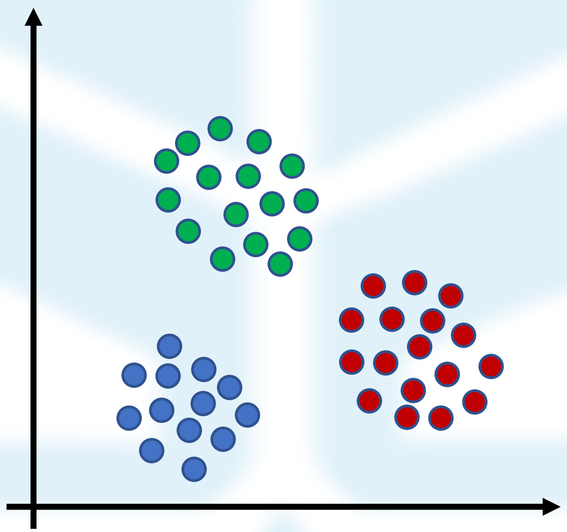
به نام او، به یاد او، برای او

Machine Learning

Clustering



Dr. *Ali Valinejad*

valinejad.ir
valinejad@umz.ac.ir

Supervised learning vs. Unsupervised learning

❖ **Supervised learning:**

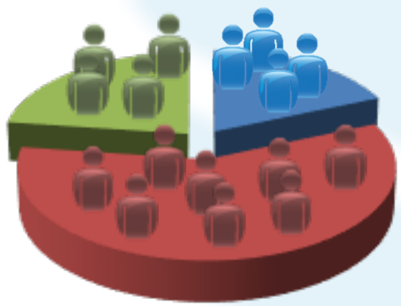
discover patterns in the data that relate data attributes with a target (class) attribute. These patterns are then utilized to predict the values of the target attribute in future data instances.

❖ **Unsupervised learning:**

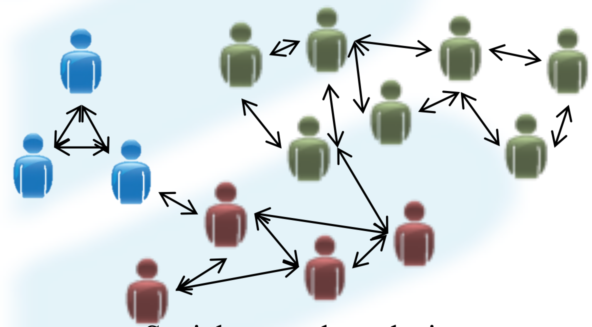
The data have no target attribute. We want to explore the data to find some intrinsic structures in them.

Clustering

❖ Clustering is a mode of unsupervised learning. Given a collection of data set, the *goal* is to find groups(clusters) of objects in data set such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups.

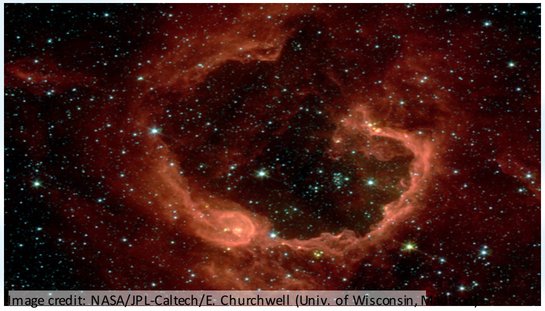


Market segmentation



Social network analysis

Organize computing clusters



Astronomical data analysis

What is Clustering for?

Example 1: groups people of similar sizes together to make “small”, “medium” and “large” T-Shirts.

Tailor-made for each person: too expensive

One-size-fits-all: does not fit all.

Example 2: In marketing, segment customers according to their similarities, to do targeted marketing.

Example 3: Given a collection of text documents, we want to organize them according to their content similarities, to produce a topic hierarchy.

Aspects of clustering

1. A clustering algorithm
2. A distance (similarity, or dissimilarity) function
3. Clustering quality

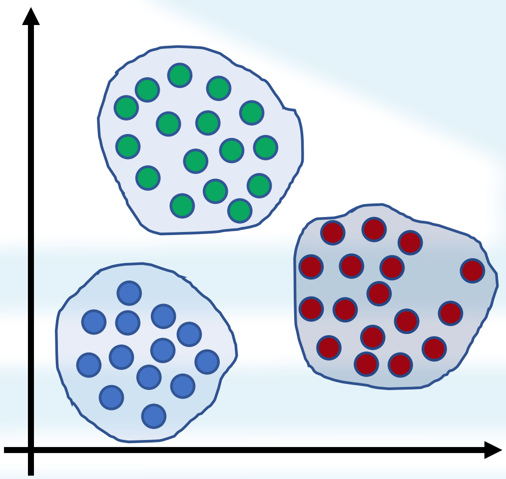
Aspects of clustering

1. A clustering algorithm
   - Partitional clustering : Partitional algorithms divide data set objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset.They include:
     - K-means and derivatives
     - Fuzzy c-means clustering
     - QT clustering algorithm

