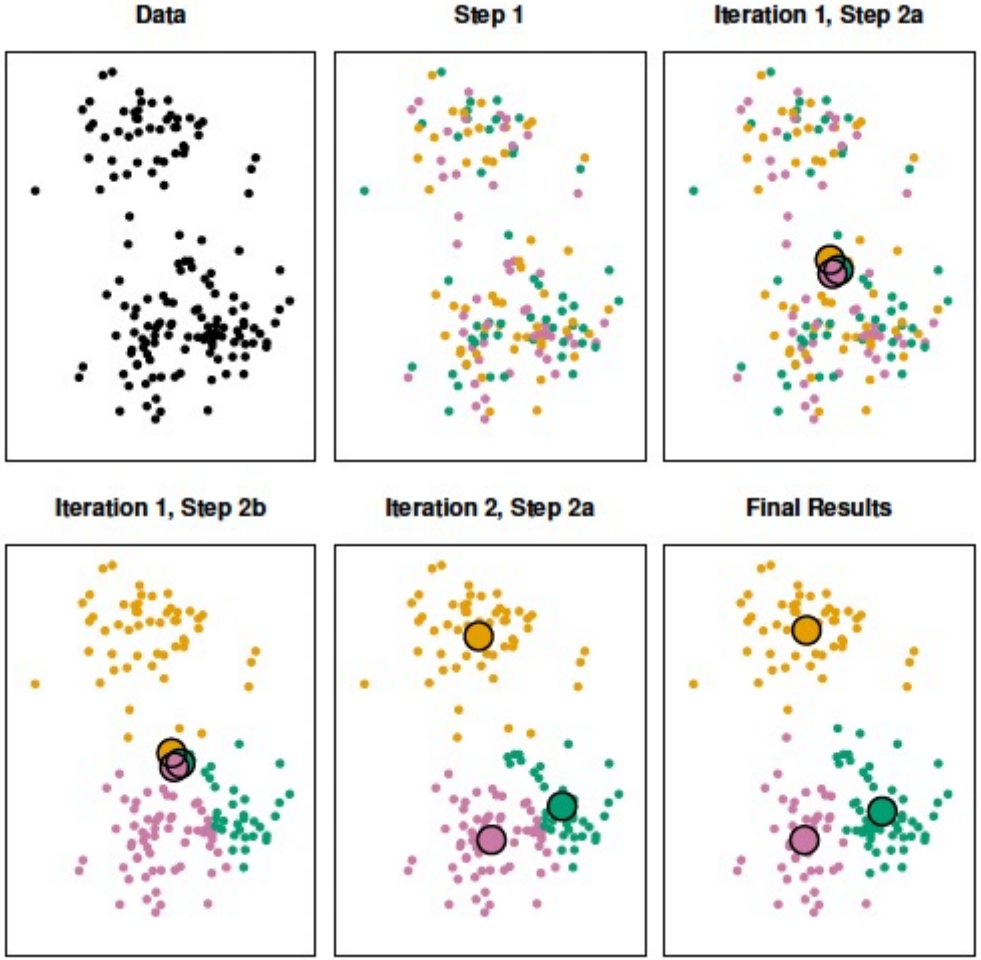


Fig 12.8: Illustration of evolution of clustering with iterations.

K-Means Clustering: Some Examples

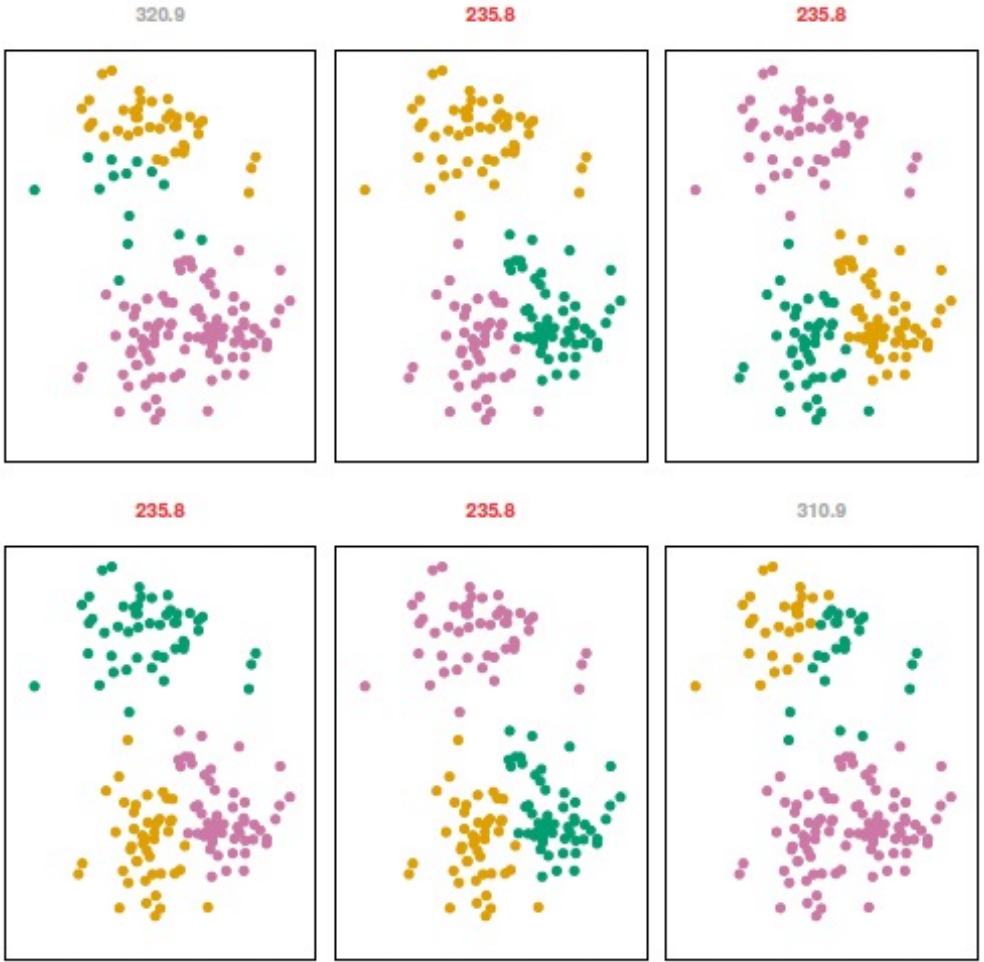
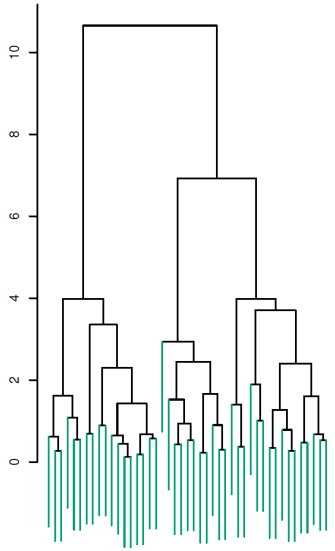


Fig 12.9: Clustering with different random initializations and different final objective values.

Hierarchical Clustering

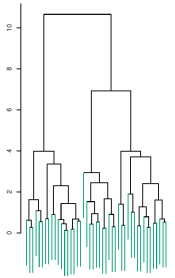


- Eliminates the need to define K beforehand.
- Builds a tree using bottom-up approach.
- At the beginning, each observation is a cluster in itself.
- Two key idea governs tree building process:
- *Dissimilarity Measure* (Distance), *Linkage*
- *Euclidean* Distance is the most commonly used dissimilarity measures.
- Types of Linkages:
- *Complete*, *Single*, *Average*, *Centroid*
- Uses Dendrogram to depict relation between observations.

We will first see:

1. How it works, and
2. How to interpret dendrogram

Hierarchical Clustering: Algorithm



Algorithm 12.3 *Hierarchical Clustering*

1. Begin with n observations and a measure (such as Euclidean distance) of all the $\binom{n}{2} = n(n-1)/2$ pairwise dissimilarities. Treat each observation as its own cluster. (Compute Distances)

2. For $i = n, n-1, \ldots, 2$: (Group from Numbers of Group = n to numbers of groups = 2)

   (a) Examine all pairwise inter-cluster dissimilarities among the i clusters and identify the pair of clusters that are least dissimilar (that is, most similar). Fuse these two clusters. The dissimilarity between these two clusters indicates the height in the dendrogram at which the fusion should be placed.

   (b) Compute the new pairwise inter-cluster dissimilarities among the $i-1$ remaining clusters.

Hierarchical Clustering: Algorithm, Example:

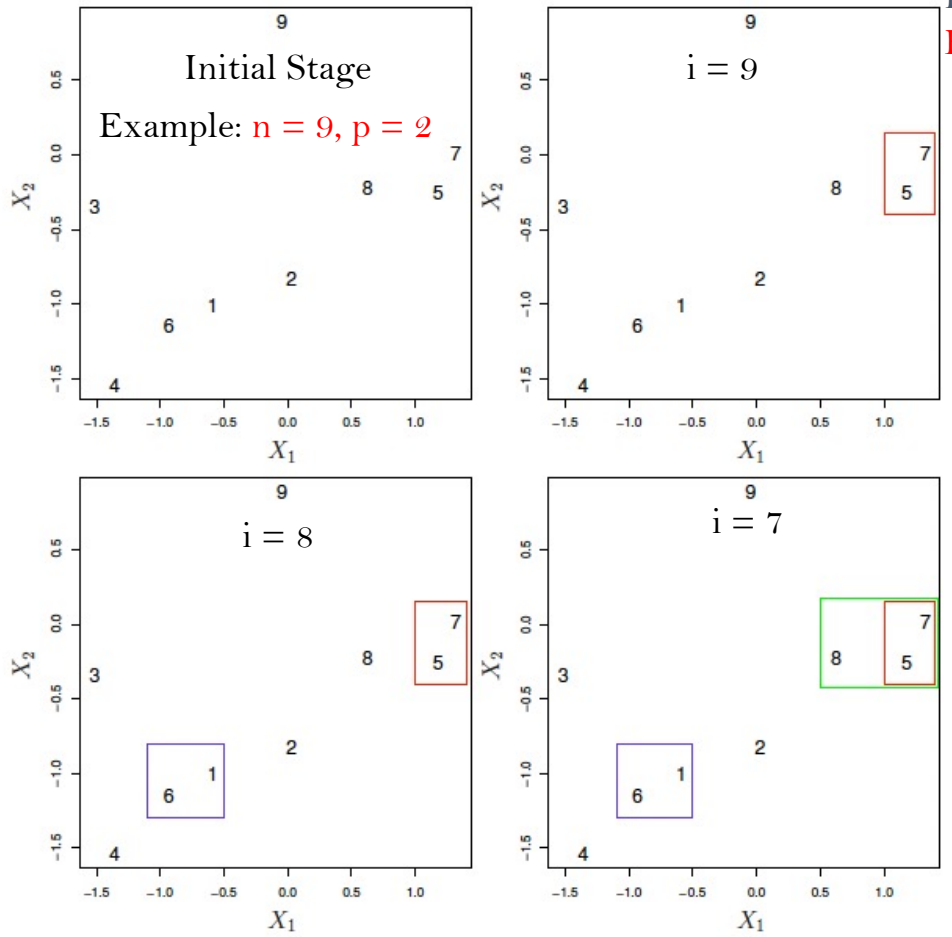


Fig 12.3: Hierarchical Clustering with Euclidean Distance and Complete Linkage

Step i = 9(n): (each observation is in cluster itself)

Compute all distance x_{qk}, q = 1:n, k = 1:n

$x_{5,7}$ is the smallest (observation 5 and 7 are most similar).

