Section 20 Summary and Future Study

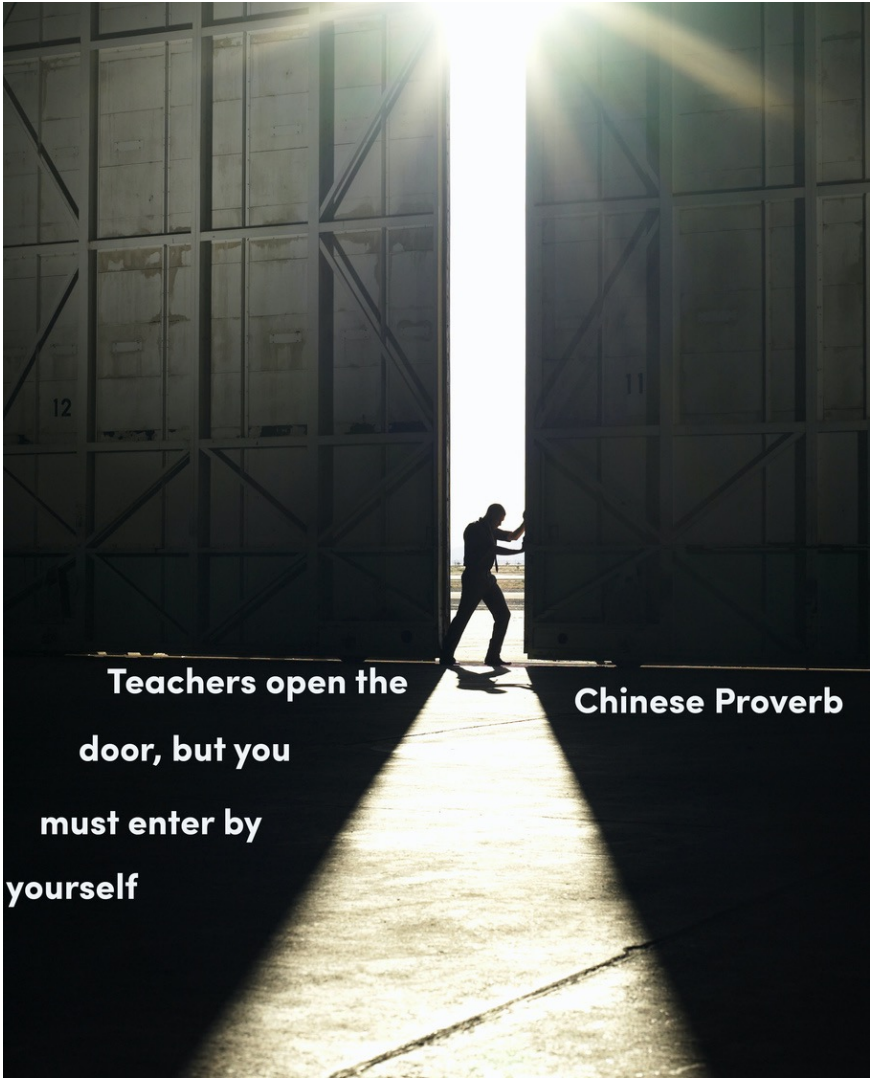
➢ Summary of what we learned.

1. Linear Algebra

- Matrix operations over fields
- Vector Spaces over fields
- Independence, basis and dimension
- Inner Product Space
- "General" Least Squares Methods
- Markov chain, Dynamical System and Perron-Frobenius Theorem
- Singular Value Decomposition(SVD)
- Matrix Calculus
- "Matrix prob"

2. Data Analysis and Machine Learning

- **Linear Methods for Regression and classification.**
- Ridge Regression
- Lasso Regression
- Locally weighted Regression
- Logistics Regression
- **Gradient descent and Newton's method**
- **Cross-Validation, Bias and Variance Trade-off**
- **Neural Network**
- **Convolutional neural network**
- **Principal Component Analysis(PCA)**
- **Support Vector Machines(SVM)**
- **Kernel Methods**


Teachers open the door, but you must enter by yourself
Chinese Proverb

➢ Suggestions for future study:

Pure Math:

- Analysis.
- Algebra.
- Geometry and Topology.
- Number theory.
- Combinatorics.
- Etc.

Applied Math (theory):

E.g., differential equations, numerical analysis, probability, **statistics**, etc.

https://en.wikipedia.org/wiki/Applied_mathematics

Applications of mathematics

E.g., in physics, engineering, medicine, biology, finance, business, computer science, and industry.