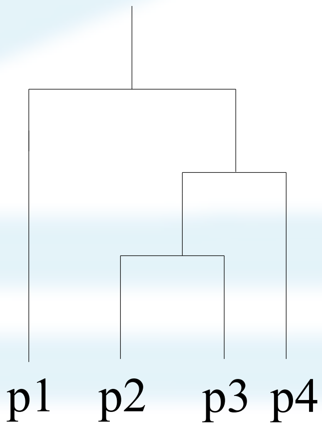
   - Hierarchical clustering: These find successive clusters using previously established clusters. A set of nested clusters organized as a hierarchical tree.

     1. Agglomerative ("bottom-up"): Agglomerative algorithms begin with each element as a separate cluster and merge them into successively larger clusters.

     2. Divisive ("top-down"): Divisive algorithms begin with the whole set and proceed to divide it into successively smaller clusters.



Partitional clustering



Hierarchical clustering

Aspects of clustering

1. A clustering algorithm
   - Partitional clustering:
   - Hierarchical clustering:

2. A distance (similarity, or dissimilarity) function

   Distance function will determine how the similarity of two elements is calculated and it will influence the shape of the clusters.

   Some distance functions:
   - The Euclidean distance (also called 2-norm distance.
   - The Manhattan distance (also called taxicab norm or 1-norm).
   - The maximum norm.
   - The Mahalanobis distance corrects data for different scales and correlations in the variables.
   - Inner product space: The angle between two vectors can be used as a distance measure when clustering high dimensional data
   - Hamming distance (sometimes edit distance) measures the minimum number of substitutions required to change one member into another.

Aspects of clustering

1. A clustering algorithm
   - Partitional clustering:
   - Hierarchical clustering:

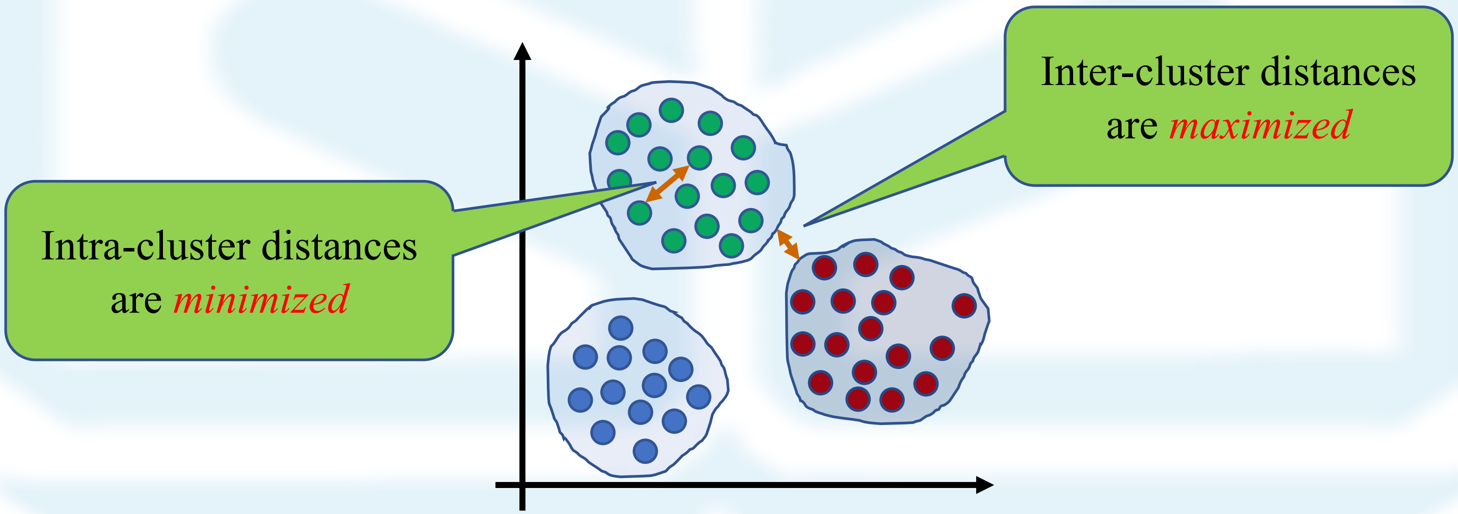
2. A distance (similarity, or dissimilarity) function

   Distance function will determine how the similarity of two elements is calculated and it will influence the shape of the clusters.

3. Clustering quality

   Inter-clusters distance ⇒ maximized

   Intra-clusters distance ⇒ minimized



❖ The quality of a clustering result depends on the ***algorithm***, the ***distance function***, and the ***application***.

