Machine Learning (CE 40477)

Fall 2024

Ali Sharifi-Zarchi

CE Department
Sharif University of Technology

September 26, 2024

Unsupervised Learning

Unsupervised Learning involves working with **unlabeled data**, where the goal is to **infer the natural structure** present within a set of data points.

- Learning from unlabeled data.
- Most of the times, there is no (or minimal) prior knowledge of the data.
- Two of the most common techniques:
  - **Clustering**: Grouping data points into clusters based on similarity towards user need.
  - **Dimensionality Reduction**: Reducing the number of features under consideration and keeping (perhaps approximately) the most informative features.

Clustering: Bio-informatics

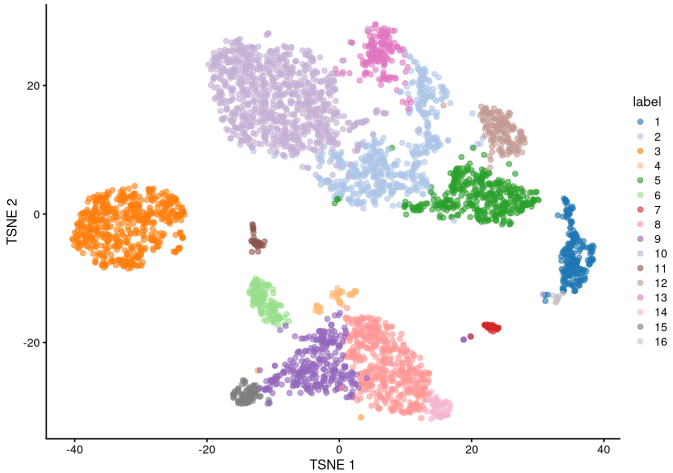
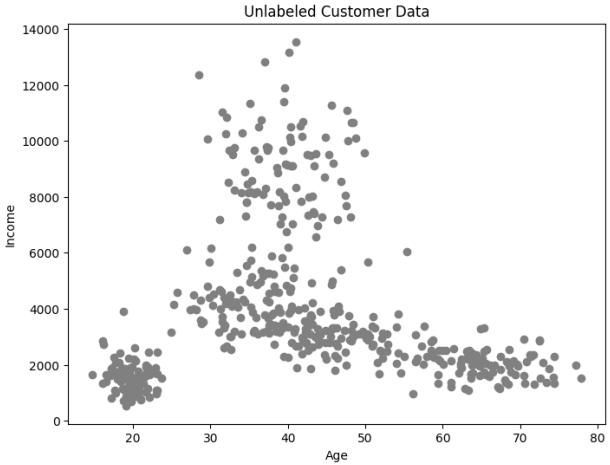


Figure adapted from bioconductor.org

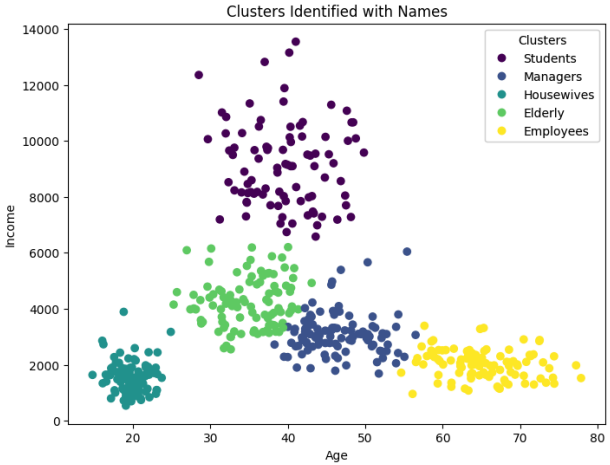
Slide showcasing reduction (PCA)

Clustering: Customer segmentation



Information of customers

Clustering: Customer segmentation (cont.)



Predicted clusters of customers

Clustering

- Assume we have a set of unlabeled data points $\{\mathbf{x}^{(i)}\}_{i=1}^{N}$.
- We intend to find **groups of similar objects** with respect to our need.
  - For example all data points having most similar number of buys in a market.
- It helps us to gain insight into structure of data prior to class design.
- Clustering could also help to compress and reduce data.