Undergraduate Courses:

Several very useful 3000 level MATH courses,

https://catalog.northeastern.edu/graduate/science/mathematics/#coursestext

Analysis Courses:

MATH 4525. Applied Analysis.
MATH 4541. Advanced Calculus.
MATH 4545. Fourier Series and PDEs
MATH 4555. Complex Variables.
…

Relating to Data Science and Machine Learning:

MATH 4570. Matrix Methods in Data Analysis and Machine Learning.
MATH 4571. Advanced Linear Algebra.
MATH 4575. Introduction to Cryptography.
MATH 4581. Statistics and Stochastic Processes.
…

MS in applied math courses:

MATH 5111 Algebra 1 MATH 5112 Algebra 2
MATH 5101 Analysis 1 MATH 5102 Analysis 2
MATH 5110 Applied Linear Algebra and matrix analysis
MATH 5131 Intro to Math Methods & Models
MATH 7241 Probability 1
MATH 7341 Probability 2
MATH 7203 Numerical Analysis 1
MATH 7205 Numerical Analysis 2
MATH 7243 Machine Learning and statistical learning 1
MATH 7339 Machine Learning and statistical learning 2
MATH 7343 Applied Statistics
MATH 7342 Mathematical Statistics
MATH 7234 Optimization and Complexity
MATH 7233 Graph Theory
MATH 7344 Regression, ANOVA and Design
MATH 7346 - Time Series
…

https://catalog.northeastern.edu/graduate/science/mathematics/applied-mathematics-ms/#programrequirementstext

CS, DS, Engineering MS courses:

CS 6140 Machine Learning
CS 6220 Data Mining Techniques
DA 5020 Collecting, Storing, and Retrieving Data
DA 5030 Introduction to Data Mining/Machine Learning
DS 5220 Supervised Machine Learning and Learning Theory
DS 5230 Unsupervised Machine Learning and Data Mining
EECE 5644 Introduction to Machine Learning and Pattern Recognition
DS 5010: Introduction to Programming for Data Science
DS 5110: Introduction to Data Management and Processing
CS 5800: Algorithms
CS 5200: Database Management Systems
CS 6120: Natural Language Processing
CS 7140: Advanced Machine Learning
CS 7150: Deep Learning
…

We do not have time to learn all courses/knowledge, even for PhD or professors.

All materials in MATH 4570 are fundamental and useful. More importantly,
I hope that you have developed or started developing self-learning ability
along study this course.

Once we have the ability of learning, you can learn the knowledge you needed.
You will keep learning in your whole life.

. Trace

- **Machine Learning:**

1. **Supervised Learning (Regression/Classification)**
2. **Unsupervised Learning**
3. **Reinforcement Learning**