Group together observations 5 and 7

Step i= 8:

(5,7) is 1 cluster and remaining each n-1 observation is in a cluster itself.

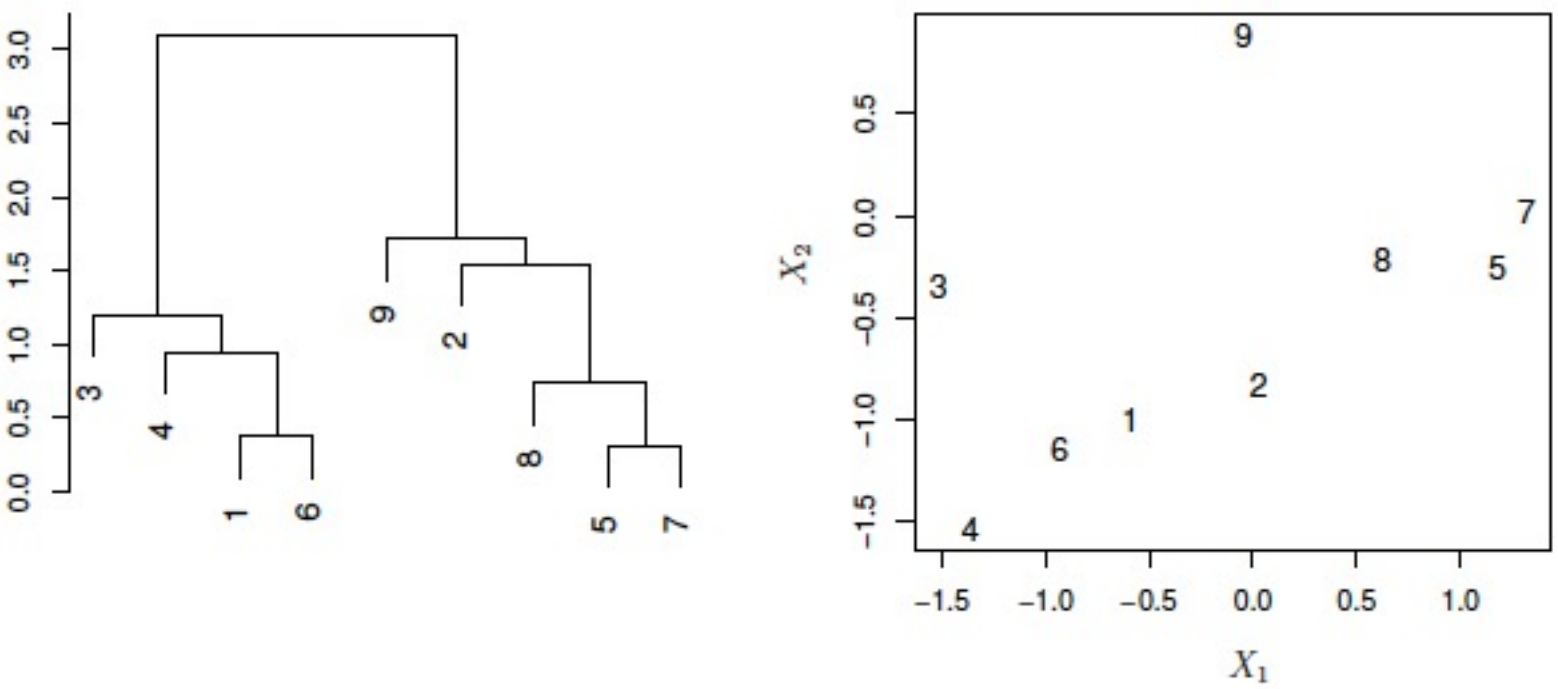
Q. How do we compute new dissimilarities?

This is where linkage plays role, after any two observations are formed. i.e., after i = n step.

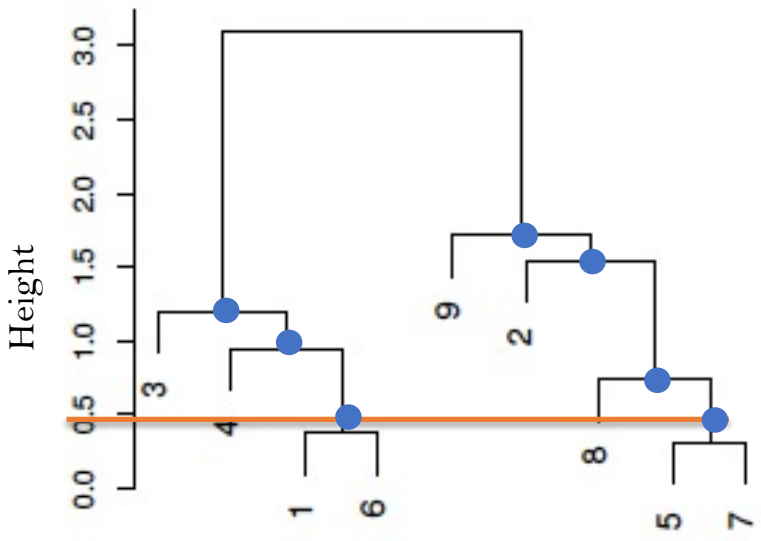
- In complete linkage we would still compute distance between each observation and use the pair observation with minimum distance. (Here, observation 1 and 6).
- In centroid Linkage, we would use centroid of the cluster.

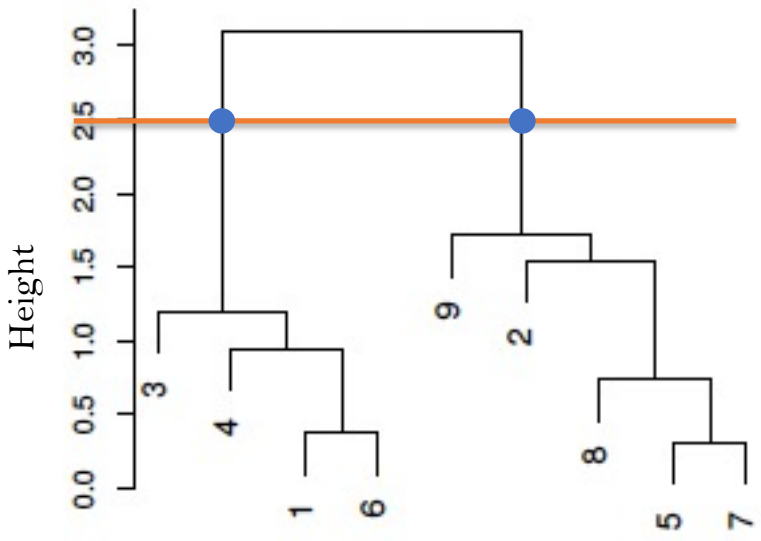
Hierarchical Clustering: Algorithm, Example:

Fig 12.2: Hierarchical Clustering with Euclidean Distance and Complete Linkage



Hierarchical Clustering: Height and Numbers of Clusters

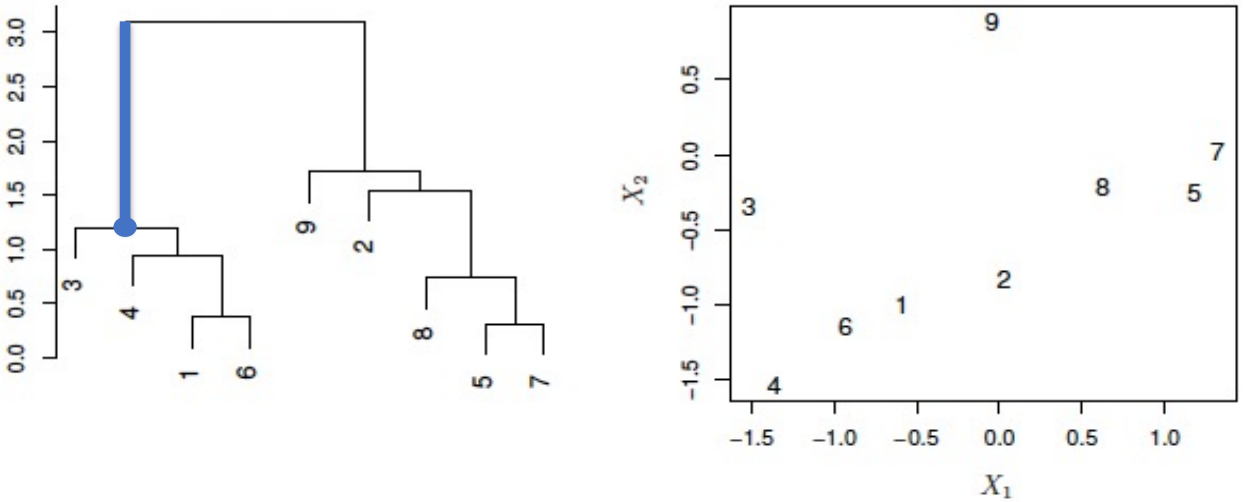




Height = 0.5, Numbers of Clusters = 7.
At this stage, we have only grouped (5,7) and (1,6)

Height = 2.5, Numbers of Clusters = 2. At this stage, we have grouped (3,4,1,6) and (9,2,8,5,7)

Hierarchical Clustering: Dendrogram Interpretation and /Misconception



- Dendrograms allow us to visualize similarity between observations(Samples).
- Misconception: shorter the height between samples in dendrogram, more similar are the samples;
- Misconception example: Since observations 4, is closer to 3 than observations 1 and 6, 4 is more similar to 3 than 1 and 6.
- Fact: It's the node/branch where the observations group together is important to interpret similarity not the height between observations.
- Example: Node 4, 1, 6 all pair/group with node 3 at same branch.(shown shaded in blue). *So, observations 1,6 are no more dissimilar to node 3 than observation 4 to node 3.*
- (Note, in the right figure, this makes more sense.)

Hierarchical Clustering: Different Linkage

Linkage	*Description*
Complete	Maximal intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *largest* of these dissimilarities.
Single	Minimal intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *smallest* of these dissimilarities. Single linkage can result in extended, trailing clusters in which single observations are fused one-at-a-time.
Average	Mean intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *average* of these dissimilarities.
Centroid	Dissimilarity between the centroid for cluster A (a mean vector of length p) and the centroid for cluster B. Centroid linkage can result in undesirable *inversions*.