Clustering (cont.)

From another point of view, clusters are regions of high density that are separated from one another with regions of low density.

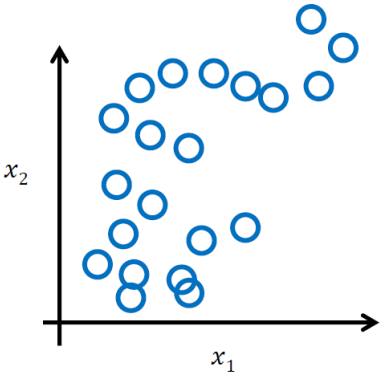


Figure adap

Hard clustering vs Soft clustering

- **Hard Clustering:** Each data point belongs to exactly one cluster
  - more common and easier to do
- **Soft Clustering**

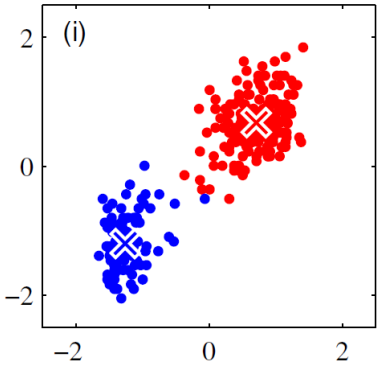


Figure adap

Hard clustering vs Soft clustering (cont.)

- **Hard Clustering**
- **Soft Clustering:** Each data point can belong to multiple clusters.
  - data point belongs to each cluster with a probability
- **From now on, we will focus on problem of hard clustering**

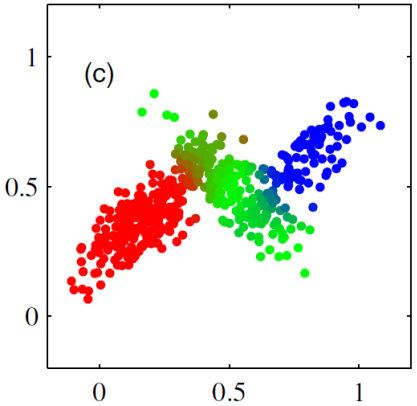


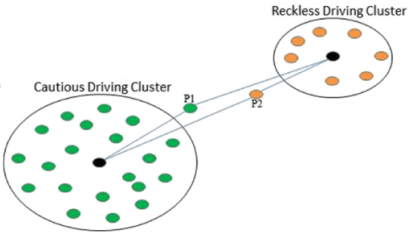
Figure adap

Hard clustering problem

- We need to **partition** N data points to K clusters.
- Good flat clustering should have two important factors:
  - Data points in a cluster should be similar (**High intra-cluster similarity**)
  - Data points in different clusters should be less similar (**Low inter-cluster similarity**)
- As mentioned earlier, each partitional clustering method requires a **similarity metric** between data points.

Similarity measure and Distance measure

- Similarity measures are used to distinguish between similar and non-similar data points.
- We usually define similarity of two data point as **inverse of distance** between them.
- Using this definition, hard clustering aims to put data points with less distance in same cluster.



Common similarity and distance measures

- Assume p and q are two data points from $\mathbb{R}^D$. most common similarity and distance measures in the problem of clustering are as follows:
  - **Euclidean distance:** Most common measure of distance between two vectors doesn't matter.

$$d^2(p,q) = \sqrt{\sum_{i=1}^{D} (p_i - q_i)^2}$$

  - **Cosine similarity:** Most common measure of similarity when the magnitude of vectors does not change the similarity

$$\textbf{similarity}(p,q) = \frac{p^T q}{||p|| \,.\, ||q||}$$

  - **Manhattan distance:** Most common measure of distance when dimensions are not equally important

$$d^2(p,q) = \sqrt{\sum_{i=1}^{D} |p_i - q_i|}$$

K-Means overview

- One of the most common partitional clustering methods used
- The idea is to **find K centers. Each center representing a cluster.**
- Each data point is assigned to cluster j if and only if it has the least distance to center of cluster j amongst all clusters.
- K-Means suggest an **iterative algorithm** to find these centers.

