Clustering

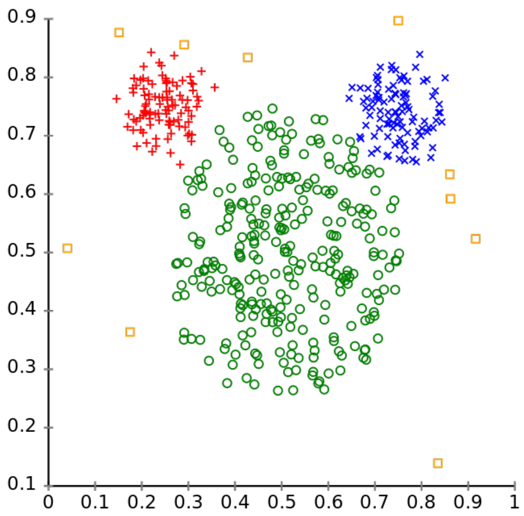
Reference book: Bishop: “Pattern Recognition and Machine Learning” Chapter 9.1

Clustering

- Unsupervised learning
- Generating “classes”
- Distance/similarity measures
- Agglomerative methods
- Divisive methods

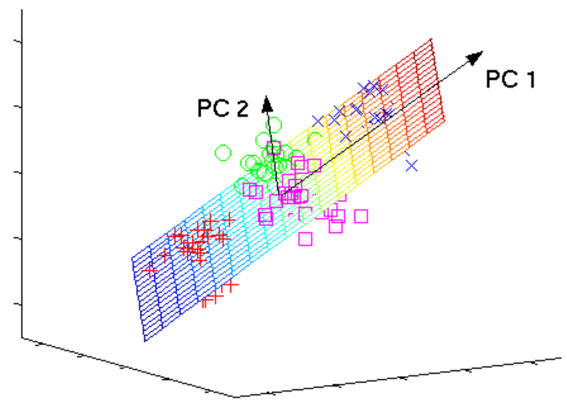
Unsupervised Learning

- No labels/responses. Finding structure in data.
- Dimensionality Reduction.



Clustering

$T: \mathbb{R}^d \rightarrow \{1, 2, \dots, K\}$

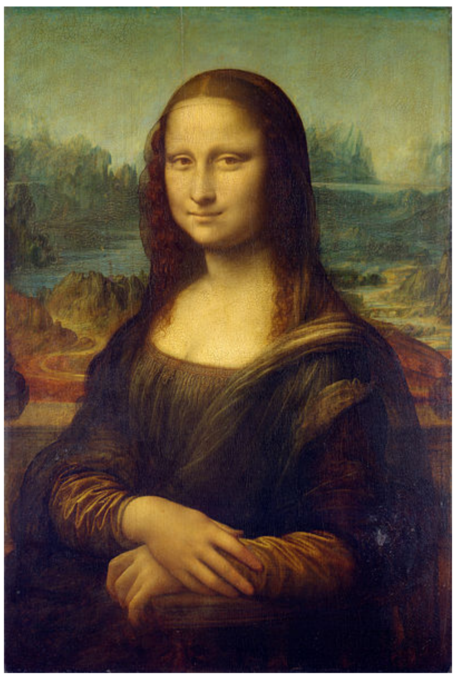


Subspace Learning

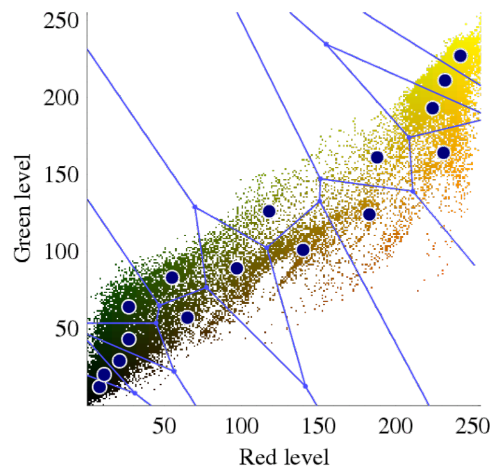
$T: \mathbb{R}^d \rightarrow \mathbb{R}^m, m < D$