1. Parametric methods
2. Un-parametric methods

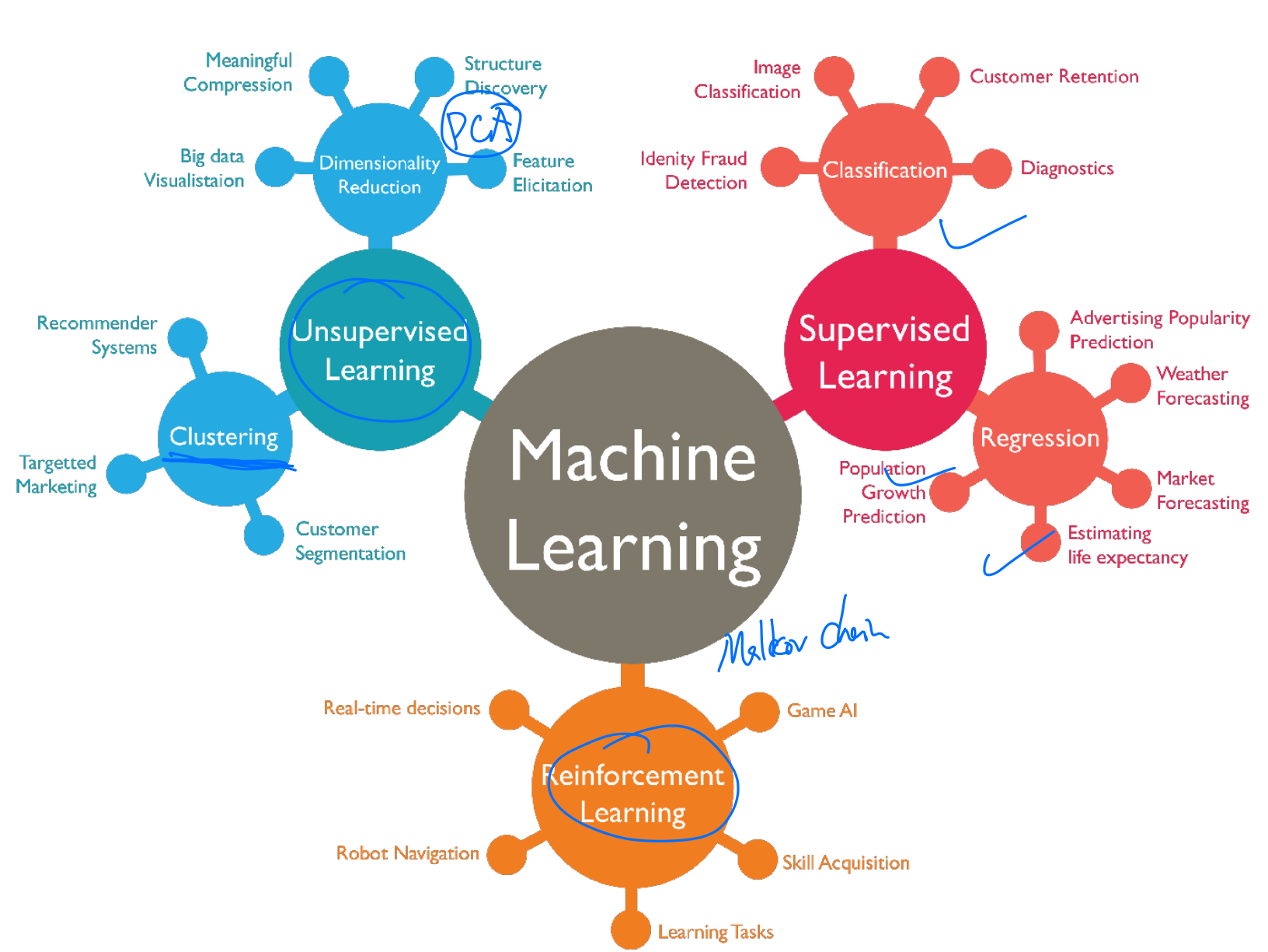
- **More topics in Data Analysis and Machine Learning:**

My **MATH 7243 course**:

https://web.northeastern.edu/he.wang/Teaching/Teaching7243/Math7243.html

A good overview of main categories in machine learning in 6 minutes:

https://www.youtube.com/watch?v=0_lKUPYEYyY


Machine Learning
Unsupervised Learning
Dimensionality Reduction
Meaningful Compression
Structure Discovery
PCA
Big data Visualistaion
Feature Elicitation
Clustering
Recommender Systems
Targetted Marketing
Customer Segmentation
Supervised Learning
Classification
Image Classification
Customer Retention
Idenity Fraud Detection
Diagnostics
Regression
Advertising Popularity Prediction
Weather Forecasting
Market Forecasting
Estimating life expectancy
Population Growth Prediction
Reinforcement Learning
Real-time decisions
Game AI
Robot Navigation
Skill Acquisition
Learning Tasks

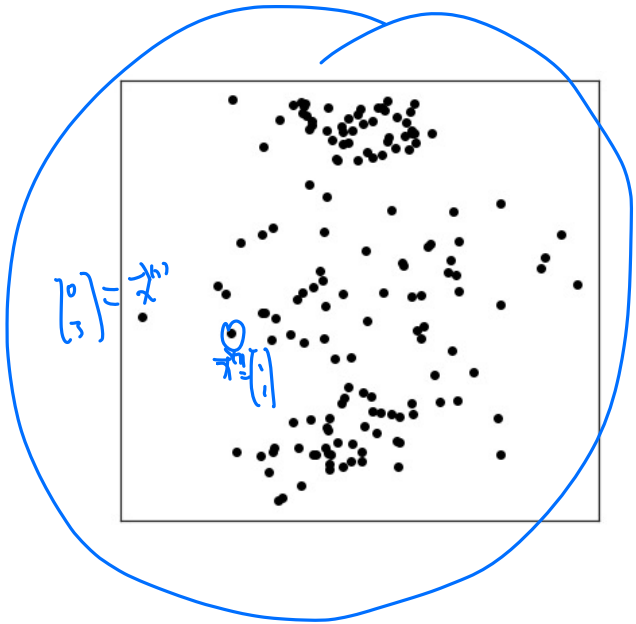
- **Unsupervised learning**
  - **Dimension Reduction**
    - ❖ **PCA**
  - **Clustering**
    - ❖ **K means.**
    - ❖ Hierarchical clustering
    - ❖ Density based
    - ❖ Spectral clustering
    - ❖ Distribution methods