K-Means Clustering

- **Input:** a set $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(N)}$ of data points ($\forall x^{(i)} \in \mathbb{R}^D$ and an integer K
- **Output:** a set of K representatives $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_K$ as the cluster representatives
  - data points are assigned to the clusters according to their distance to $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_K$
  - each data is assigned to the cluster whose representative is nearest to it
- **Objective:** choose $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_K$ as to minimize intra-cluster similarity:

$$\sum_{i=1}^{N} min_{j \in 1,2,\ldots,K} d^2 \left(\mathbf{x}^{(i)}, \mathbf{c_j} \right)$$

K-Means Clustering (cont.)

- K-Means uses **Euclidean Distance** measure thus we can rewrite the objective as follows:

$$J(\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_K) = \sum_{i=1}^{N} min_{j \in 1,2,\ldots,K} \left|\left| \mathbf{x}^{(i)} - \mathbf{c}_j \right|\right|^2$$

- This objective function is sometimes called **distortion** as well.

K-Means Clustering (cont.)

- **Why the idea of minimizing distortion works ?**
  - Distortion is used to model intra-cluster similarity score. If we can show that $J(\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_K)$ is optimizable, we can get suggest K-Means actually works.
  - We just have to make sure each iteration of reaching optimum centers, is decreasing distortion. (or at least doesn't increase it)
  - Then simply use optimization methods to reach optimum or at least, get as close as possible to it.

K-Means pseudo-code

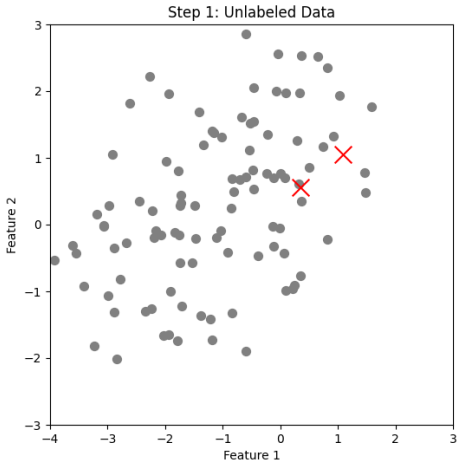
Algorithm 1 K-Means

1: Select k random points $c_1, c_2, \ldots, c_k$ as clusters' initial centroids
2: **repeat**
3: **for** each $i = 1$ to N **do**
4: Assign $x^{(i)}$ to the closest cluster C_j such that C_j contains all data that are closer to c_j than to any other cluster.
5: **end for**
6: **for** each $j = 1$ to k **do**
7: $c_j = \frac{1}{|C_j|} \sum_{x^{(i)} \in C_j} x^{(i)}$
8: **end for**
9: **until** the centroids no longer change

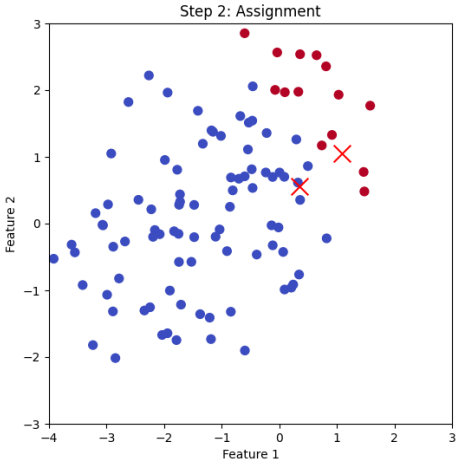
K-Means pseudo-code (cont.)

- It is notable that K-Means follows the following two steps until reaching the best clustering status:
  1. **Assigning data points to the closest cluster**
  2. **Computing new center for each cluster**