Uses of Unsupervised Learning

- Data compression



RGB
(12,165,73)
(11,167,79)
:



Labels
3
43
:

Dictionary
1 ~ (10, 160, 70)
2 ~ (40, 240, 20)
:

Uses of Unsupervised Learning

- To improve classification/regression (semi-supervised learning)

1. From unlabeled data, learn a good feature $T: \mathbb{R}^d \to \mathbb{R}^m$.
2. To labeled data, apply transformation $T: \mathbb{R}^d \to \mathbb{R}^m$.

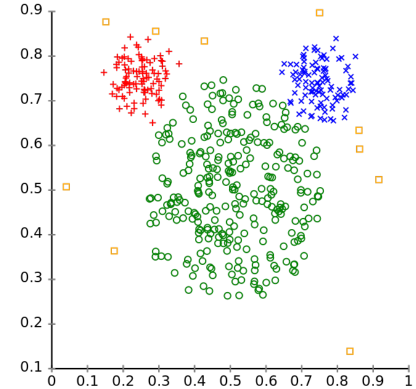
$$(T(\boldsymbol{x}_1), \boldsymbol{y}_1), \ldots, (T(\boldsymbol{x}_N), \boldsymbol{y}_N)$$

3. Perform classification/regression on transformed low-dimensional data.

What is Clustering?

- Unsupervised learning - no information from teacher
- The process of partitioning a set of data into a set of meaningful (hopefully) sub-classes, called clusters
- Cluster:
  - collection of data points that are “similar” to one another and collectively should be treated as group (*it is not the group we learned in linear algebra*)
  - as a collection, are sufficiently different from other groups

What is Clustering



Clustering Problem.

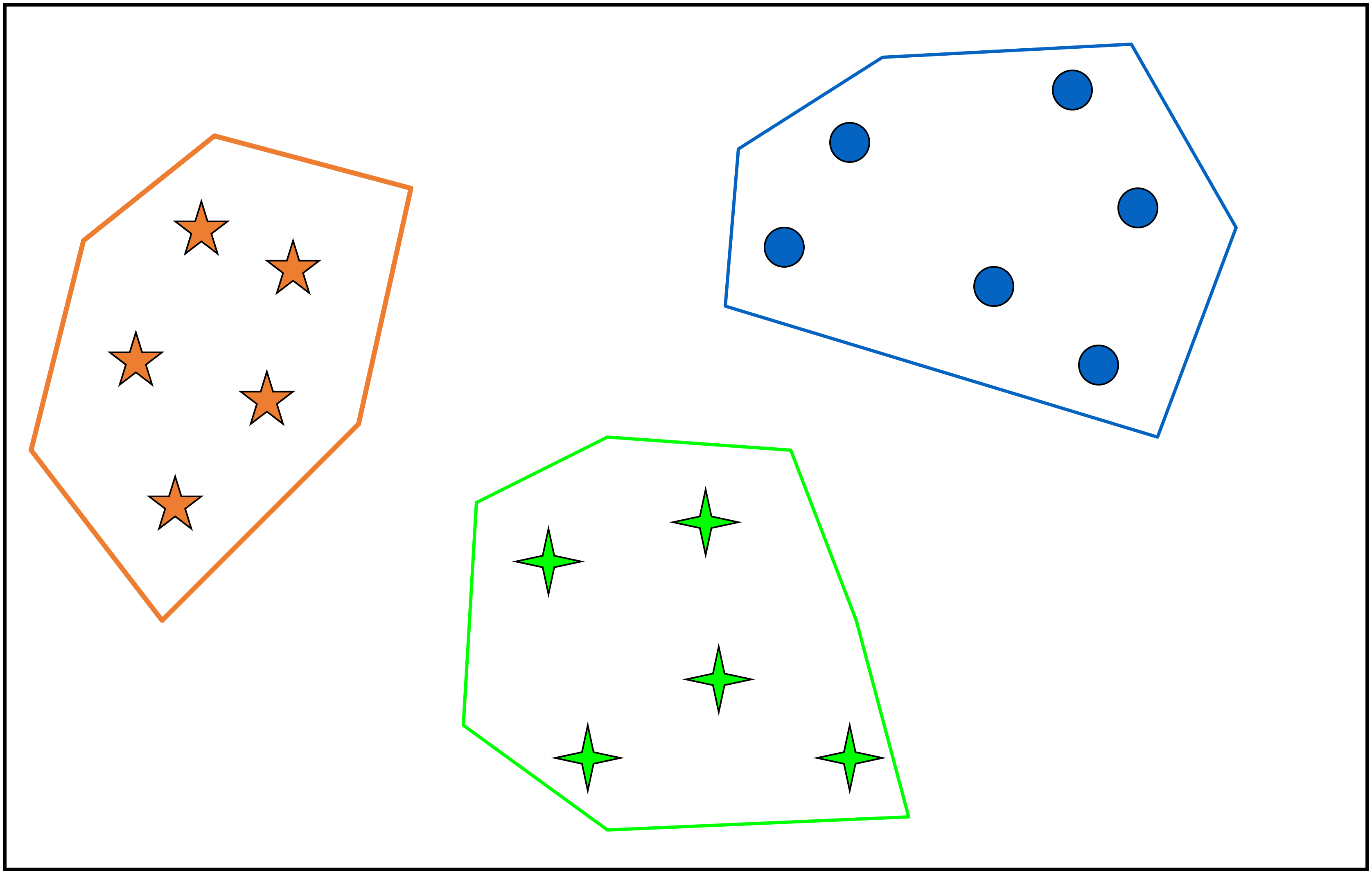
Input. Training data $\mathcal{S}_N = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N\}, \quad \boldsymbol{x}_n \in \mathbb{R}^d$.
Integer K