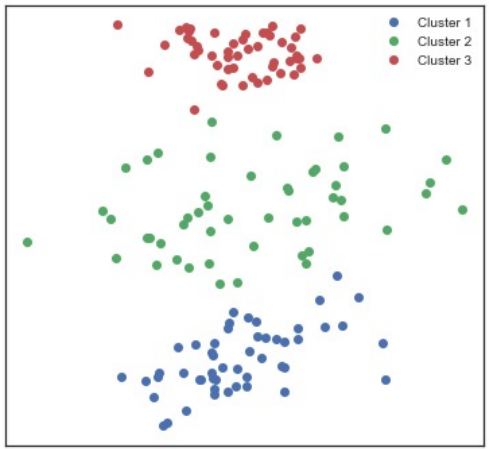
❖ Clustering

Data: $\vec{x}^{(i)}$ **or Data matrix** X

Just input data, no output labels no $\vec{y}$

Unsupervised Learning Goal: Learn some **underlying hidden structure** of the data. Often used as part of exploratory data analysis. **Clustering** looks to find homogeneous subgroups among the observations.





Example: Market segmentation

Cus

You are the owner of a shop. It doesn't matter if you own an e-commerce or a supermarket. It doesn't matter if it is a small shop or a huge company such as Amazon or Netflix, it's better to know your customers.

Customer ID	Gender	Age	Annual Income	Spending Score
1	Male	19	15	39
2	Male	21	15	81
3	Female	20	16	6
4	Female	23	16	77
5	Female	31	17	40

You were able to collect basic data about your customers holding a membership card such as Customer ID, age, gender, annual income, and spending score. This last one is a score based on customer behavior and purchasing data. There are some new products on the market that you are interested in selling. But you want to target a specific type of clients for each one of the products.

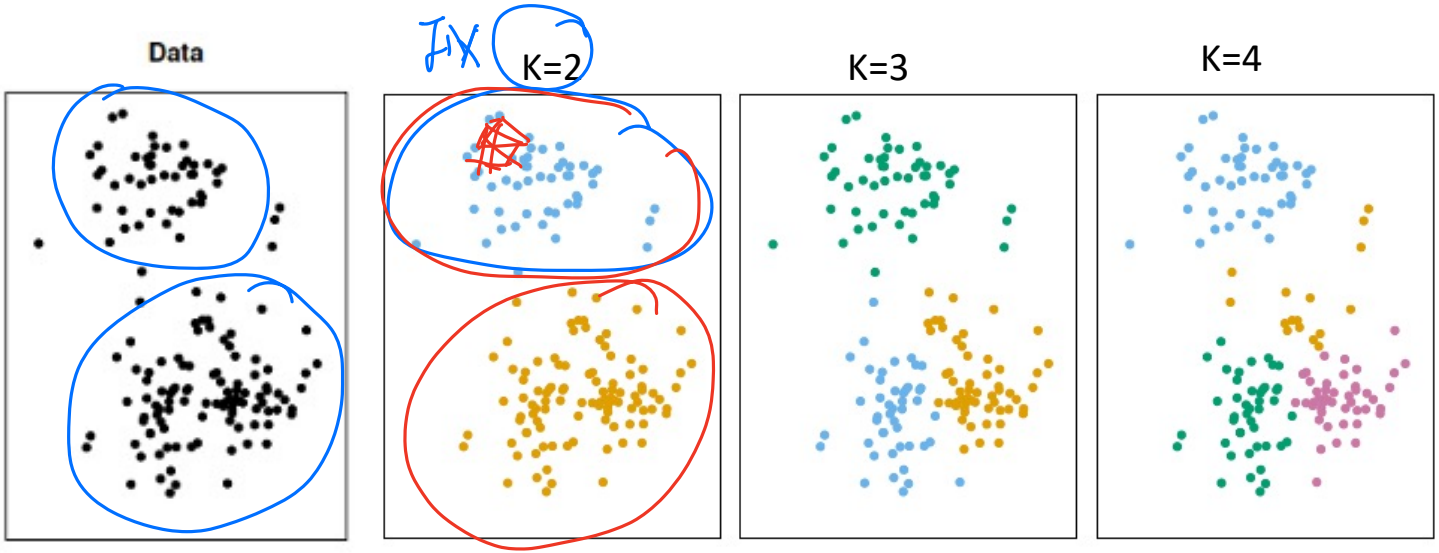
The **goal** is to perform market segmentation by identifying subgroups of people who might be more receptive to a particular form of advertising, or more likely to purchase a particular product.

The task of performing market segmentation amounts to **clustering** the people in the data set.

https://www.kaggle.com/vjchoudhary7/customer-segmentation-tutorial-in-python

Partition the data set of n observations into K distinct, non-overlapping subsets,

denoted as C_k, for $k = 1, \ldots, K$, is called a **cluster**.