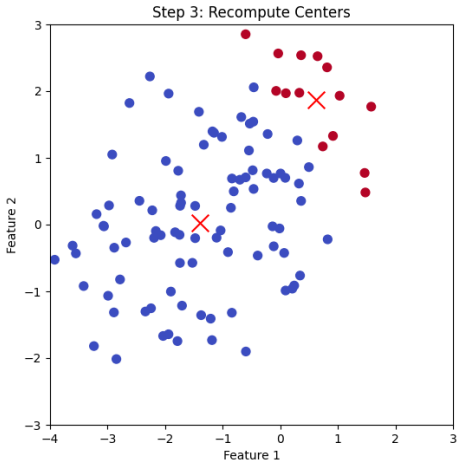
K-Means in action



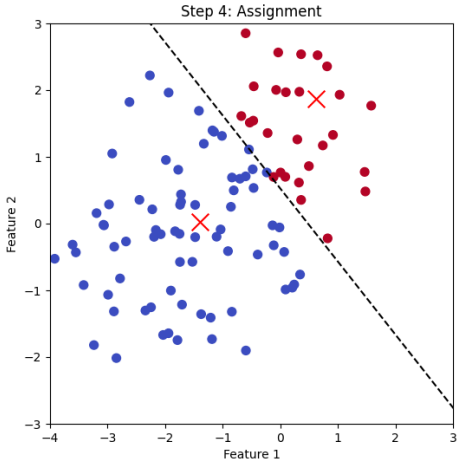
K-Means in action (cont.)



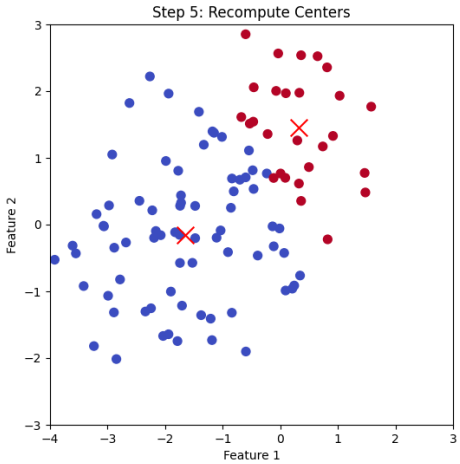
K-Means in action (cont.)



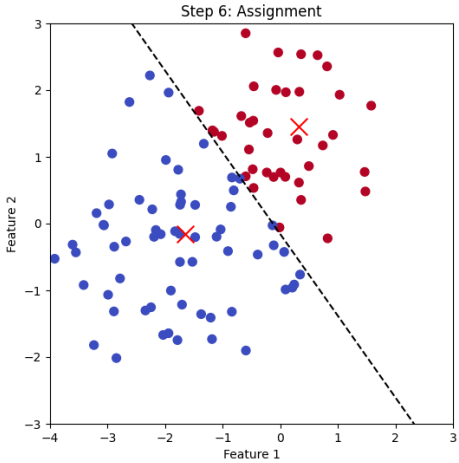
K-Means in action (cont.)



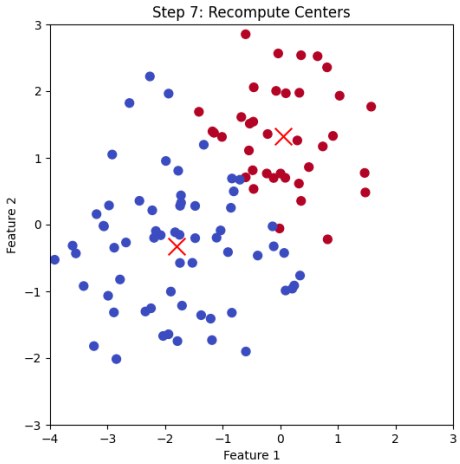
K-Means in action (cont.)



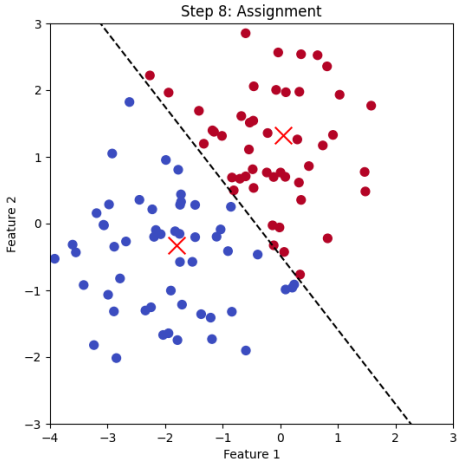
K-Means in action (cont.)