Hierarchical Clustering: Different Linkage

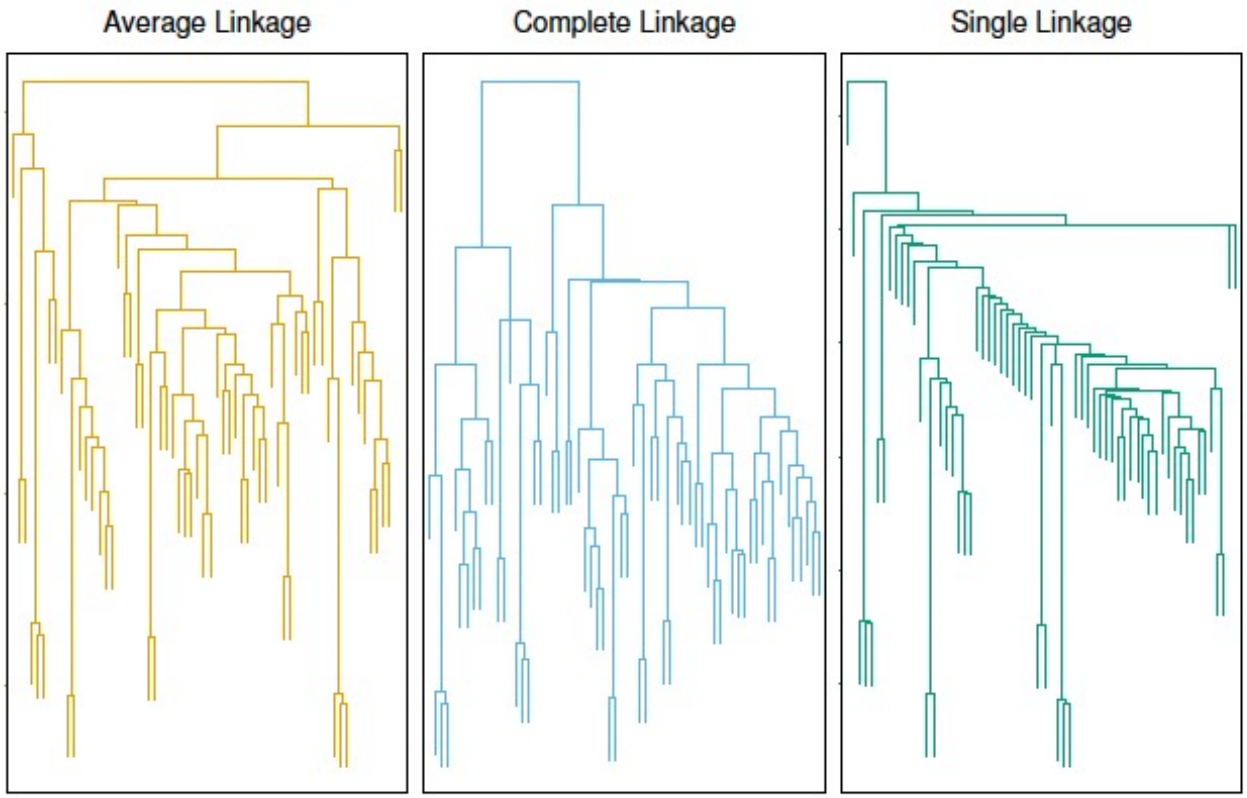


Fig 12.14: Average, Centroid and Complete Linkage gives more balanced clusters than Single.

Hierarchical Clustering: Choice of: Distance/Standardization

- Euclidean is not always optimal.
- Depends on the end goal and purpose of clustering. Sometimes correlation-based distance makes more sense.
- Choice to whether to standardize or not standardize also depends on end goal and objective of clustering.

Example: Euclidean Distance vs Correlation based Distance

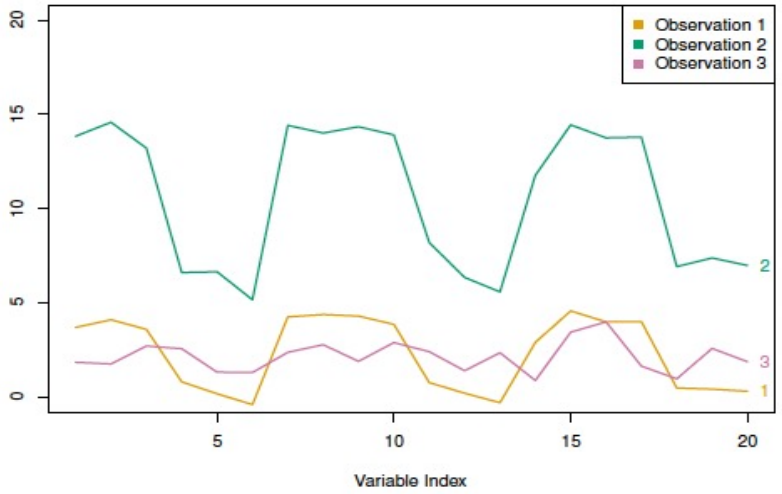


Fig 12.15. n = 3, p = 20.

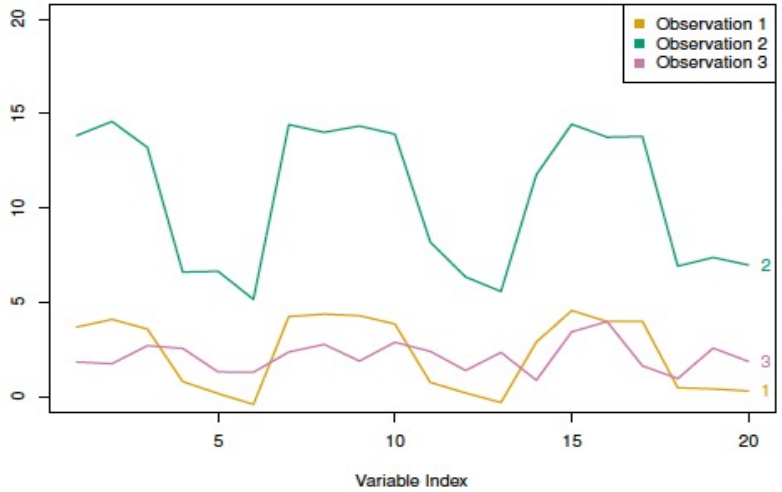
- Think of shopping data.
- Each variable is a different shopping items, example: computers, books, clothes, cosmetics etc…..
- Each observation is a different user/shopper.
- Observations (shopper) 1 and and 3 have almost same values for each variable. So, shopper 1 and shopper 3 are similar in terms of Euclidean distance.
- But shoppers 1 and 2 have similar shape of the curve (*they tend to buy similar things*) but different values for each variables. So, 1 and 2 have high correlation.

Q. What makes sense?

A. Grouping 1 and 3 together based on Euclidean Distance.

B. Grouping 1 and 2 based on Correlation?

Hierarchical Clustering: Choice of: Distance/Standardization



Q. What makes sense?

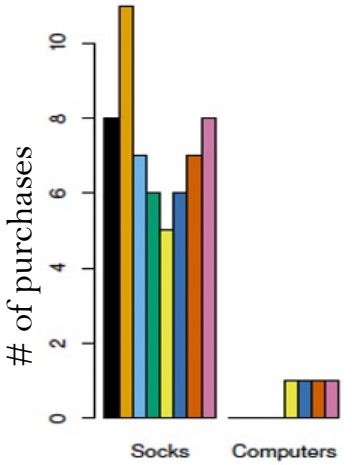
Answer: B. Grouping 1 and 2 based on Correlation if we want to targeted advertisement/recommendation.

But, it might also make sense to do Euclidean distance if we want to study characteristics of shoppers who have very low expenditure on shopping and want to study how can we increase their expenditure habit?

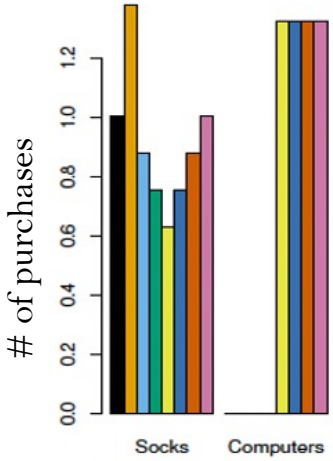
Hierarchical Clustering: Choice of: Distance/Standardization

- Choice to whether to standardize or not standardize also depends on end goal and objective of clustering.

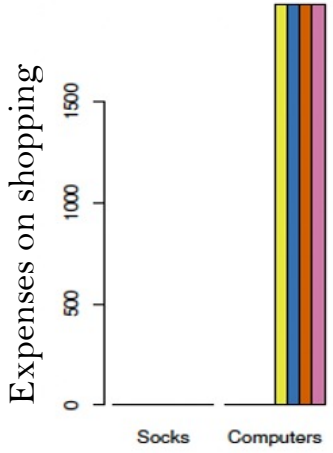
  Example: Shoppers Data, Fig 12.16. n = 8 (shoppers), p = 2(socks and computer).



No Standardization:
Socks would heavily determine Euclidean Distance

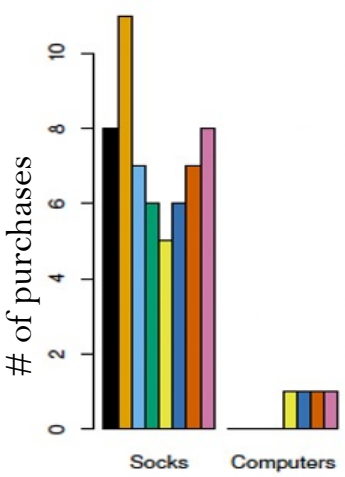


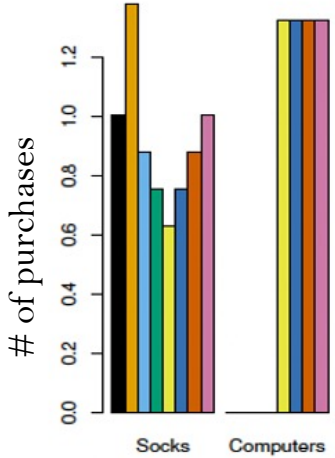
Standardization:
Socks and Computers would have similar contribution on Euclidean Distance.
Does this make sense?
Maybe we want to have computers dominate the grouping.

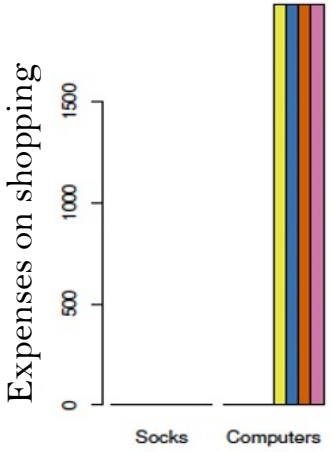


No Standardization but change numbers of purchases to total expense by the user.
This would make more sense because now people who spend more $$ on shopping are grouped together and we can make advertisement decision based on that.