K-means Algorithm

Input:

- K, (number of clusters)
- Training set: $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$, $x^{(i)} \in \mathbb{R}^n$ for i=1,2,…, m.

Goal:

- The *k*-means algorithm partitions the given data set into *k* different groups(clusters), so that the data in each cluster (ideally) share some common trait - often according to some defined distance measure.

K-means Algorithm

K-means Algorithm

Input: Number of clusters (K)

Training set $\{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$, $x^{(i)} \in \mathbb{R}^n$ for i=1,2,…, m

Output:

$c^{(i)}$: =index (from 1 to K) of cluster centroid closest to $x^{(i)}$.

Step 1. Randomly initialize K cluster centroids $\mu_1, \mu_2, \dots, \mu_K \in \mathbb{R}^n$.

- Randomly pick K training examples.
- Set $\mu_1, \mu_2, \dots, \mu_K$ equal to these K examples

$$1_{\{c^{(i)}-k\}} = \begin{cases} 1, & if\ c^{(i)} = k \\ 0, & o.w. \end{cases}$$

Step 2.

Repeat until convergence{

for i=1:m

$$c^{(i)} = \underset{k}{\operatorname{argmin}} \left\| \boldsymbol{x}^{(i)} - \mu_k \right\|^2$$

for k=1:K

$$\mu_k = \frac{\sum_{i=1}^{m} 1_{\{c^{(i)}-k\}}\, \boldsymbol{x}^{(i)}}{\sum_{i=1}^{m} 1_{\{c^{(i)}-k\}}}$$ **(**μ_k is average (mean) of points assigned to cluster k.**)**

}

Stopping Criteria: In order to see if the algorithm converges, we look at the **distortion function** defined as follows:

$$J(\boldsymbol{c}, \boldsymbol{\mu}) = \sum_{i=1}^{m} \left\| \boldsymbol{x}^{(i)} - \mu_{c^{(i)}} \right\|^2$$

* Depending on the initialization of cluster centroids K-means can produce different results

K-means Algorithm

K-means Optimization Objective

Training set: $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$, $x^{(i)} \in \mathbb{R}^n$ for i=1,2,…, m.

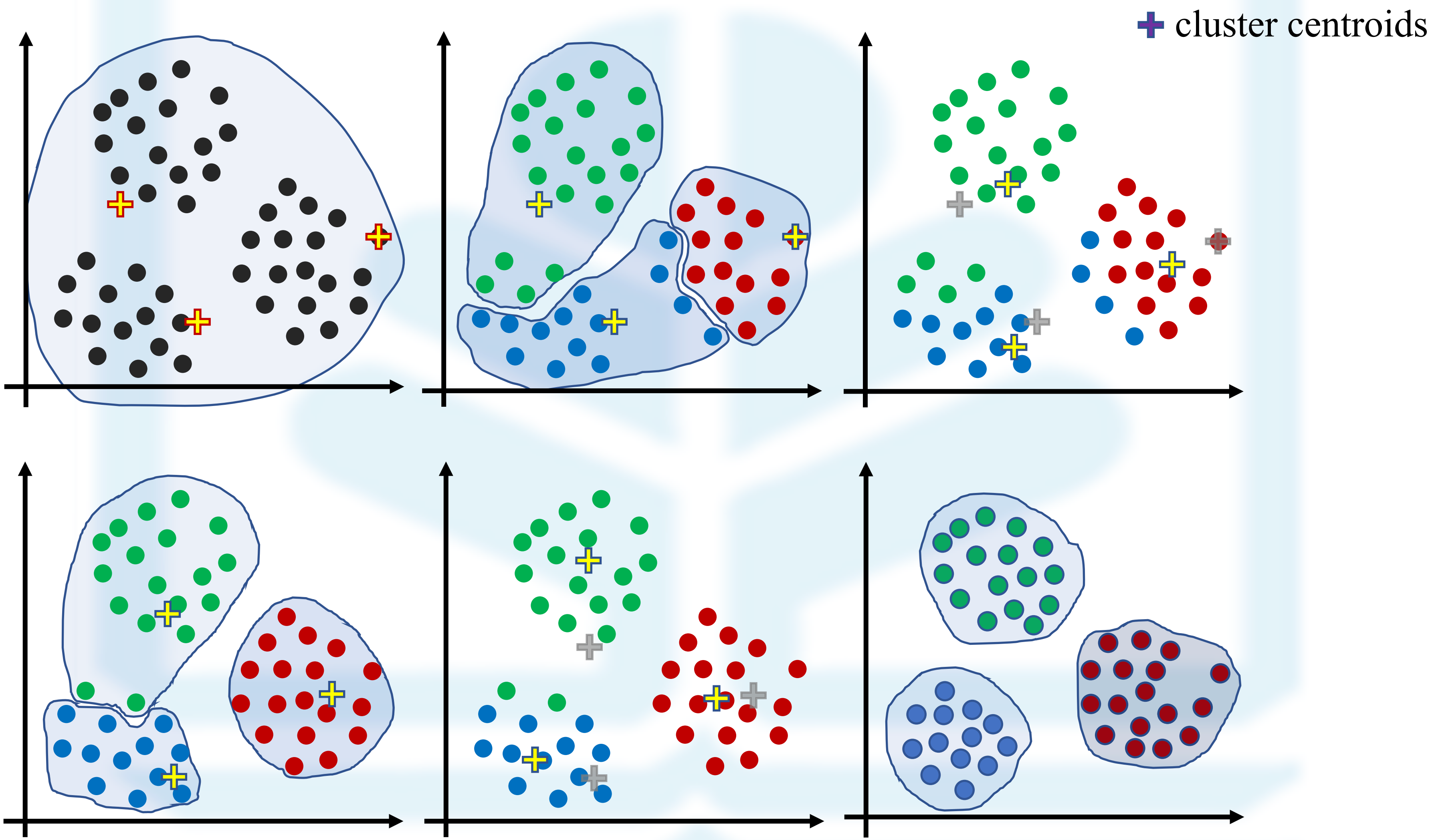
$c^{(i)}$:=index (from 1 to K) of cluster centroid closest to $x^{(i)}$.