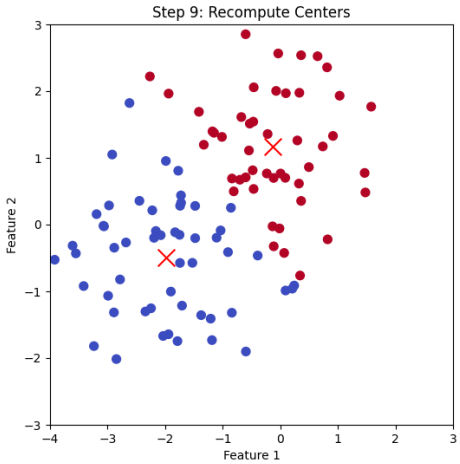
K-Means in action (cont.)



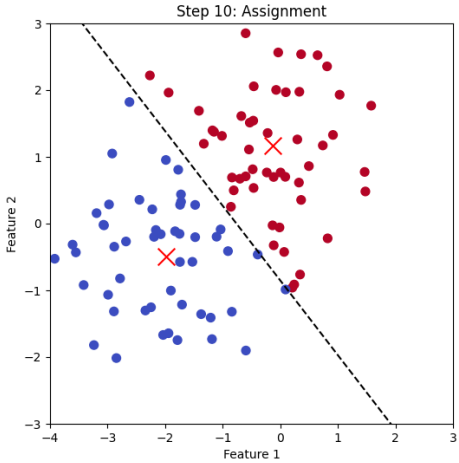
K-Means in action (cont.)



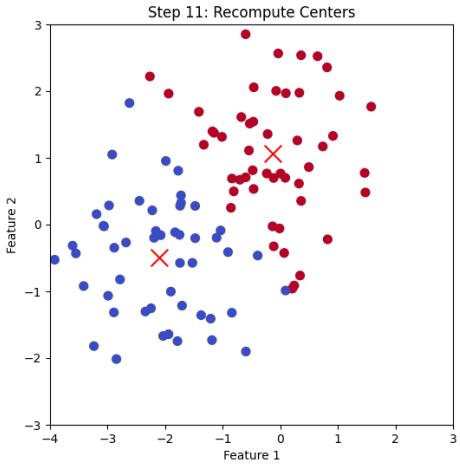
K-Means in action (cont.)



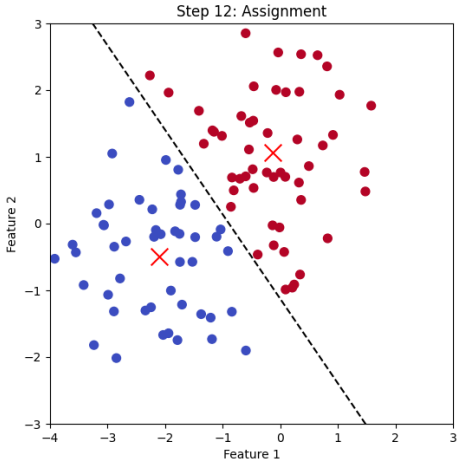
K-Means in action (cont.)



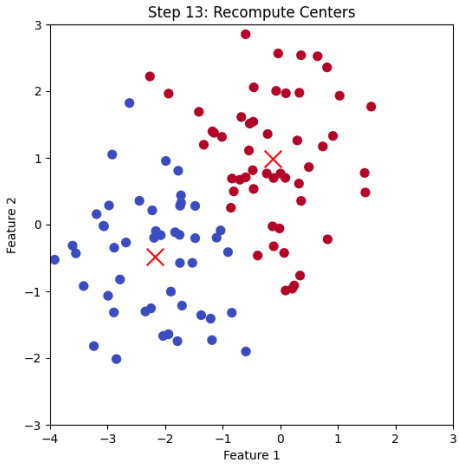
K-Means in action (cont.)