Output. Clusters $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_K \subset \{1, 2, \dots, N\}$ such that every data point is in one and only one cluster.

Cluster representatives $\{\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_K\}$

Some clusters could be empty!

Clusters

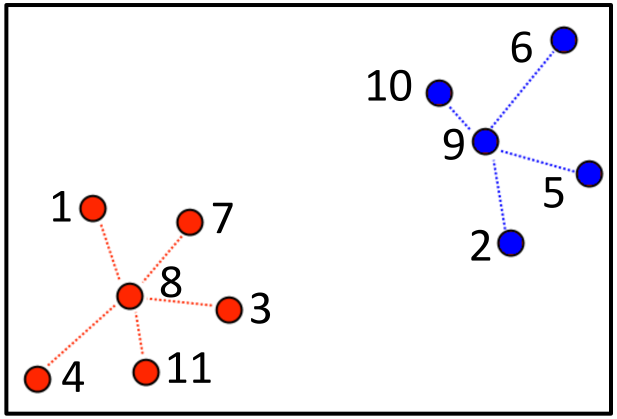


How to Specify a Cluster

- By listing all its elements

$$\mathcal{C}_1 = \{1,3,4,7,8,11\}$$

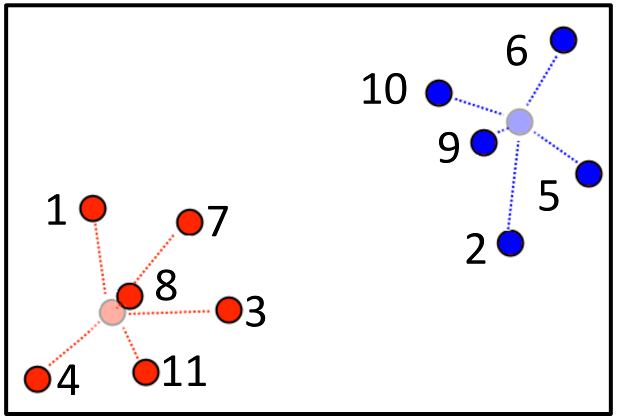
$$\mathcal{C}_2 = \{2,5,6,9,10\}$$