A simulated data set with 150 observations in two-dimensional space. Panels show the results of applying K-means clustering with different values of K, the number of clusters. The color of each observation indicates the cluster to which it was assigned using the K-means clustering algorithm. Note that there is no ordering of the clusters, so the cluster coloring is arbitrary. These cluster labels were not used in clustering; instead, they are the outputs of the clustering procedure.

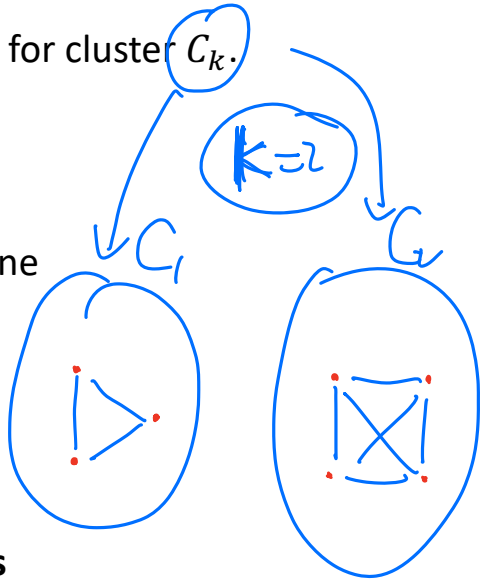
Good General Clustering: the within-cluster variation is as small as possible

Let $W(C_k)$ be a measure of the **within-cluster variation** for cluster C_k.

There are several different ways to define $W(C_k)$.

For example, using **squared Euclidean distance**, we define

$$W(C_k) = \frac{1}{|C_k|} \sum_{i,j \in C_k} \|\mathbf{x}_i - \mathbf{x}_j\|^2$$



We wish to **minimize the total within-cluster variations**

$$\text{minmize}_{C_1,\ldots,C_K} \left\{ \sum_{i=1}^{K} W(C_K) \right\}$$

The problem is computationally difficult ("non-deterministic polynomial acceptable problems" NP-hard).

❖ K-Means cluster algorithm

example

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:

   1) For each of the $\boldsymbol{K}$ **clusters**, compute the **cluster centroid** $\boldsymbol{\mu_k}$. The k-th cluster centroid μ_k is the vector of the $\boldsymbol{p}$ **feature means** for the observations in the k-th cluster.
   2) Assign each observation to the cluster whose centroid is closest (where closest is defined using Euclidean distance).

K=2

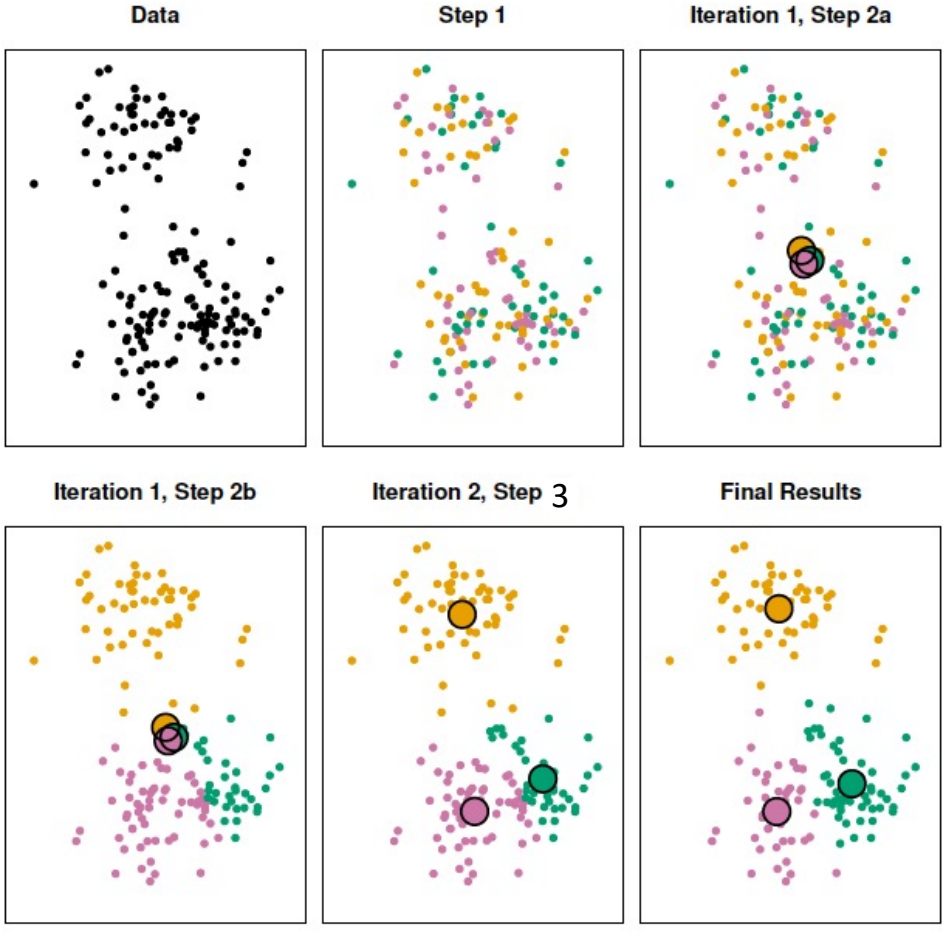
- For every j, compute **new cluster centers**:

$$\mu_j := \frac{\sum_{i=1}^{m} \mathbb{I}\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{m} \mathbb{I}\{c^{(i)} = j\}}$$

- For every i, **assign** each point to cluster,

$$c^{(i)} := \arg\min_{j} ||x^{(i)} - \mu_j||$$

An example of the K-means algorithm with K = 3.



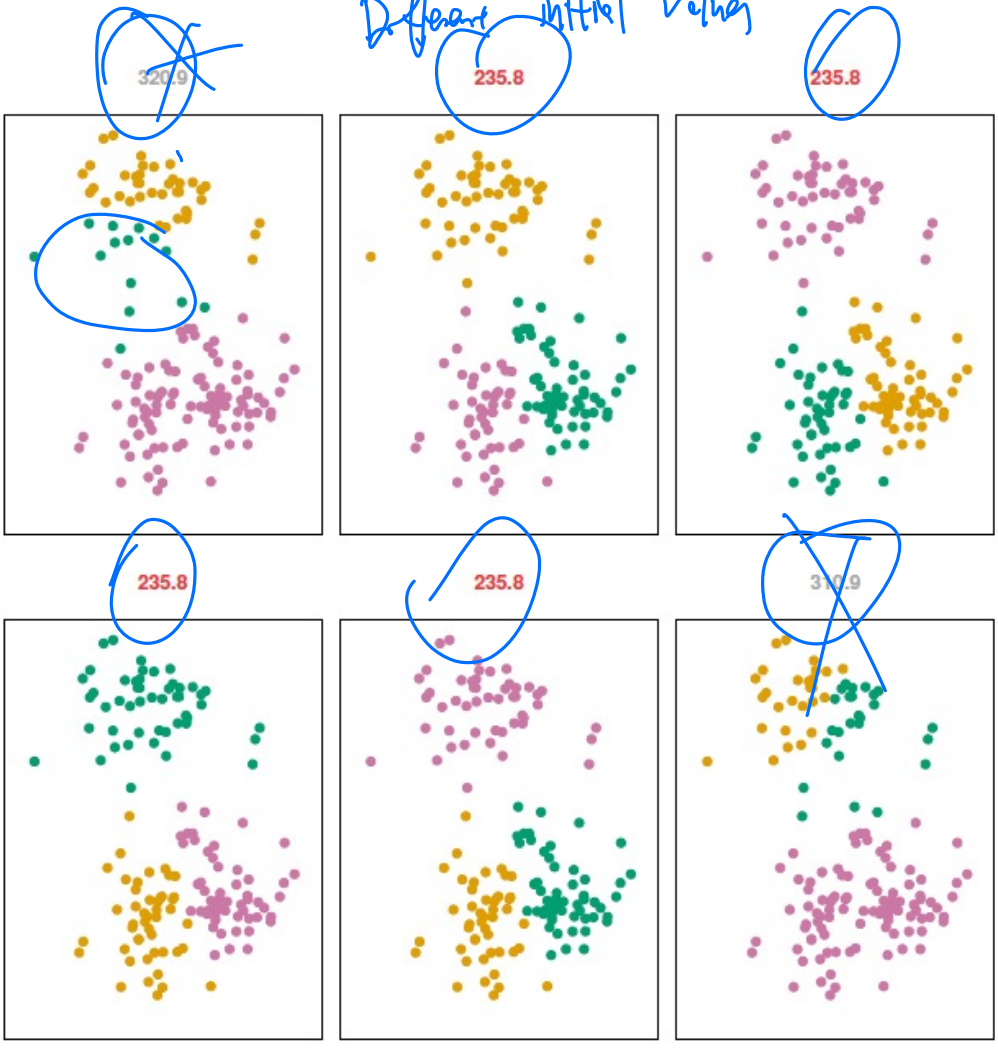
Step 1, each observation is randomly assigned to a cluster.