Hierarchical Clustering: Choice of: Distance/Standardization







- The discussion of Standardization applies to K-Means clustering and PCA analysis as well.
- Discussion on choice of distance applies to K-means clustering as well not only to hierarchical.

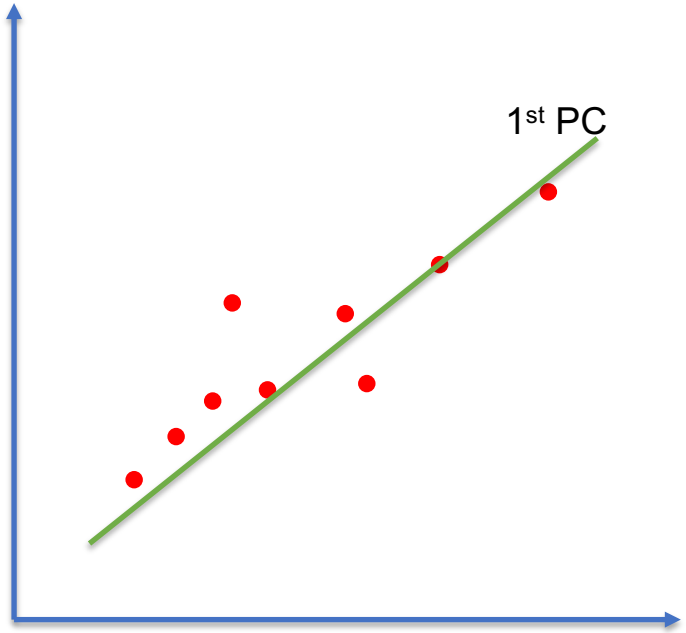
UNIVERSITY OF ARKANSAS

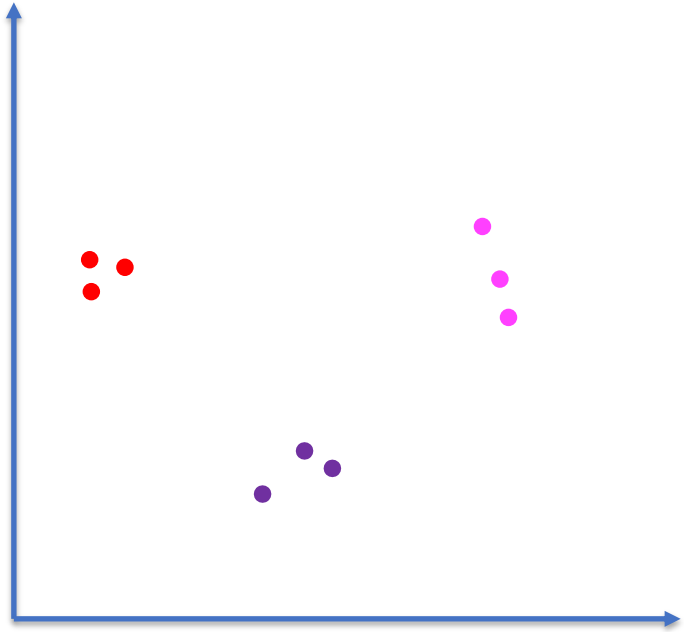
T-SNE visualization

Some content on this part of slides are taken from:

1. Dimitry Kobak, Tubingen University

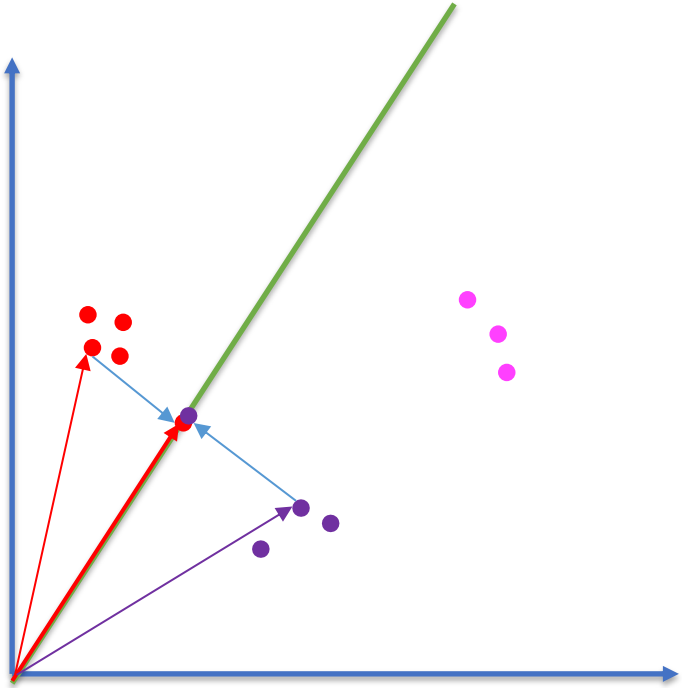
Dimensionality Reduction





- Principal Components is probably not the best approach for the dimension reduction for data like this

Dimensionality Reduction

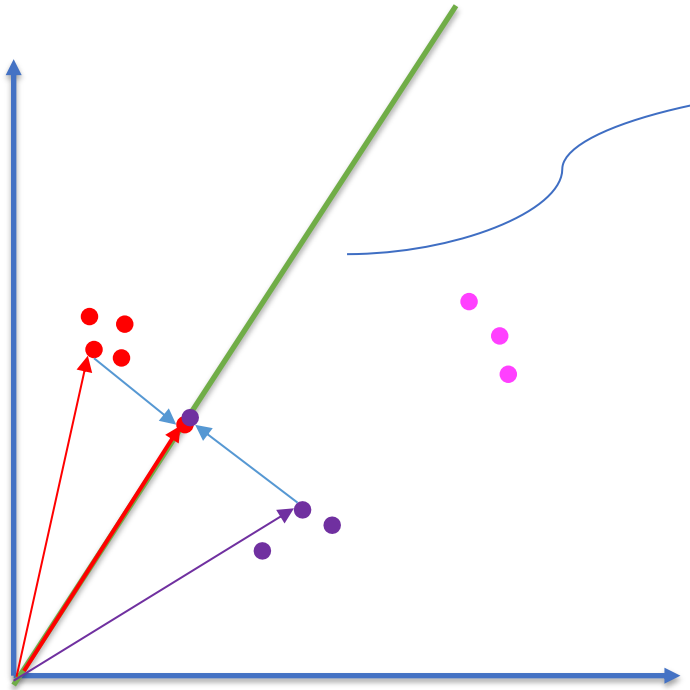


- PC is not a good choice

- What we might want!

- What we get!

Dimensionality Reduction: Pairwise Distance Preservation



High Dimension

Low Dimension

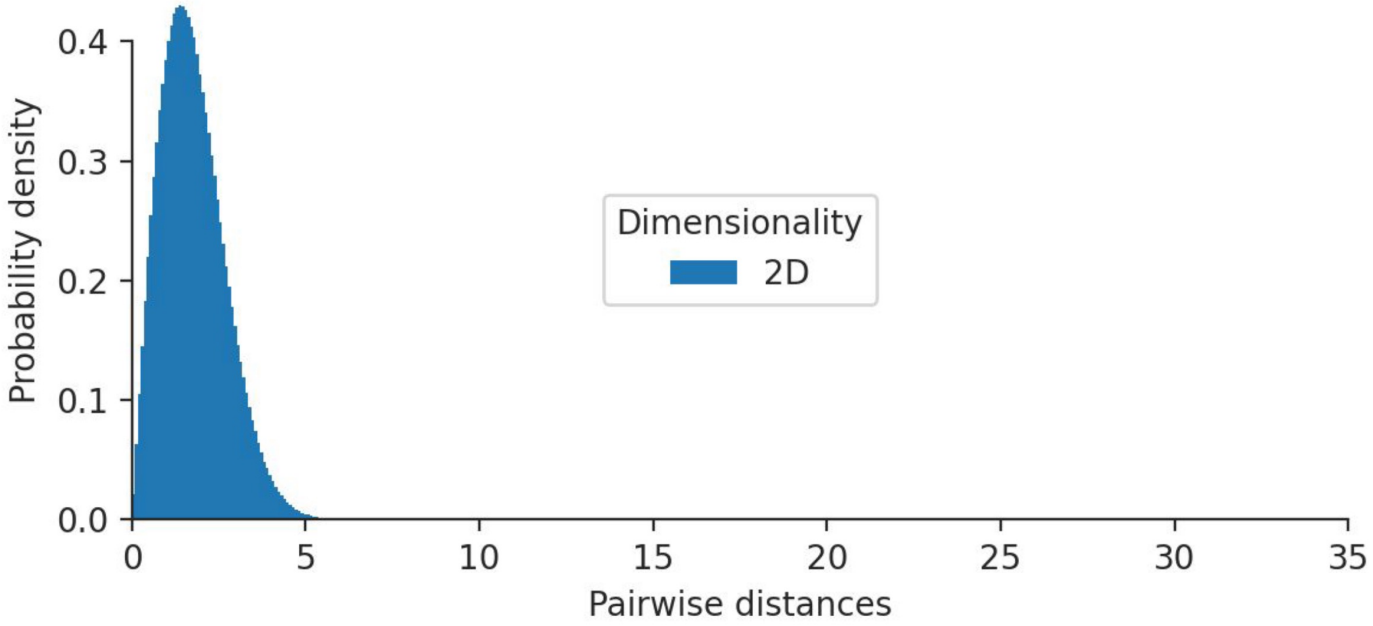
Pair wise Distance Preserving Data Embedding
(Dimensionality Reduction)
Example: *Multidimensional Scaling*