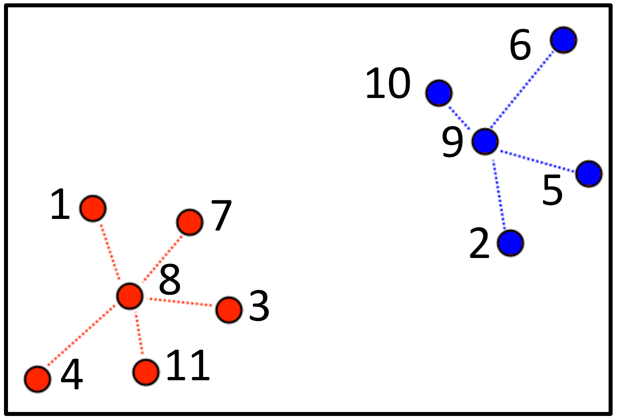
How to Specify a Cluster

- Using a representative
    a. A point in center of cluster (centroid)
    b. A point in the training data (exemplar)

Each point x_i will be assigned the closest representative.



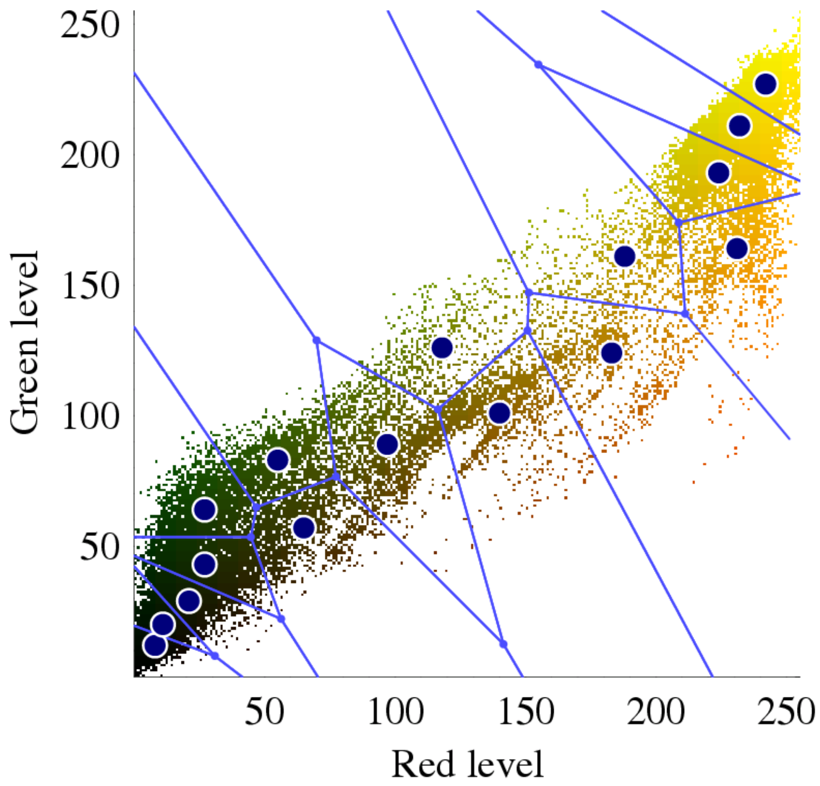
centroid



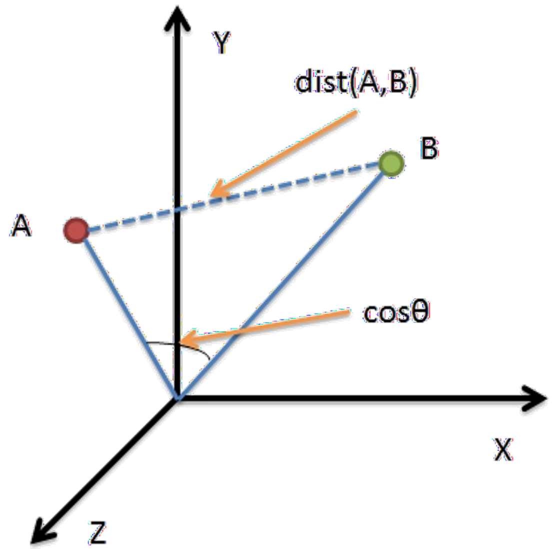
exemplar

Voronoi Diagram

We can partition all the points in the space into regions, according to their closest representative.