Step 2(a), the cluster centroids are computed. These are shown as large colored disks. Initially the centroids are almost completely overlapping because the initial cluster assignments were chosen at random.

Step 2(b), each observation is assigned to the nearest centroid.

Step 3 is once again performed, leading to new cluster centroids.

The final results obtained after ten iterations.



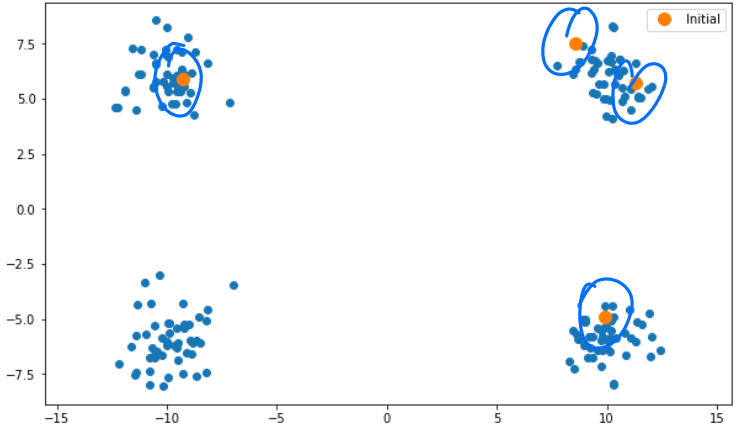
K-means clustering performed **six times** on the previous data with K = 3, each time with a **different** random assignment of the observations in Step 1 of the K-means algorithm.

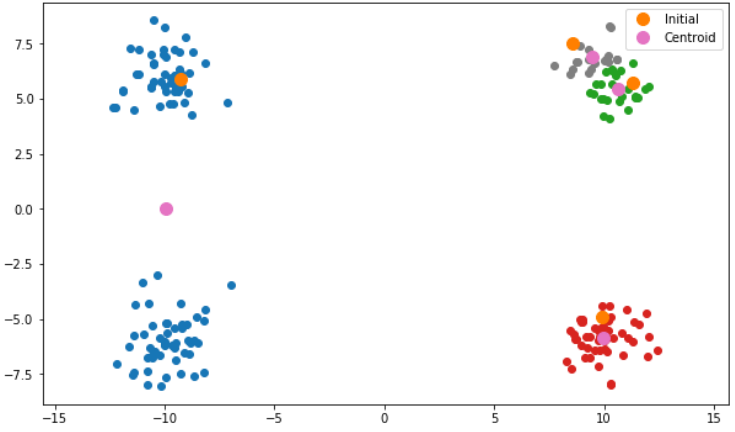
Above each plot is the value of the objective. **Three different** local optima were obtained, one of which resulted in a smaller value of the objective and provides better separation between the clusters.

Those labeled in red all achieved the same best solution, with an objective value of 235.8.

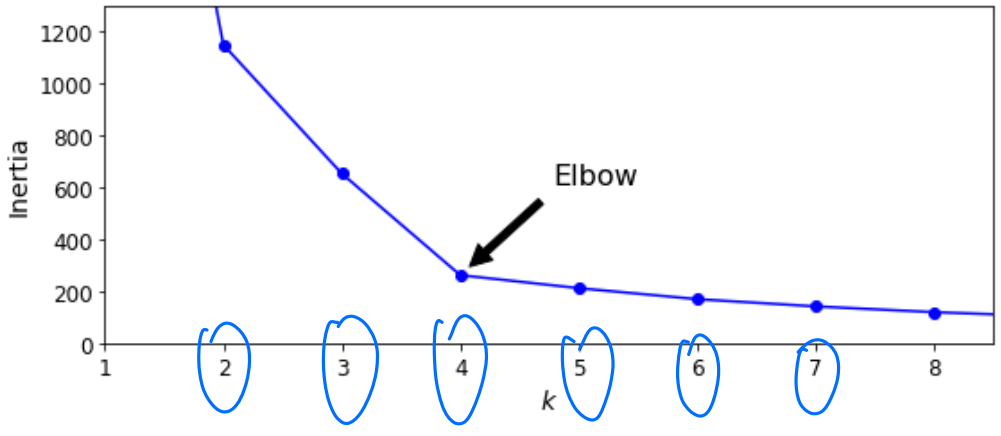
Practical Issues: Initialization

Initialization is probably the greatest difficulty in most greedy algorithms.