μ_k: average (mean) of points assigned to cluster k, for $k = 1, 2, \ldots, K$.

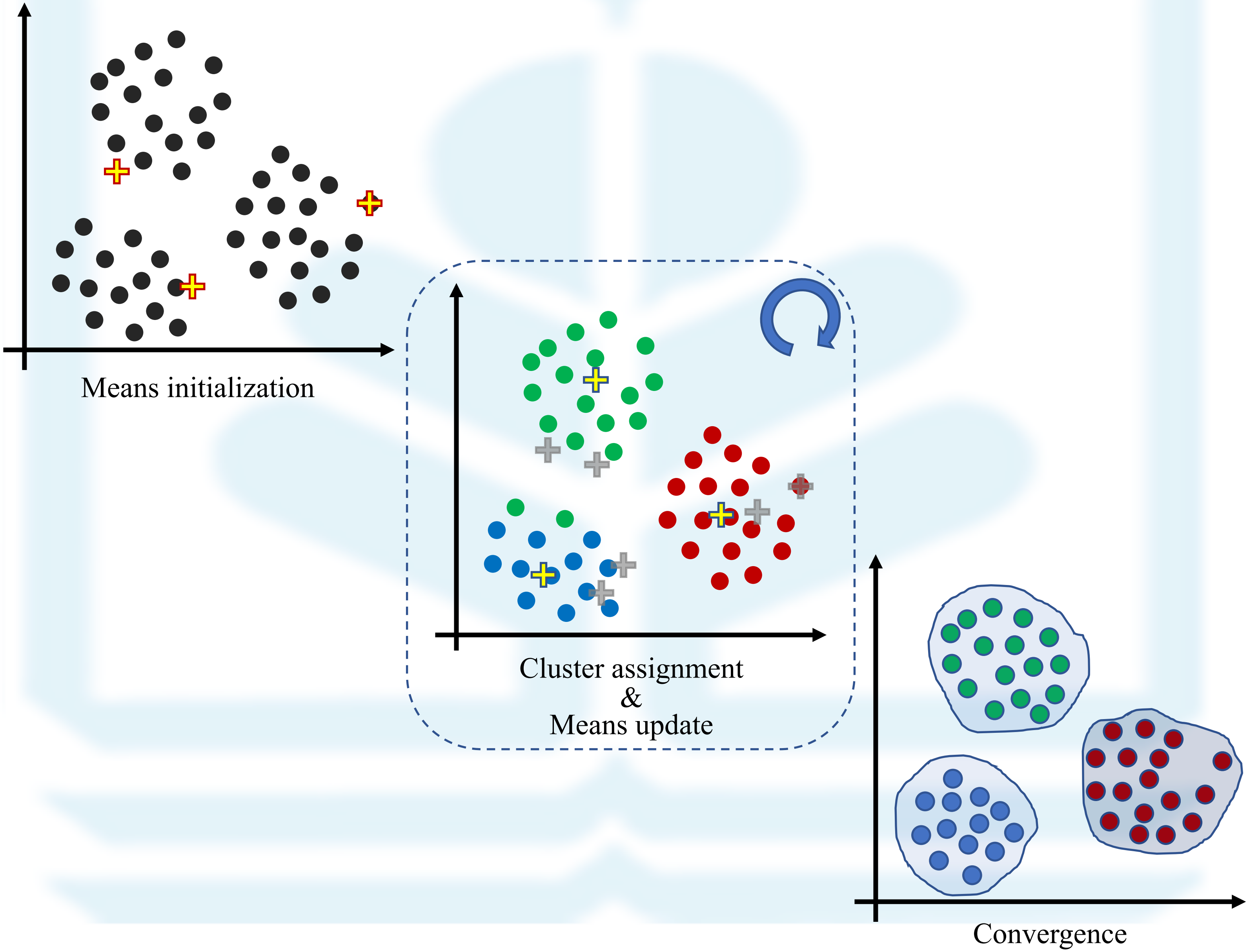
$\mu_{c^{(i)}}$: cluster centroid of cluster to which example $\boldsymbol{x}^{(i)}$ has been assigned.