Minimize:

$$\mathcal{L} = \sum_{i<j} (d_{ij} - \|\mathbf{y}_i - \mathbf{y}_j\|)^2$$

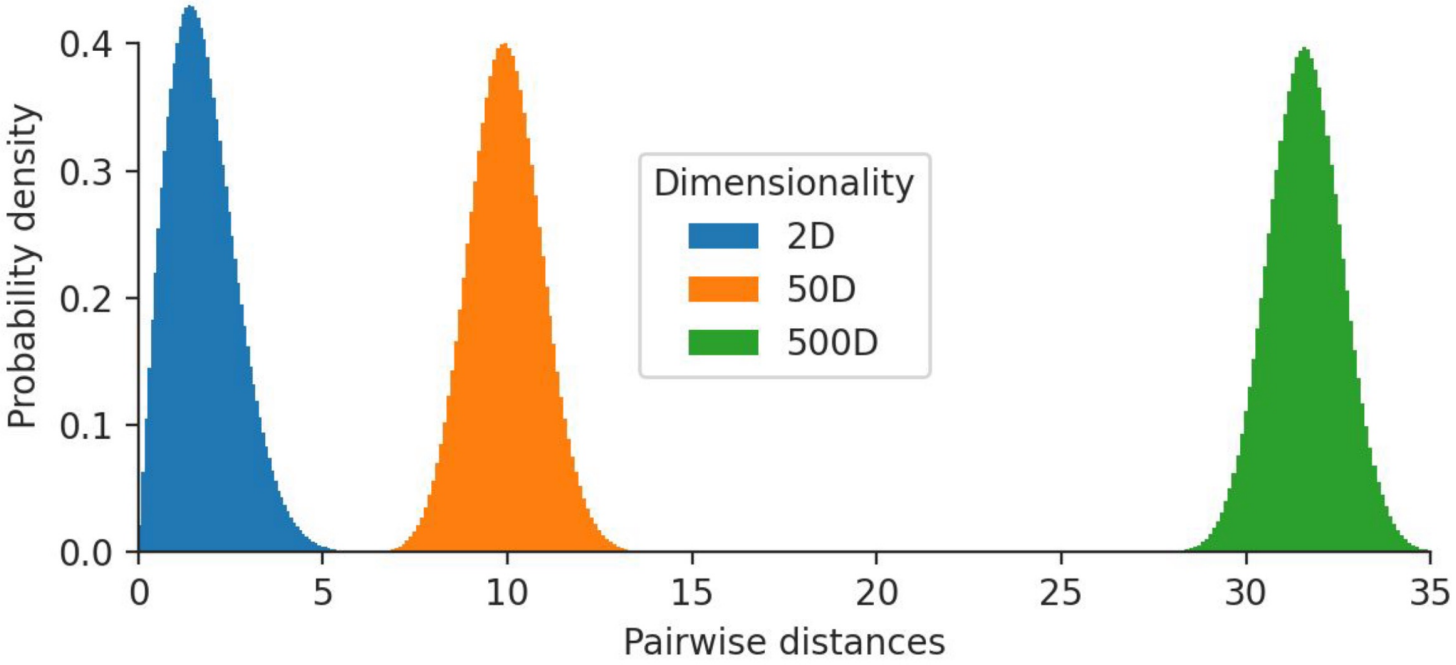
d_{ij}: *Distance between data points in High Dimension*
y_i: *Representation of Data Points in Low Dimension*

Pairwise Distance Preservation: Is it a good idea?



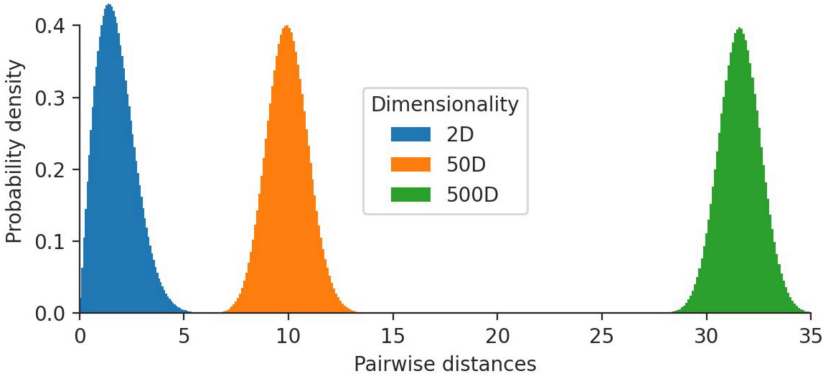
Random Data Generation and Probability Density of pair wise distances in 2 D.

Pairwise Distance Preservation: Is it a good idea?



Random Data Generation and Probability Density of pair wise distances. When D = 2, D = 50 and D = 500

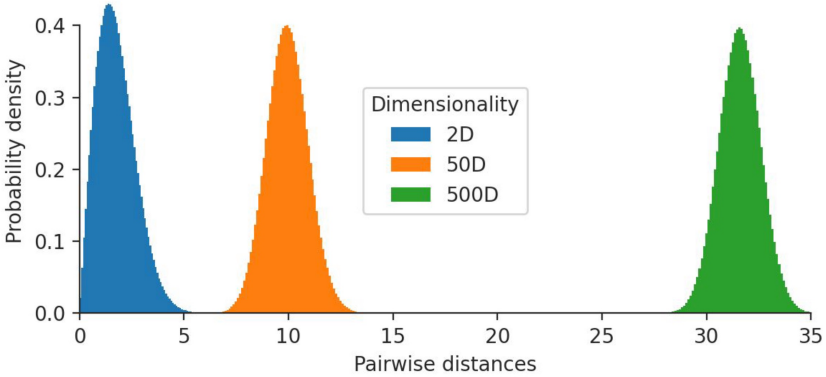
Pairwise Distance Preservation: Is it a good idea?



Case 1; D = 2, and we generated 5 random points
Case 2; D = 500, and we generated 5 random points again.

Q. Which case would have more probability that the 5 data points have smaller pairwise distance?

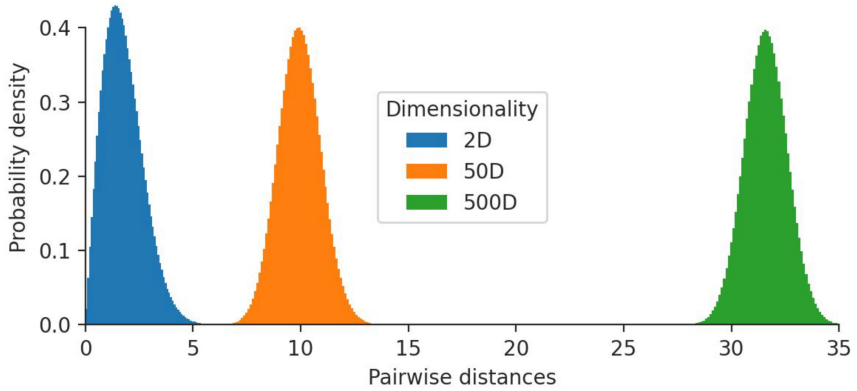
Pairwise Distance Preservation: Is it a good idea?



Case 1; D = 2, and we generated 5 random points
Case 2; D = 500, and we generated 5 random points again.

Q. Which case would have higher probability that the 5 data points have smaller pairwise distance?
Answer: Case 1

Pairwise Distance Preservation: Is it a good idea?



Random Data Generation and Probability Density of pair wise distances. When D = 2, D = 50 and D = 500

Curse of Dimensionality!

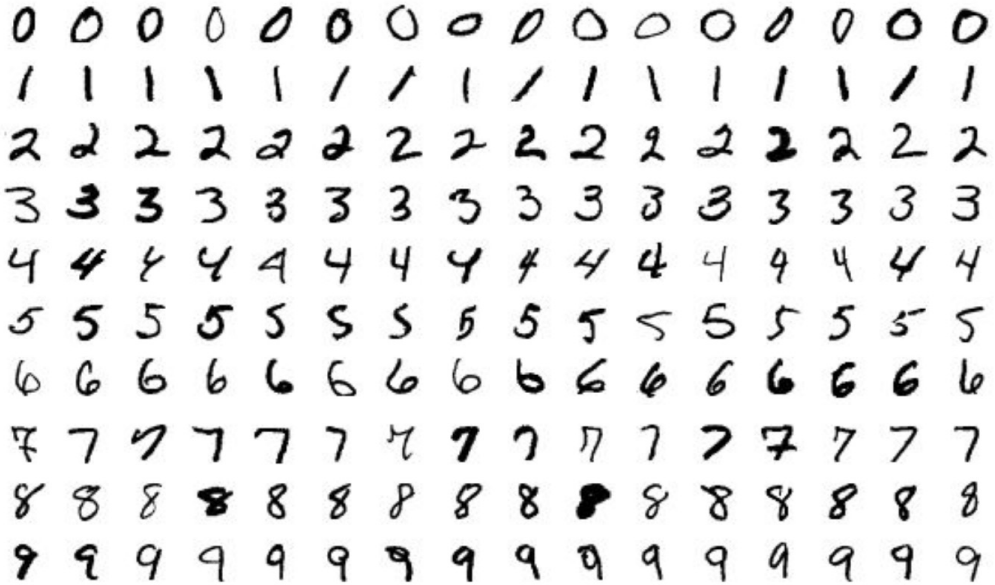
- When we increase the original dimension D, the numbers of data points with small pairwise distance decreases.
- In other words, there is very small/weak global structure.
- Instead, we might want to think about the data *locally*!

Dimensionality Reduction: Neighborhood Preserving Embedding

Motivation:

When we increase the original dimension D, preserving global pairwise-distances is not necessarily a good idea.
Instead, we might want to preserve neighborhood.

Example: Visualization of MNIST dataset



n = 70,000 (70,000 images/Data Points)
D = 28x28 = 784 pixels (size of each image)

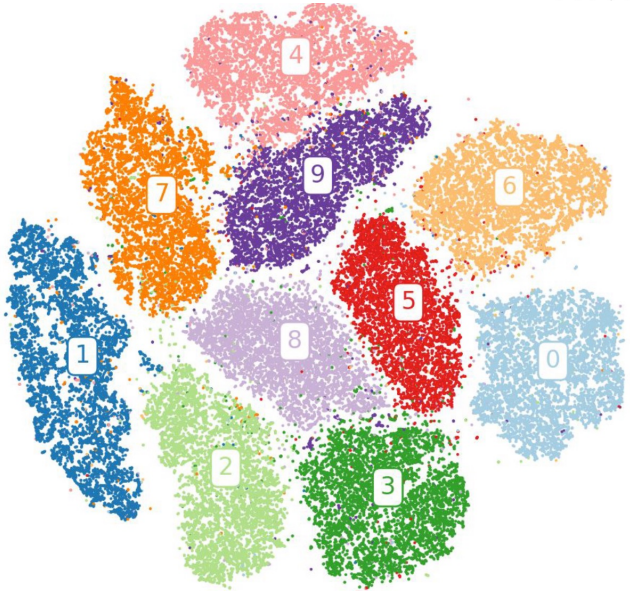
Goal:

Represent: D = 28x28 size datapoints (images) in 2 dimension (d = 2) such that:
Same Digits images are close to each other (i.e., they are neighbors.)

Dimensionality Reduction: Neighborhood Preserving Embedding

Example: Visualization of MNIST dataset

Visualization in 2D using Distance Preserving Embedding (MDS)



Visualization in 2D using Neighborhood Preserving Embedding (t-SNE)

t- distribution Stochastic Neighborhood Embedding

Dimensionality Reduction: Neighborhood Preserving Embedding

Stochastic Neighborhood Embedding: Main Idea

Pairwise Distance Preserving Embedding

Minimize:

$$\mathcal{L} = \sum_{i<j} (d_{ij} - \|\mathbf{y}_i - \mathbf{y}_j\|)^2$$

Stochastic Neighborhood Embedding

Minimize:

$$\mathcal{L} = \sum_{i,j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

KL (Kullback-Leibler) Divergence

Dimensionality Reduction: Neighborhood Preserving Embedding

Stochastic Neighborhood Embedding: Main Idea

Minimize:

$$\mathcal{L} = \sum_{i,j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

- p_{ij} =Probability of a data points j being neighbor of data point i, in high dimensional space.
- q_{ij} =~in a low dimensional space.