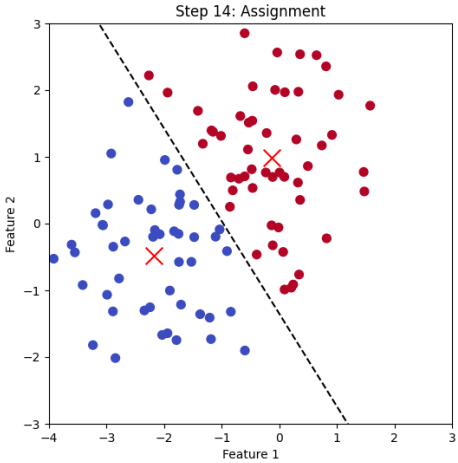
K-Means in action (cont.)



K-Means in action (cont.)



K-Means in action (cont.)



Steps of K-Means

- **Why the steps of K-Means have been chosen as explained ?**
- Let us rewrite distortion as follows:

$$J(\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_K) = \sum_{i=1}^{N} \sum_{j=1}^{K} r_{ij} \, ||\mathbf{x}^{(i)} - \mathbf{c}_j||^2$$

- In which r_{ij} is an indicator defined as:

$$r_{ij} = \begin{cases} 1 & \mathbf{x}^{(i)} \\ & is \;\; assigned \;\; to \;\; cluster \;\; j \\ 0 & o.w. \end{cases}$$

Steps of K-Means (cont.)

- Let us first assume K centers are fixed. We need to optimize J with respect to r_{ij}.
- Because J is a linear function of r_{ij}, this optimization can be performed easily to give a closed form solution.
- The terms involving different n are independent and so we can optimize for each n separately by choosing r_{ij} to be 1 for whichever value of k gives the minimum value of $||\mathbf{x}^{(i)} - \mathbf{c}_j||$
- r_{ij} can be written as follows:

$$r_{ij} = \begin{cases} 1 & if \;\; j = arg\; min_j \; ||\mathbf{x}^{(i)} - \mathbf{c}_j|| \\ 0 & o.w. \end{cases}$$

Steps of K-Means (cont.)

- Now let us assume one step of assignment has been performed. We need to optimize J with respect to $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_K$
- Objective function J is quadratic with respect to each $\mathbf{c}_j$, thus can be solved by setting it's partial derivatives equal to zero:

$$\frac{\partial J}{\partial \mathbf{c}_j} = 0 \Longrightarrow 2\sum_{i=1}^{N} r_{ij}\left(\mathbf{x}^{(i)} - \mathbf{c}_j\right) = 0$$

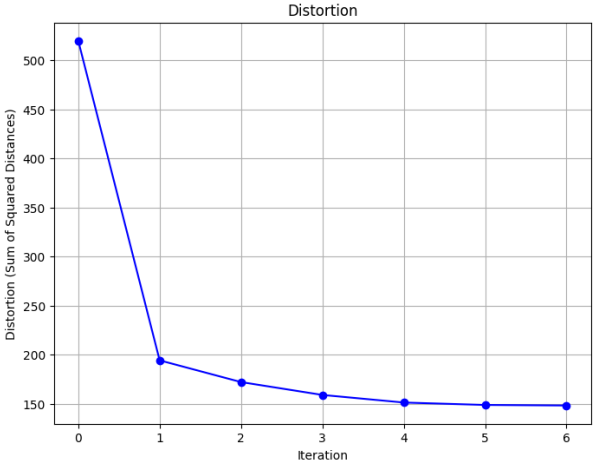
- Solving the equations above gives us:

$$\mathbf{c}_j = \frac{\sum_{i=1}^{N} r_{ij}\mathbf{x}^{(i)}}{\sum_{i=1}^{N} r_{ij}}$$

k-Means convergence

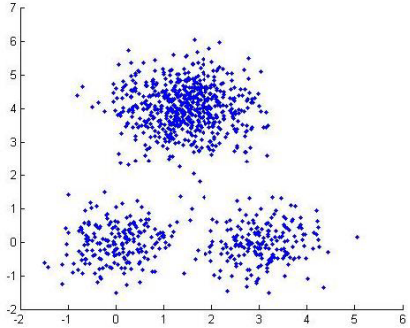
- It always converges.
- Why should the K-Means algorithm ever reach a state in which clustering doesn't change ?
  - We have shown reassignment step monotonically decreases J since each data point is assigned to the nearest cluster.
  - We have also proven center updates also minimizes sum of squared distances of the assigned data points to the cluster from its center.