$$\min_{\substack{c^{(1)}, c^{(2)}, \ldots, c^{(m)} \\ \mu_1, \mu_2, \ldots, \mu_K}} J\left(c^{(1)}, c^{(2)}, \ldots, c^{(m)}, \mu_1, \mu_2, \ldots, \mu_K\right) = \frac{1}{m} \left\| \boldsymbol{x}^{(i)} - \mu_{c^{(i)}} \right\|^2$$

K-means Algorithm



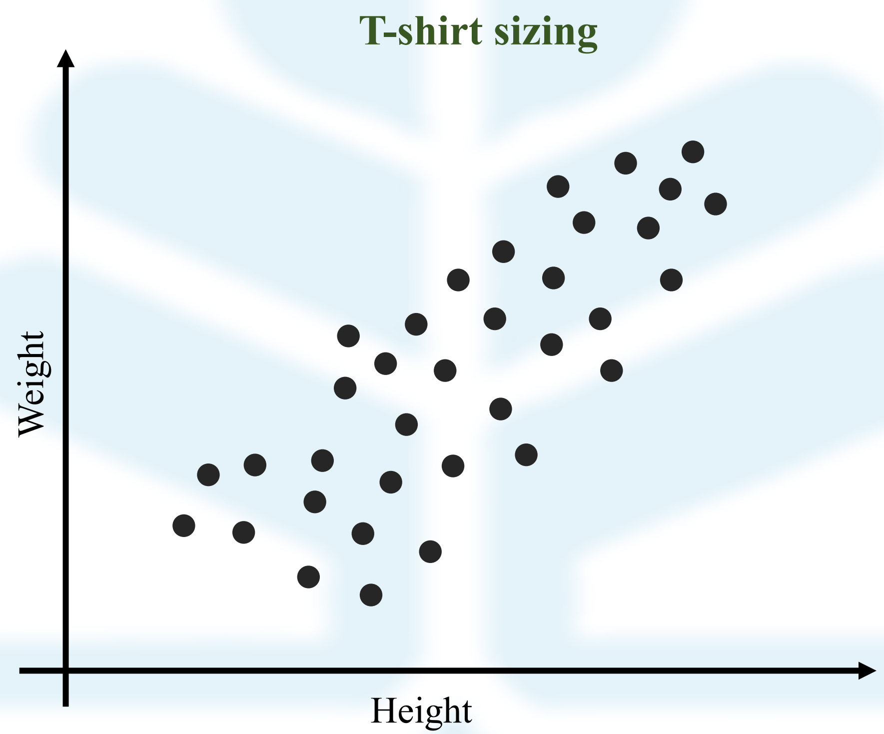
K-means Algorithm



K-means Algorithm

What is the right value of K?

Sometimes, you’re running K-means to get clusters to use for some later/downstream purpose. Evaluate K-means based on a metric for how well it performs for that later purpose.