- KL (Kullback-Leibler) Divergence: Measures the similarity between two distributions.

$$D_{\mathrm{KL}}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \log\left(\frac{P(x)}{Q(x)}\right).$$

- KL Divergence is not symmetric

$$D_{\mathrm{KL}}(P \parallel Q) \neq D_{\mathrm{KL}}(Q \parallel P)$$

Dimensionality Reduction: Neighborhood Preserving Embedding

Minimize:

$$\mathcal{L} = \sum_{i,j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

$$D_{\mathrm{KL}}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \log\left(\frac{P(x)}{Q(x)}\right).$$

KL Divergence: Interpretation

- Relative Entropy from Q to P.
- If we use representation P instead of Q, how much is the information gain.
- In our case of dimension reduction, we can think of this as:
- How much information do we gain using High Dimensional Representation instead of Low dimensional Representation?
- So, minimizing KL divergence from Q to P is equivalent to our goal:
- i.e., Find low dimensional representation such that the information is preserved.

Dimensionality Reduction: Neighborhood Preserving Embedding

Stochastic Neighborhood Embedding: Main Idea

Recall, We want to Minimize:

$$\mathcal{L} = \sum_{i,j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

- p_{ij} =Probability of data points j being neighbor of data point i, in high dimensional space.
- q_{ij} =~in a low dimensional space.

p_{ij}, q_{ij} : How are these values calculated?

Dimensionality Reduction: Neighborhood Preserving Embedding

Stochastic Neighborhood Embedding: Main Idea

Minimize:

$$\mathcal{L} = \sum_{i,j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

p_{ij}, q_{ij} : How are these values calculated?

Similarities in high-dimensional space is defined as:

$$p_{j|i} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2 / 2\sigma_i^2)}$$

One of many different ways, t-SNE uses this definition.

Dimensionality Reduction: Neighborhood Preserving Embedding

$$\mathcal{L} = \sum_{i,j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

Similarities in high-dimensional space is defined as:

$$p_{j|i} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2 / 2\sigma_i^2)}$$

Numerator:

Measure of similarity (*Attractive Force*)between data point i and j. What is the likeliness that data point j is a neighbor of data point i?

Denominator:

Dissimilarity (*Repulsive Force*) between other data points and i. What is the likeliness of other data points being not neighbor of i?.

Dimensionality Reduction: Neighborhood Preserving Embedding

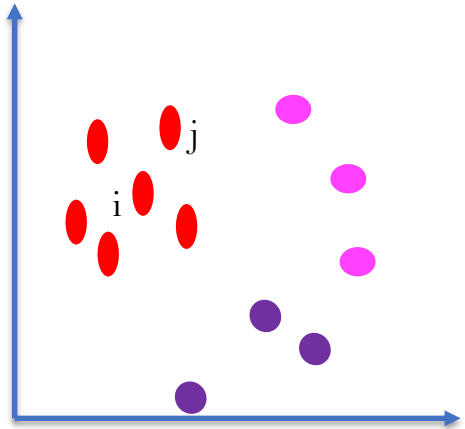
Similarities in high-dimensional space is defined as:

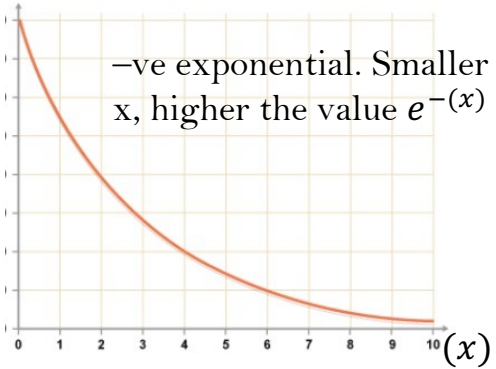
$$p_{j|i} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma_i^2)}{\sum_{k\neq i}\exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2/2\sigma_i^2)}$$

Numerator:

If data points i and j are similar, $\|x_i - x_j\|$ is a small value. –ve exponential of a small value is large. i and j are similar. Will try to increase the $p_{j|i}$.

Denominator : ~. but since its in denominator, large value will try to decrease the $p_{j|i}$.





Dimensionality Reduction: Neighborhood Preserving Embedding

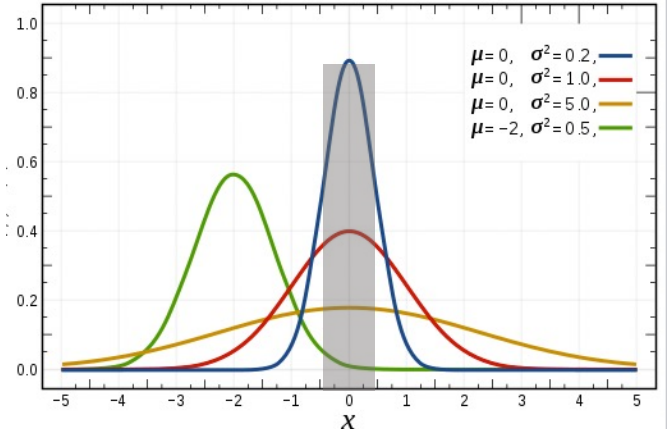
Similarities in high-dimensional space is defined as:

$$p_{j|i} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2 / 2\sigma_i^2)}$$

What about the σ_i ?, What does it do?

- Determines how dense the neighborhood of node i is.
- Larger the sigma less dense is the neighborhood and vice versa.
- Note: $p_{j|i}$ uses Gaussian kernel, so σ_i actually controls the width of a Gaussian.
- Termed, *Perplexity*.

Q. If you count numbers of data samples in the shaded region for each dataset with yellow, green and blue density function, which one would have more samples in that region?



Dimensionality Reduction: Neighborhood Preserving Embedding

Recall our Loss function:

$$\mathcal{L} = \sum_{i,j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

$$p_{j|i} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2 / 2\sigma_i^2)}$$

We defined $p_{j|i}$, but we need p_{ij}

p_{ij} =Probability of data points j being neighbor of data point i, in high dimensional space.

Dimensionality Reduction: Neighborhood Preserving Embedding

Recall our Loss function: $\mathcal{L} = \sum_{i,j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$

$$p_{j|i} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2 / 2\sigma_i^2)}$$

We defined $p_{j|i}$, but we need p_{ij}

p_{ij} =Probability of data point j being neighbor of data point i, in a high dimensional space.

We want p_{ij} to have two properties:

1. It sums to 1. (because its probabilities.)
2. We want it to be symmetric, (note, $p_{i|j} \neq p_{j|i}$)

i.e., $p_{ij} = p_{ij}$ i.e., if j is the neighbor of i, then i is also the neighbor of j.

So, we average $p_{j|i}$ and $p_{i|j}$ and normalize to get p_{ij}

$$p_{ij} = \frac{p_{i|j} + p_{j|i}}{2n}$$

Dimensionality Reduction: Neighborhood Preserving Embedding

Recall our Loss function: $\mathcal{L} = \sum_{i,j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$

$$p_{j|i} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2 / 2\sigma_i^2)}$$

Let's Look into the Denominator now:

Similar arguments:
But since its in denominator, large value of this term would decrease p_{ij}, essentially suggesting the data points are not neighbors. (Repulsive force)

Dimensionality Reduction: Stochastic Neighborhood Embedding

Recall our Loss function:

$$\mathcal{L} = \sum_{i,j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

$$p_{j|i} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2 / 2\sigma_i^2)}$$

Similarity in the lower dimension: q_{ij}

$$q_{ij} = \frac{w_{ij}}{Z}, \quad w_{ij} = k(\|\mathbf{y}_i - \mathbf{y}_j\|), \quad Z = \sum_{k \neq l} w_{kl}$$

k is some kernel.

Dimensionality Reduction: Stochastic Neighborhood Embedding

Recall our Loss function: $\mathcal{L} = \sum_{i,j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$

$$q_{ij} = \frac{w_{ij}}{Z}, \quad w_{ij} = k(\|\mathbf{y}_i - \mathbf{y}_j\|), \quad Z = \sum_{k \neq l} w_{kl}$$

k is some kernel.

In SNE paper: $k(d) = \exp(-d^2)$

In t- SNE paper: $k(d) = 1/(1 + d^2)$

Dimensionality Reduction: t-SNE

Recall our Loss function: $\mathcal{L} = \sum_{i,j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$

$$\mathcal{L} = -\sum_{i,j} p_{ij} \log q_{ij} = -\sum_{i,j} p_{ij} \log \frac{w_{ij}}{Z}$$

$$= -\sum_{i,j} p_{ij} \log w_{ij} + \log \sum_{i,j} w_{ij},$$

Dimensionality Reduction: t-SNE

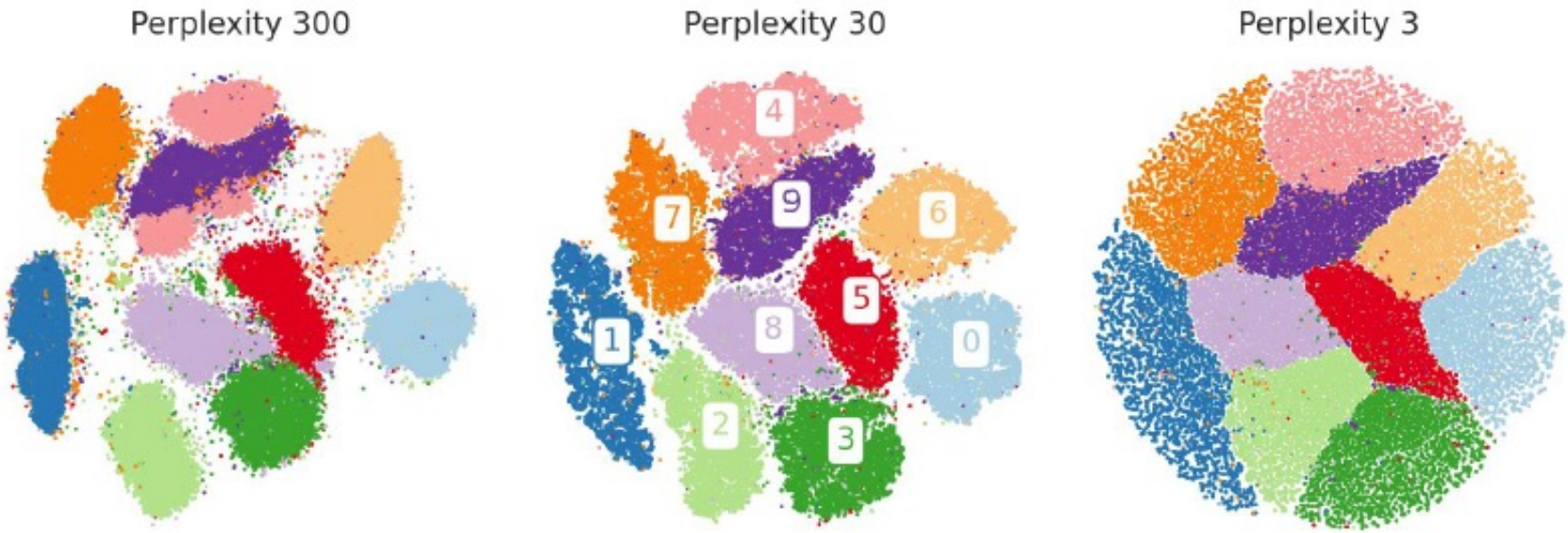
Recall our Loss function: $\mathcal{L} = \sum_{i,j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$