Distance/Similarity Measures



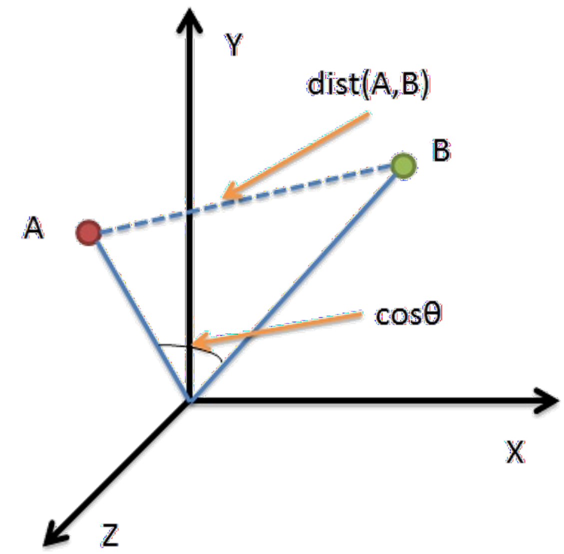
(sometimes called loss functions*)*

A measure of how close two data points are. Nearby points (i.e. distance is small) are more likely to belong to the same cluster.

- Euclidean Distance

$$\text{dist}(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|_2 := \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Distance/Similarity Measures



A measure of how alike two data points are. Similar points (i.e. similarity is large) are more likely to belong to the same cluster.

- Cosine Similarity $\cos(\boldsymbol{x}, \boldsymbol{y}) = \dfrac{\boldsymbol{x}^\top \boldsymbol{y}}{\|\boldsymbol{x}\| \, \|\boldsymbol{y}\|}$

Distance (Similarity) Matrix

- Similarity (Distance) Matrix
  - based on the distance or similarity measure we can construct a symmetric matrix of distance (or similarity values)
  - (*i*, *j*)th entry in the matrix is the distance (similarity) between items *i* and *j*

$$
\begin{array}{c|cccc|}
 & I_1 & I_2 & \cdots & I_n \\
\hline
I_1 & \bullet & d_{12} & \cdots & d_{1n} \\
I_2 & d_{21} & \bullet & \cdots & d_{2n} \\
\vdots & \vdots & \vdots & \bullet & \vdots \\
I_n & d_{n1} & d_{n2} & \cdots & \bullet \\
\hline
\end{array}
$$

$$d_{ij} = \text{similarity (or distance) of } D_i \text{ to } D_j$$

Note that $d_{ij} = d_{ji}$ (i.e., the matrix is symmetric). So, we only need the lower triangle part of the matrix.

The diagonal is all 1's (similarity) or all 0's (distance)

Example: Term Similarities in Documents

	T1	T2	T3	T4	T5	T6	T7	T8
Doc1	0	4	0	0	0	2	1	3
Doc2	3	1	4	3	1	2	0	1
Doc3	3	0	0	0	3	0	3	0
Doc4	0	1	0	3	0	0	2	0
Doc5	2	2	2	3	1	4	0	2

$$\boxed{sim(T_i, T_j) = \sum_{k=1}^{N} (w_{ik} \cdot w_{jk})}$$

Term-Term Similarity Matrix

	T1	T2	T3	T4	T5	T6	T7
T2	7						
T3	16	8					
T4	15	12	18				
T5	14	3	6	6			
T6	14	18	16	18	6		
T7	9	6	0	6	9	2	
T8	7	17	8	9	3	16	3