K-Means convergence (cont.)

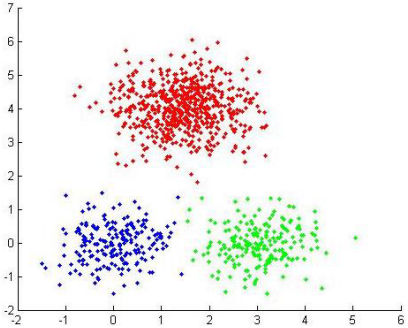


Local optimum

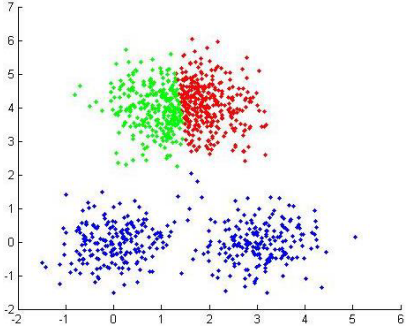
- k-Means always converges.
- However; it may converge at local optimum that is different from the global optimum in terms of objective score.



Local optimum (cont.)



Optimal clustering



Possible clustering

K-Means limitations

- Initialization is crucial as it can determine how fast the algorithm converges.
- To overcome the problem of local minima, there are numerous solutions:
  - Selecting random centers from data points
  - Initialize with the suggested results of another method
  - Use heuristics to find good initial centers
    - K-Means++
    - Furthest point
- Often, k-Means fails to find clusters of arbitary shapes and sizes.
  - Except to very distant clusters.

How many clusters?

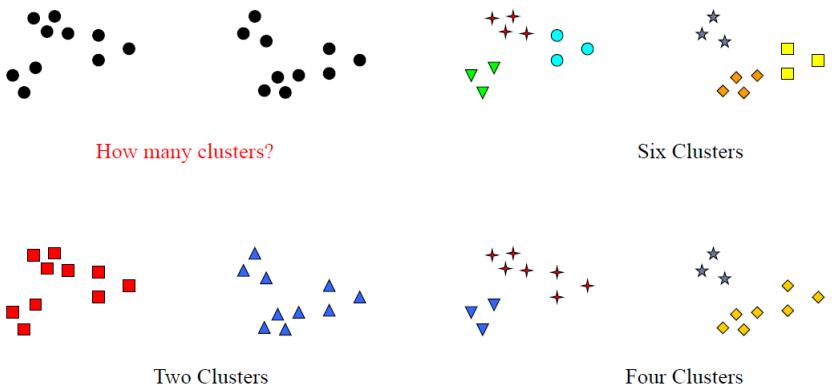


Figure adapted from slides of Dr. Soleymani, Modern Information Retrieval Course, Sharif University of technology.

How many clusters? (cont.)

- Number of clusters is given in advance in the problem of clustering. However; finding the **right** number of clusters is also a problem.
- There is a tradeoff between having better focus within each cluster or having too many clusters.
- **Optimization problem:** penalize having too much clusters
  - Application dependent

$$K^* = arg\ min_k\ J(k) + \lambda k$$

External criteria

External clustering criteria (purity, r-index, NMI, F-measure)

Contributions

- **This slide has been prepared thanks to:**

[1] C. M., *Pattern Recognition and Machine Learning*.
Information Science and Statistics, New York, NY: Springer, 1 ed., Aug. 2006.

[2] M. Soleymani Baghshah, "Machine learning." Lecture slides.

[3] A. Ng and T. Ma, *CS229 Lecture Notes*.

[4] T. Mitchell, *Machine Learning*.
McGraw-Hill series in computer science, New York, NY: McGraw-Hill Professional, Mar. 1997.

[5] Y. S. Abu-Mostafa, M. Magdon-Ismail, and H.-T. Lin, *Learning From Data: A Short Course*.
New York, NY: AMLBook, 2012.

[6] S. Goel, H. Bansal, S. Bhatia, R. A. Rossi, V. Vinay, and A. Grover, "CyCLIP: Cyclic Contrastive Language-Image Pretraining," *ArXiv*, vol. abs/2205.14459, May 2022.