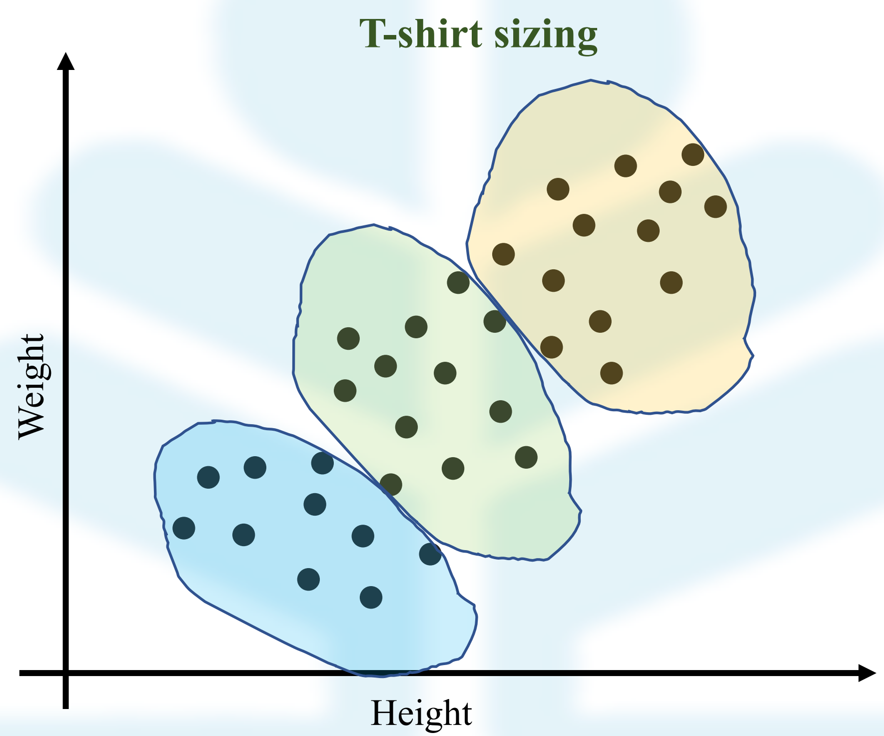
S, M, L: Small, Medium, Large

S, M, L, XL: Small, Medium, Large, Extra Large

XS, S, M, L, XL: Extra Small, Small, Medium, Large, Extra Large

K-means Algorithm

What is the right value of K?



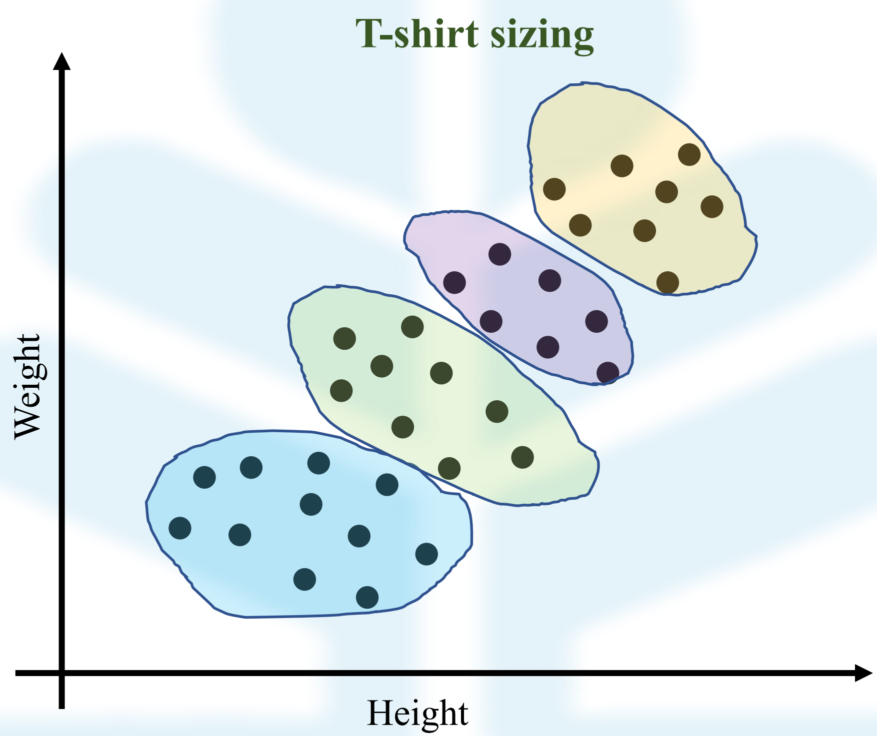
S, M, L: Small, Medium, Large

S, M, L, XL: Small, Medium, Large, Extra Large

XS, S, M, L, XL: Extra Small, Small, Medium, Large, Extra Large

K-means Algorithm

What is the right value of K?