Similarity (Distance) Thresholds

- A similarity (distance) threshold may be used to mark pairs that are “sufficiently” similar

	T1	T2	T3	T4	T5	T6	T7
T2	7						
T3	16	8					
T4	15	12	18				
T5	14	3	6	6			
T6	14	18	16	18	6		
T7	9	6	0	6	9	2	
T8	7	17	8	9	3	16	3

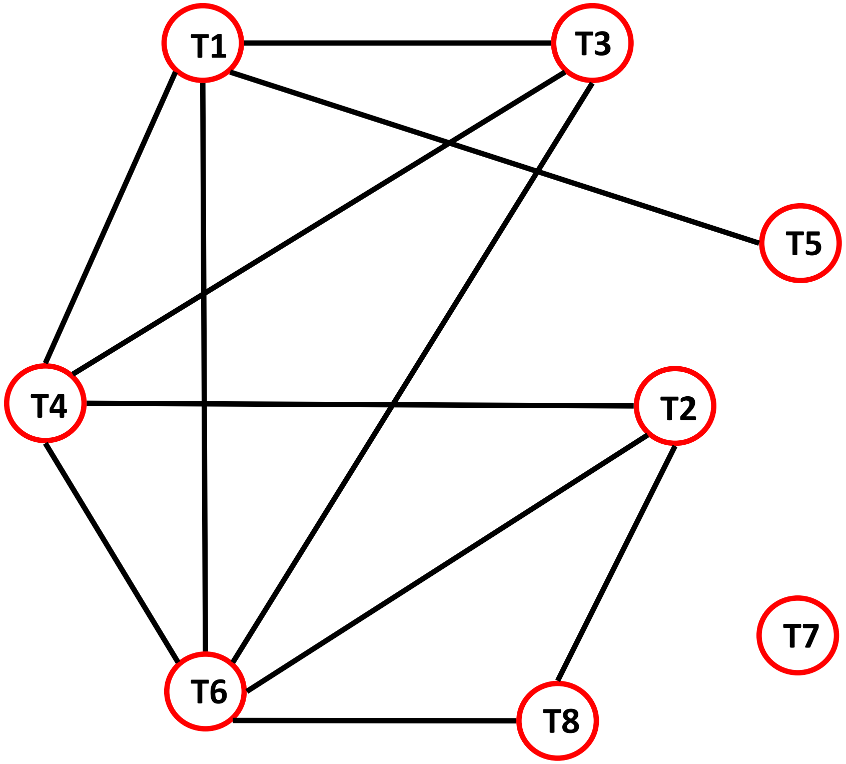
	T1	T2	T3	T4	T5	T6	T7
T2	0						
T3	1	0					
T4	1	1	1				
T5	1	0	0	0			
T6	1	1	1	1	0		
T7	0	0	0	0	0	0	
T8	0	1	0	0	0	1	0

Using a threshold value of 10 in the previous example

Graph Representation

- The similarity matrix can be visualized as an undirected graph
  - each item is represented by a node, and edges represent the fact that two items are similar (a 1 in the similarity threshold matrix)

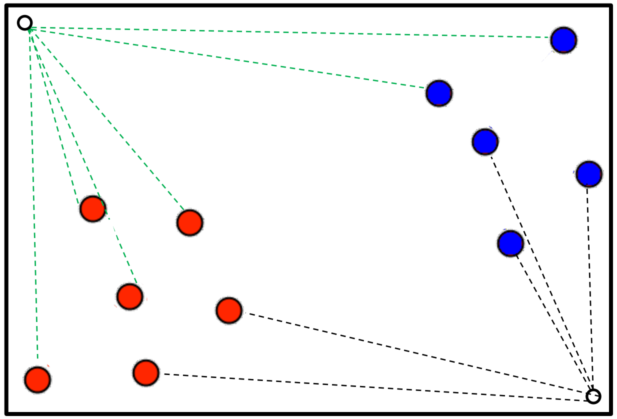
	T1	T2	T3	T4	T5	T6	T7
T2	0						
T3	1	0					
T4	1	1	1				
T5	1	0	0	0			
T6	1	1	1	1	0		
T7	0	0	0	0	0	0	
T8	0	1	0	0	0	1	0



If no threshold is used, then matrix can be represented as a weighted graph

Training Loss

Sum of squared distances to closest representative.