Loss Function and Gradient Descent

$$\mathcal{L} = -\sum_{i,j} p_{ij} \log q_{ij} = -\sum_{i,j} p_{ij} \log \frac{w_{ij}}{Z}$$

$$= -\sum_{i,j} p_{ij} \log w_{ij} + \log \sum_{i,j} w_{ij},$$

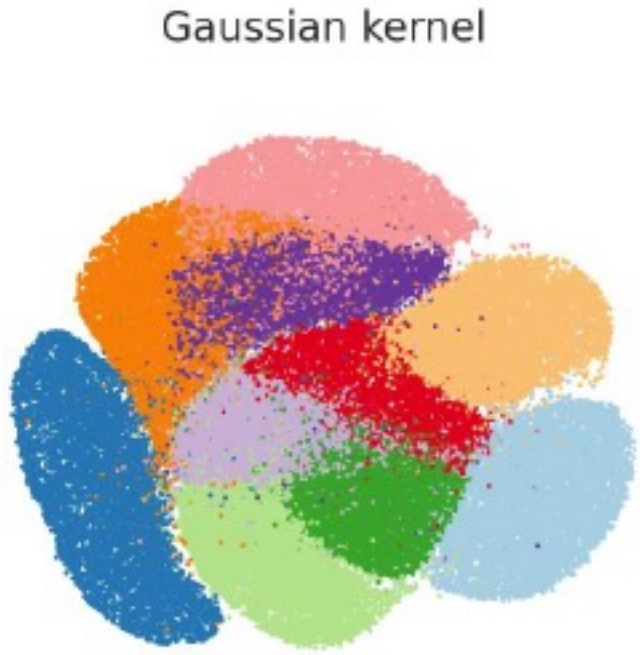
t-SNE Visualization: Examples



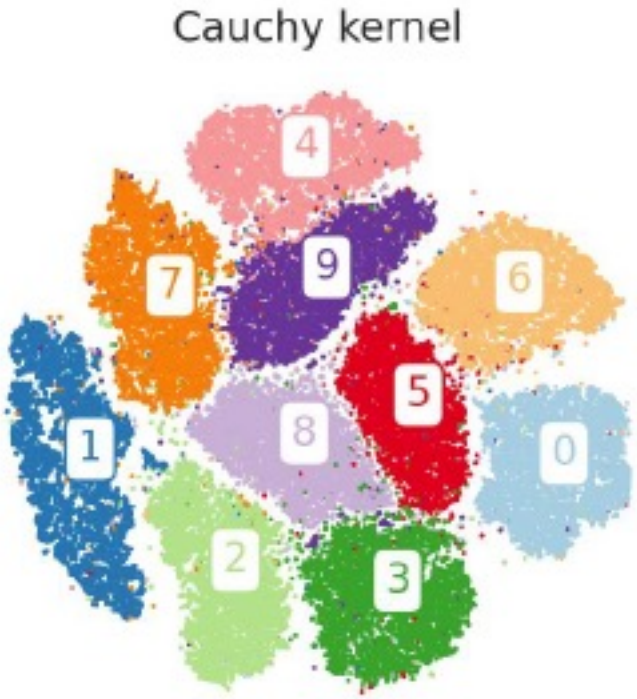
MNIST Data Visualization in 2D for different value of σ . Smaller the sigma denser are the σ

https://opentsne.readthedocs.io/en/latest/

t-SNE Visualization: Examples



SNE



t- SNE

UNIVERSITY OF ARKANSAS

Kernel PCA (KPCA)

Slides by: Rita Osadchy (cs.haifa.ac.il/~rita)

UNIVERSITY OF ARKANSAS

Kernel PCA (KPCA)

Slides by: Rita Osadchy (cs.haifa.ac.il/~rita)

UNIVERSITY OF ARKANSAS

Notations

- Inputs (**high dimensional**)

  $\boldsymbol{x_1, x_2, \ldots, x_n}$ points in R^D

- Outputs **(low dimensional)**

  $\boldsymbol{y_1, y_2, \ldots, y_n}$ points in R^d ($d << D$)

UNIVERSITY OF ARKANSAS

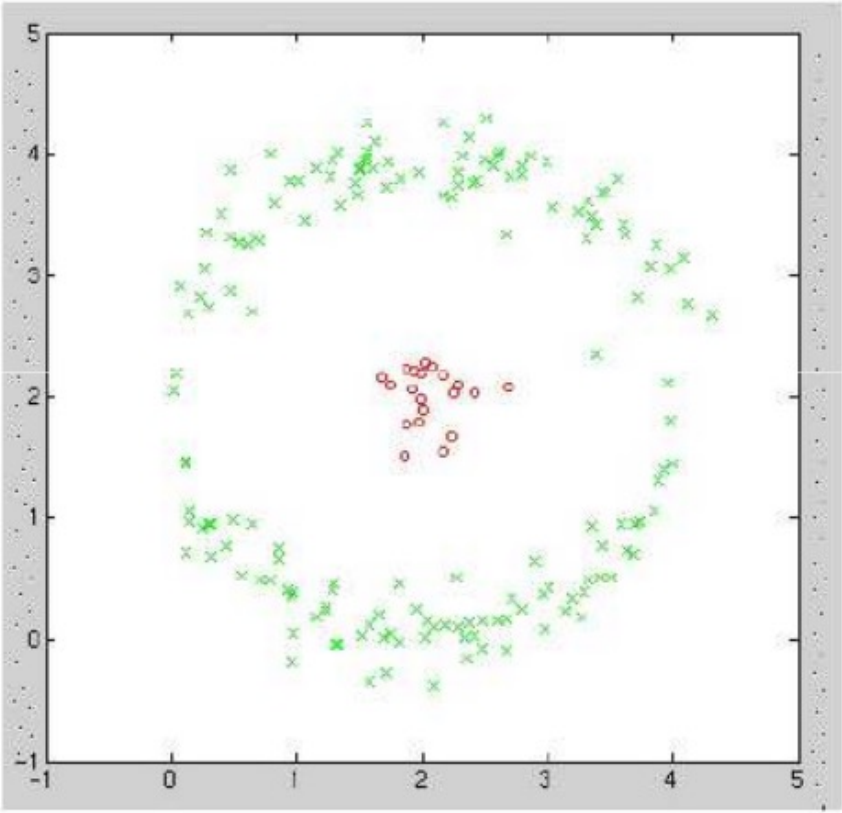
The “magic” of high dimensions

- Given some problem, how do we know what classes of functions are capable of solving that problem?

- VC (Vapnik-Chervonenkis) theory tells us that often mappings which take us into a higher dimensional space than the dimension of the input space provide us with greater classification power.

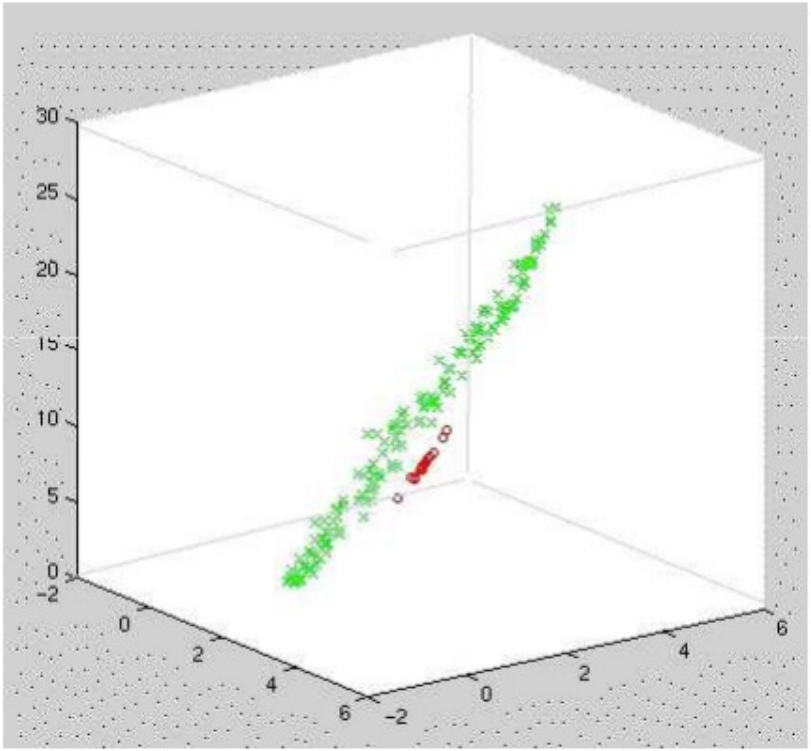
UNIVERSITY OF ARKANSAS

Example in $\mathbb{R}^2$



These classes are linearly inseparable in the input space.

Example: High-Dimensional Mapping



We can make the problem linearly separable by a simple mapping

$$\Phi : \mathbf{R}^2 \to \mathbf{R}^3$$

$$(x_1, x_2) \mapsto (x_1, x_2, x_1^2 + x_2^2)$$

UNIVERSITY OF ARKANSAS

Kernel Trick

- High-dimensional mapping can seriously increase computation time.
- Can we get around this problem and still get the benefit of high-D?
- Yes! Kernel Trick

$$K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$$

- Given *any* algorithm that can be expressed solely in terms of dot products, this trick allows us to construct different nonlinear versions of it.

UNIVERSITY OF ARKANSAS

Popular Kernels

Gaussian $K(\vec{x},\vec{x}') = \exp\left(-\beta \| \vec{x} - \vec{x}' \|^2\right)$

Polynomial $K(\vec{x},\vec{x}') = (1 + \vec{x} \cdot \vec{x}')^p$

Hyperbolic tangent $K(\vec{x},\vec{x}') = \tanh(\vec{x} \cdot \vec{x}' + \delta)$

UNIVERSITY OF ARKANSAS

Popular Kernels

Gaussian $K(\vec{x},\vec{x}') = \exp\left(-\beta \| \vec{x} - \vec{x}' \|^2\right)$

Polynomial $K(\vec{x},\vec{x}') = (1 + \vec{x} \cdot \vec{x}')^p$

Hyperbolic tangent $K(\vec{x},\vec{x}') = \tanh(\vec{x} \cdot \vec{x}' + \delta)$

UNIVERSITY OF ARKANSAS

Kernel Principal Component Analysis (KPCA)

- Extends conventional principal component analysis (PCA) to a high dimensional feature space using the “kernel trick”.
- Can extract up to n (number of samples) nonlinear principal components without expensive computations.