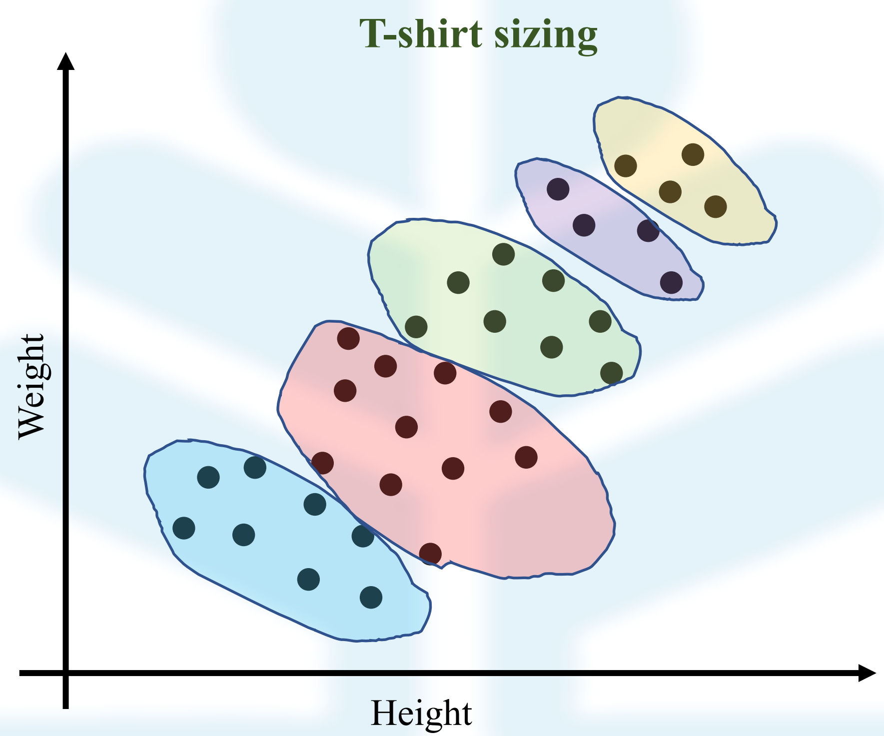
S, M, L: Small, Medium, Large

S, M, L, XL: Small, Medium, Large, Extra Large

XS, S, M, L, XL: Extra Small, Small, Medium, Large, Extra Large

K-means Algorithm

What is the right value of K?