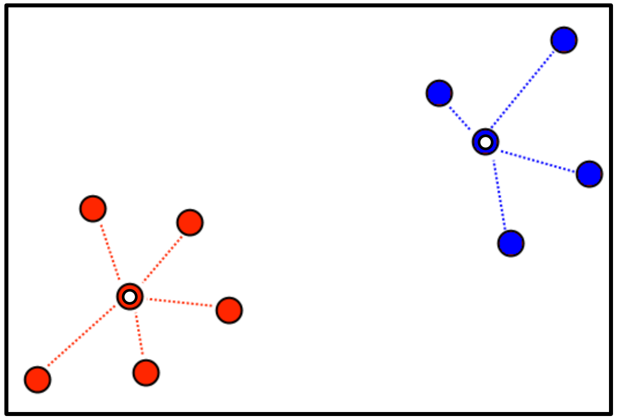


$\text{loss} \approx 11 \times (1)^2 = 11$

assume length of each
edge is about 1

Training Loss

Sum of squared distances to closest representative (cluster center).



$\text{loss} \approx 9 \times (0.1)^2 = 0.09$

assume length of each
edge is about 0.1

Training loss

Optimizing both clusters and representatives.

$$\mathcal{L}(\boldsymbol{R}, \boldsymbol{M}; \mathcal{S}_n) = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|^2$$

where

$\mathcal{S}_n = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N\}, \ \boldsymbol{x}_i \in \mathbb{R}^d$

$r_{nk} \in \{0,1\}$ denotes which of the K clusters data point $\boldsymbol{x}_n$ is assigned to. If $\boldsymbol{x}_n$ is assigned to cluster k then $r_{nk} = 1$, and $r_{nj} = 0$ if $j \neq k$.

$\boldsymbol{R} = \{r_{nk}\} \in \{0,1\}^{n \times k}, n = 1, \dots, N, k = 1, \dots, K$

$\boldsymbol{M} = [\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_K]^{\mathrm{T}}$

Goal: find values for $\boldsymbol{R}$ and $\boldsymbol{M}$ so as to minimize $\mathcal{L}$.

Basic Clustering Methodology

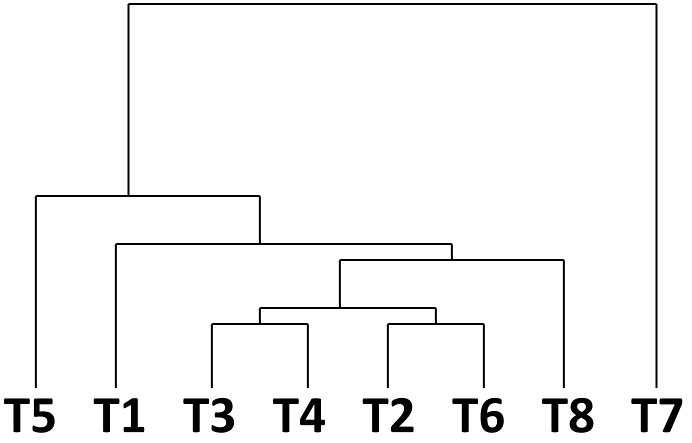
Two approaches:

- Agglomerative: pairs of items/clusters are successively linked to produce larger clusters
- Divisive (partitioning): items are initially placed in one cluster and then divided into separate groups

Hierarchical/Agglomerative Methods

- Based on some methods of representing hierarchy of data points
- One idea: hierarchical dendogram (connects points based on similarity)