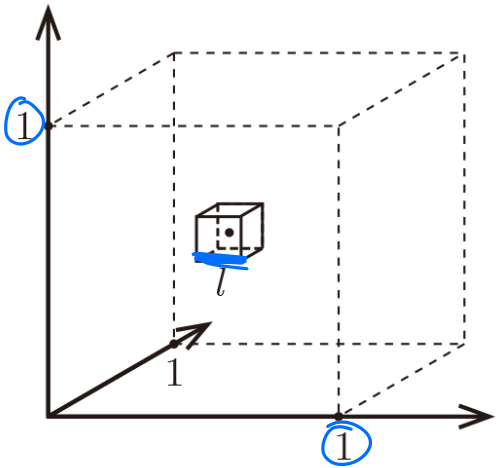
Choosing K



The choice of K is the other main practical issue in K-means clustering. A common criteria is to compute the clustering for many K, plot them, and set K to be just larger than the steepest “elbow”.

Bad News: Curse of dimensionality

In high dimensional spaces, points that are drawn from a probability distribution, tend to never be close together.



Distances between points

Draw n points **uniformly** at random within the unit cube $D = [0,1]^d$.

Considering the k nearest neighbors of such a test point.

Let l be the edge length of the smallest hyper-cube that contains all k-nearest neighbor of a test point.

Then $l^d \approx \frac{k}{n}$ and $l = \left(\frac{k}{n}\right)^{1/d}$

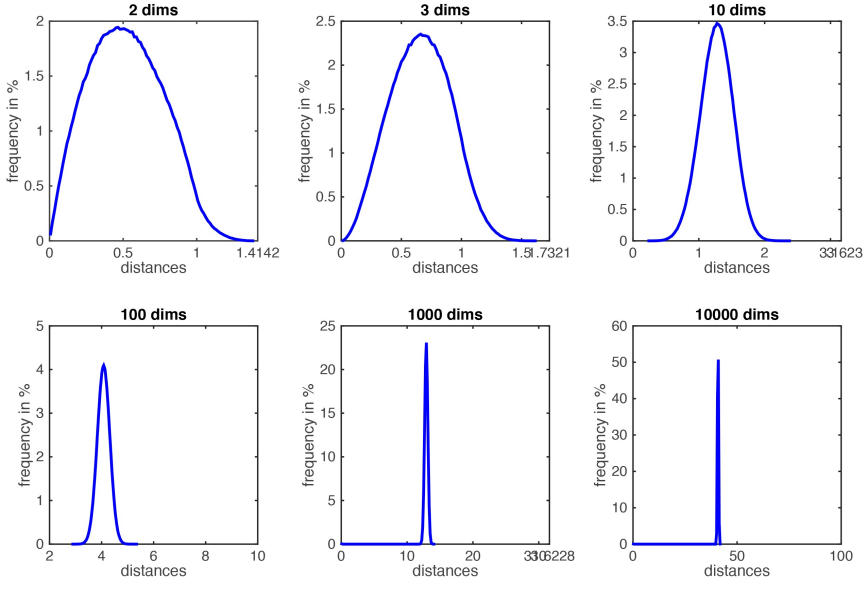
So as d large enough, almost the entire space is needed to find the 10-NN.

Suppose n=1000 and k=10,

d	l
2	0.1
10	0.63
100	0.955
1000	0.9954

Distance between two randomly drawn data points $\vec{x}$ and $\vec{y}$ increases drastically with their dimensionality.

$$dist(\vec{x}, \vec{y}) = \sqrt{(x_1 - y_1)^2 + \cdots + (x_d - y_d)^2}$$



The histogram plots show the distributions of all pairwise distances between randomly distributed points within d-dimensional unit squares.

As the dimensions d grows, all distances concentrate within a very small range.

Remarks on Project Presentation:

1. Submit your slides to Canvas on Tuesday
2. For the slides, have the complete material including the introduction and background.
3. For in-class oral presentation, we can skip the introduction and background to save some time.
4. Each group have 9 minutes.
5. We can try to start 5 minutes earlier at 1:30pm if the one group can come earlier.

I will be available most of the time on Tuesday. If your group want to ask me a question, feel free to email me to set a time.

Remarks on Project Presentation:

Due: May 1, Sunday.

Please submit the following 3 files: (Do NOT zip your files for the final paper and slides)

1. final paper (pdf file)
2. updated presentation slides (ppt or pdf)
3. python code (.ipynb or .py file, or a GitHub link for your code.)

Final Paper submission. May 1 Sunday. (You can use the tex template in the Project page for writing the paper. Your paper should look beautiful with no typo. The included images should be generated but not screenshots. If you want to include some key code, it needs to be written in a good format in tex.)