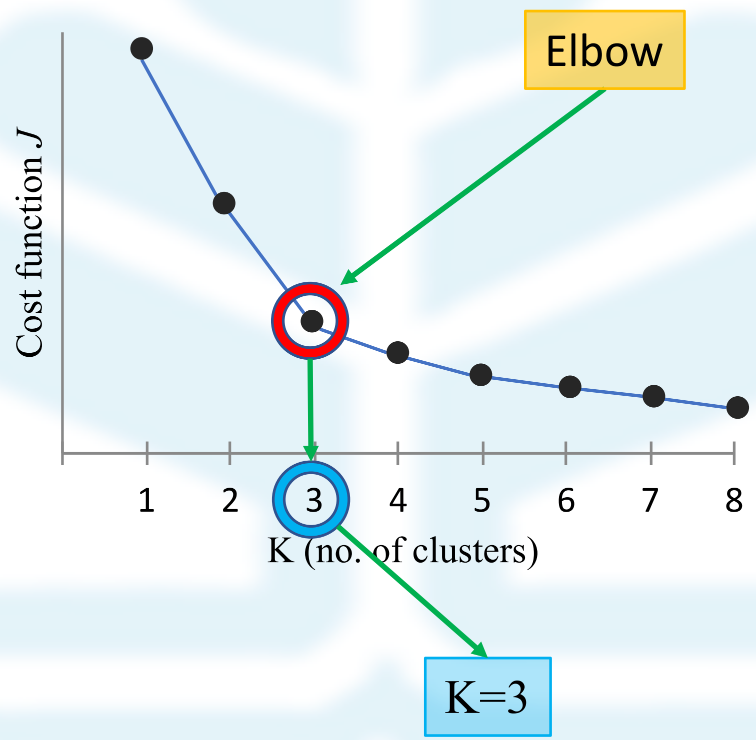
S, M, L: Small, Medium, Large

S, M, L, XL: Small, Medium, Large, Extra Large

XS, S, M, L, XL: Extra Small, Small, Medium, Large, Extra Large

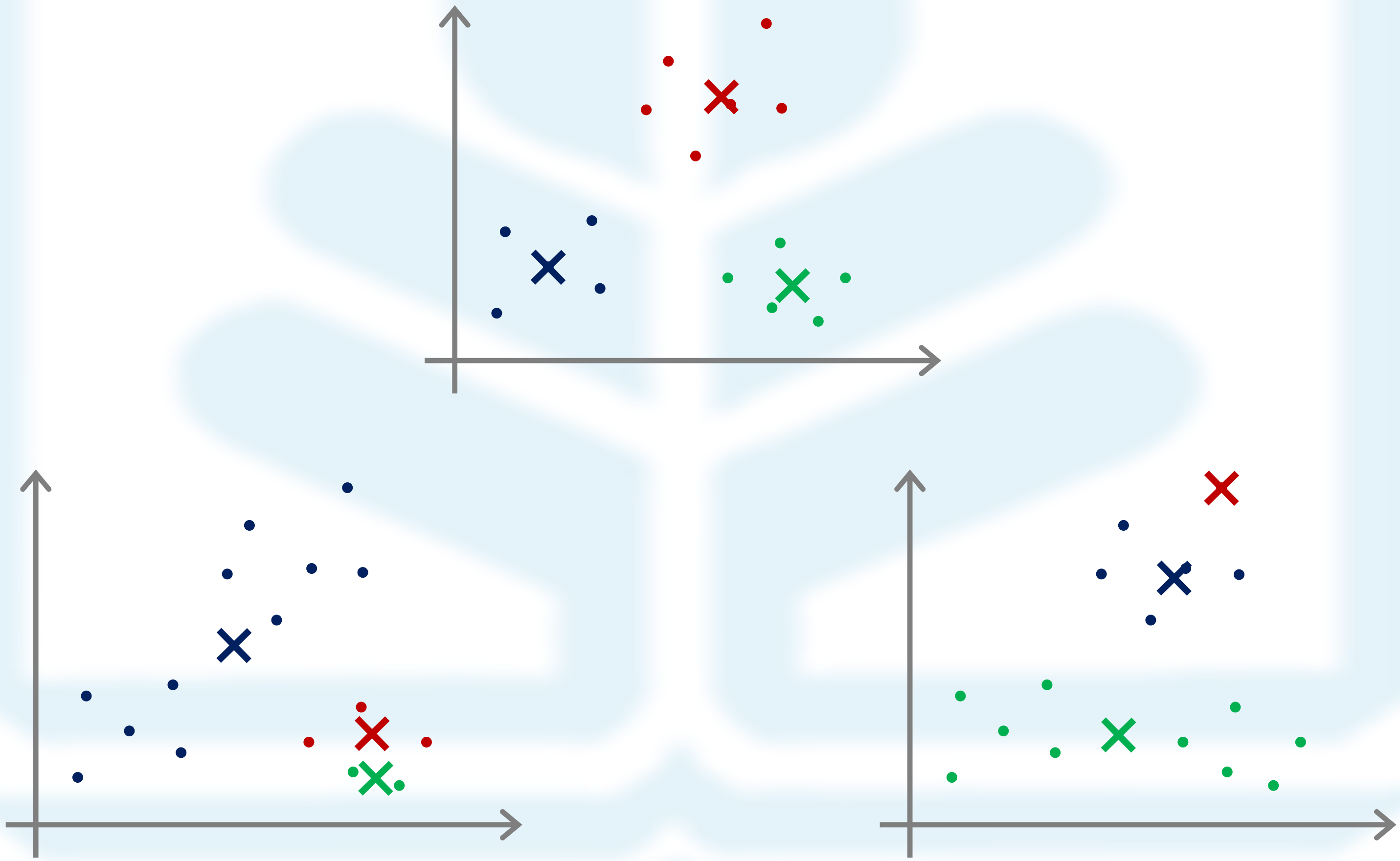
K-means Algorithm

Choosing the value of K, Elbow method:



K-means Algorithm

Local Optima



K-means Algorithm

Local Optima

Random initialization

For i = 1 to 100 {

Randomly initialize K-means.

Run K-means, get $c^{(1)}, c^{(2)}, \ldots, c^{(m)}, \mu_1, \mu_2, \ldots, \mu_K$.

Compute cost function (distortion), $J\left(c^{(1)}, c^{(2)}, \ldots, c^{(m)}, \mu_1, \mu_2, \ldots, \mu_K\right)$

}

Pick clustering that gave lowest cost, distortion.

K-means Algorithm

K-means Algorithm

Input: Number of clusters (K)

Training set $\{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$, $x^{(i)} \in \mathbb{R}^n$ for i=1,2,…, m

Output:

$c^{(i)}$: =index (from 1 to K) of cluster centroid closest to $x^{(i)}$.

Step 1. Randomly initialize K cluster centroids $\mu_1, \mu_2, \dots, \mu_K \in \mathbb{R}^n$.

- Randomly pick K training examples.
- Set $\mu_1, \mu_2, \dots, \mu_K$ equal to these K examples

Step 2.

Repeat until convergence{

for i=1:m

$$c^{(i)} = \underset{k}{\operatorname{argmin}} \left\| \boldsymbol{x}^{(i)} - \mu_k \right\|^2$$

for k=1:K

$$\mu_k = \frac{\sum_{i=1}^{m} 1_{\{c^{(i)}-k\}}\ \boldsymbol{x}^{(i)}}{\sum_{i=1}^{m} 1_{\{c^{(i)}-k\}}}$$

(μ_k is average (mean) of points assigned to cluster k.)

}

$$1_{\{c^{(i)}-k\}} = \begin{cases} 1, & \text{if } c^{(i)} = k \\ 0, & \text{o.w.} \end{cases}$$

Stopping Criteria: In order to see if the algorithm converges, we look at the **distortion function** defined as follows:

$$J(\boldsymbol{c}, \boldsymbol{\mu}) = \sum_{i=1}^{m} \left\| \boldsymbol{x}^{(i)} - \mu_{c^{(i)}} \right\|^2$$

* Depending on the initialization of cluster centroids K-means can produce different results