UNIVERSITY OF ARKANSAS

Making PCA Non-Linear

- Suppose that instead of using the points x_i we would first map them to some nonlinear feature space $\phi(x_i)$

  E.g. using polar coordinates instead of cartesian coordinates would help us deal with the circle.

- Extract principal component in that space (PCA)
- The result will be non-linear in the original data space!

UNIVERSITY OF ARKANSAS

Derivation

- Suppose that the mean of the data in the feature space is

$$\mu = \frac{1}{n}\sum_{i=1}^{n}\phi(x_i) = 0$$

- Covariance:

$$\mathrm{C} = \frac{1}{n}\sum_{i=1}^{n}\phi(x_i)\phi(x_i)^T$$

- Eigenvectors

$$\mathrm{C}v = \lambda v$$

Derivation Cont.

- Eigenvectors can be expressed as linear combination of features:

$$v = \sum_{i=1}^{n} \alpha_i \phi(x_i)$$

- Proof:

$$Cv = \frac{1}{n} \sum_{i=1}^{n} \phi(x_i) \phi(x_i)^T v = \lambda v$$

thus

$$v = \frac{1}{\lambda n} \sum_{i=1}^{n} \phi(x_i) \phi(x_i)^T v = \frac{1}{\lambda n} \sum_{i=1}^{n} (\phi(x_i) \cdot v) \phi(x_i)^T$$

Showing that $xx^T v = (x \cdot v)x^T$

$$(\boldsymbol{x}\boldsymbol{x}^T)\boldsymbol{v} = \begin{pmatrix} x_1x_1 & x_1x_2 & \dots & x_1x_M \\ x_2x_1 & x_2x_2 & \dots & x_2x_M \\ \vdots & \vdots & \ddots & \vdots \\ x_Mx_1 & x_Mx_2 & \dots & x_Mx_M \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_M \end{pmatrix}$$

$$= \begin{pmatrix} x_1x_1v_1 + x_1x_2v_2 + \ldots + x_1x_Mv_M \\ x_2x_1v_1 + x_2x_2v_2 + \ldots + x_2x_Mv_M \\ \vdots \\ x_Mx_1v_1 + x_Mx_2v_2 + \ldots + x_Mx_Mv_M \end{pmatrix}$$

Showing that $xx^T v = (x \cdot v)x^T$

$$= \begin{pmatrix} (x_1v_1 + x_2v_2 + \ldots + x_Mv_M)\, x_1 \\ (x_1v_1 + x_2v_2 + \ldots + x_Mv_M)\, x_2 \\ \vdots \\ (x_1v_1 + x_2v_2 + \ldots + x_Mv_M)\, x_M \end{pmatrix}$$

$$= \begin{pmatrix} x_1v_1 + x_2v_2 + \ldots + x_Mv_M \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_M \end{pmatrix}$$

$$= (\boldsymbol{x} \cdot \boldsymbol{v})\boldsymbol{x}$$

Derivation Cont.

- So, from before we had,

$$v = \frac{1}{n\lambda}\sum_{i=1}^{n}\phi(x_i)\phi(x_i)^T v = \frac{1}{n\lambda}\sum_{i=1}^{n}(\phi(x_i)\cdot v)\phi(x_i)^T$$

just a scalar

$$v = \sum_{i=1}^{n}\alpha_i\phi(x_i)$$

- Finding the eigenvectors is equivalent to finding the coefficients α_i

Derivation Cont.

- By substituting this back into the equation we get:

$$\frac{1}{n}\sum_{i=1}^{n}\phi(x_i)\phi(x_i)^T\left(\sum_{l=1}^{n}\alpha_{jl}\phi(x_l)\right)=\lambda_j\sum_{l=1}^{n}\alpha_{jl}\phi(x_l)$$

J: index for jth Eigen Vector

- We can rewrite it as

$$\frac{1}{n}\sum_{i=1}^{n}\phi(x_i)\left(\sum_{l=1}^{n}\alpha_{jl}K(x_i,x_l)\right)=\lambda_j\sum_{l=1}^{n}\alpha_{jl}\phi(x_l)$$

- Multiply this by $\phi(x_k)$ from the left:

$$\frac{1}{n}\sum_{i=1}^{n}\phi(x_k)^T\phi(x_i)\left(\sum_{l=1}^{n}\alpha_{jl}K(x_i,x_l)\right)=\lambda_j\sum_{l=1}^{n}\alpha_{jl}\phi(x_k)^T\phi(x_l)$$

Derivation Cont.

- By substituting this back into the equation we get:

$$\frac{1}{n}\sum_{i=1}^{n}\phi(x_i)\phi(x_i)^T\left(\sum_{l=1}^{n}\alpha_{jl}\phi(x_l)\right)=\lambda_j\sum_{l=1}^{n}\alpha_{jl}\phi(x_l)$$

- We can rewrite it as

$$\frac{1}{n}\sum_{i=1}^{n}\phi(x_i)\left(\sum_{l=1}^{n}\alpha_{jl}K(x_i,x_l)\right)=\lambda_j\sum_{l=1}^{n}\alpha_{jl}\phi(x_l)$$

- Multiply this by $\phi(x_k)$ from the left:

$$\frac{1}{n}\sum_{i=1}^{n}\phi(x_k)^T\phi(x_i)\left(\sum_{l=1}^{n}\alpha_{jl}K(x_i,x_l)\right)=\lambda_j\sum_{l=1}^{n}\alpha_{jl}\phi(x_k)^T\phi(x_l)$$

Derivation Cont.

- By plugging in the kernel and rearranging we get:

$$\mathrm{K}^2\alpha_j = n\lambda_j \mathrm{K}\alpha_j$$

We can remove a factor of K from both sides of the matrix (this will only affects the eigenvectors with zero eigenvalue, which will not be a principle component anyway):

$$\mathrm{K}\alpha_j = n\lambda_j \alpha_j$$

- We have a normalization condition for the α_j vectors:

$$v_j^T v_j = 1 \Rightarrow \sum_{k=1}^{n}\sum_{l=1}^{n} \alpha_{jl}\alpha_{jk}\phi(x_l)^T\phi(x_k) = 1 \Rightarrow \alpha_j^{\mathrm{T}}\mathrm{K}\alpha_j = 1$$

UNIVERSITY OF ARKANSAS

Derivation Cont.

- By multiplying $\mathrm{K}\alpha_j = n\lambda_j\alpha_j$ by α_j and using the normalization condition we get:

$$\lambda_j n \alpha_j^T \alpha_j = 1, \quad \forall j$$

- For a new point x, its projection onto the principal components is:

$$\phi(x)^T v_j = \sum_{i=1}^{n} \alpha_{ji} \phi(x)^T \phi(x_i) = \sum_{i=1}^{n} \alpha_{ji} K(x, x_i)$$

Normalization!

- In a matrix form

$$\tilde{\mathrm{K}} = \mathrm{K} - 2\mathbf{1}_{1/n}\mathrm{K} + \mathbf{1}_{1/n}\mathrm{K}\mathbf{1}_{1/n}$$

- where $\mathbf{1}_{1/n}$ is a matrix with all elements 1/n.

Instead of using a kernel Matrix K, we find eigen vectors of Normalized $\tilde{K}$ to estimate $\boldsymbol{\alpha}$

Summary of kernel PCA

- Pick a kernel
- Construct the normalized kernel matrix of the data (dimension m x m):

$$\tilde{\mathrm{K}} = \mathrm{K} - 2\mathbf{1}_{1/n}\mathrm{K} + \mathbf{1}_{1/n}\mathrm{K}\mathbf{1}_{1/n}$$

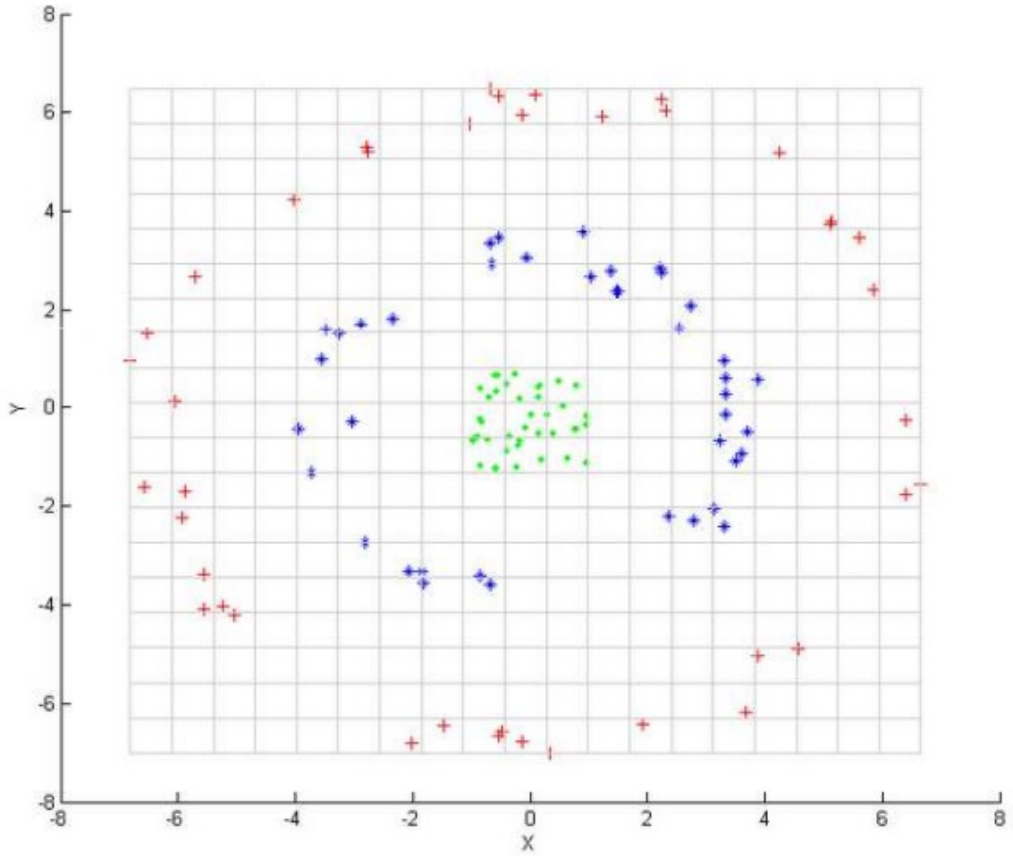
- Solve an eigenvalue problem:

$$\tilde{\mathrm{K}}\alpha_i = \lambda_i \alpha_i$$

- For any data point (new or old), we can represent it as

$$y_j = \sum_{i=1}^{n} \alpha_{ji} K(x, x_i), \quad \mathrm{j} = 1,..,d$$

Example: Input Points



UNIVERSITY OF ARKANSAS

Example: KPCA