	T1	T2	T3	T4	T5	T6	T7
T2	7						
T3	16	8					
T4	15	12	18				
T5	14	3	6	6			
T6	14	18	16	18	6		
T7	9	6	0	6	9	2	
T8	7	17	8	9	3	16	3



Hierarchical Agglomerative

- Compute distance matrix
- Put each data point in its own cluster
- Find most similar pair of clusters
  - merge pairs of clusters (show merger in dendogram)
  - update similarity matrix
  - repeat until all data points are in one cluster

K-Means

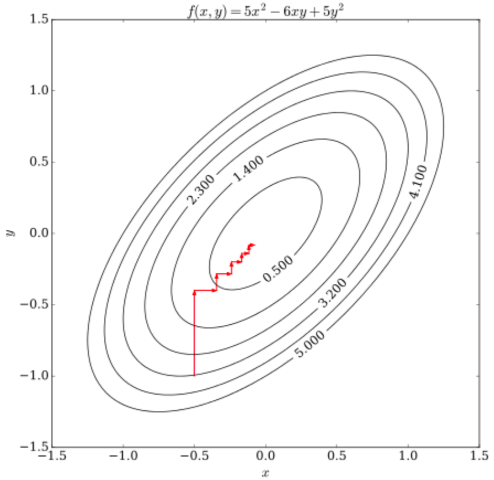
Optimization Algorithm

$$\mathcal{L}(\boldsymbol{R}, \boldsymbol{M}; \mathcal{S}_n) = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|^2$$

Goal. Minimize $\mathcal{L}(x, y)$.

Coordinate Descent (Optimization).

Repeat until convergence:
1. Find optimal x while holding y constant.
2. Find optimal y while holding x constant.



Coordinate descent is an optimization algorithm that successively minimizes along coordinate directions to find the minimum of a function.

Optimization Algorithm

Coordinate Descent (Optimization)

Repeat until convergence:

- Find best clusters given centres
- Find best centres given clusters

$$\mathcal{L}(\boldsymbol{R}, \boldsymbol{M}; \mathcal{S}_n) = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|^2$$

Optimization Algorithm

1. Initialize centers $\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_K$ from the data.
2. Repeat until no further change in training loss:

   a. For each $n \in \{1, \dots, N\}$,

   $$r_{nk} = \begin{cases} 1, \text{if } k = \arg\min_j \left\| \boldsymbol{x}_n - \boldsymbol{\mu}_j \right\|^2 \\ 0, \text{ otherwise.} \end{cases}$$

   We assign the nth data point to the closest cluster centre.

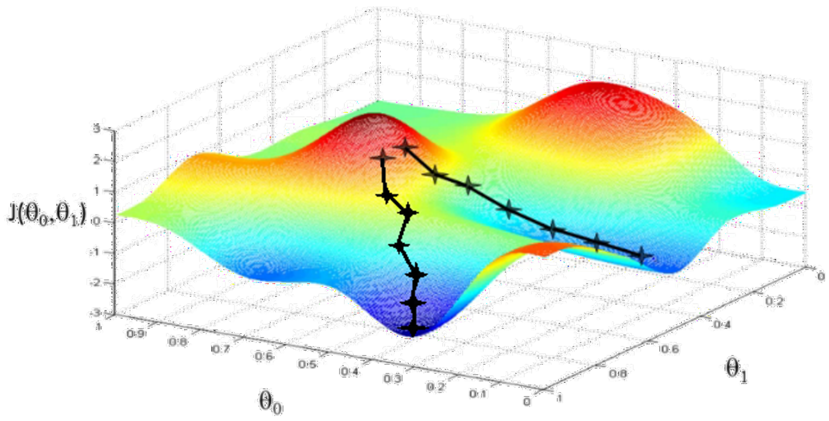
   b. For each $j \in \{1, \dots, k\}$,

   $$\boldsymbol{\mu}_k = \frac{\sum_n r_{nk} \boldsymbol{x}_n}{\sum_n r_{nk}}$$

   We recompute cluster means.

Convergence

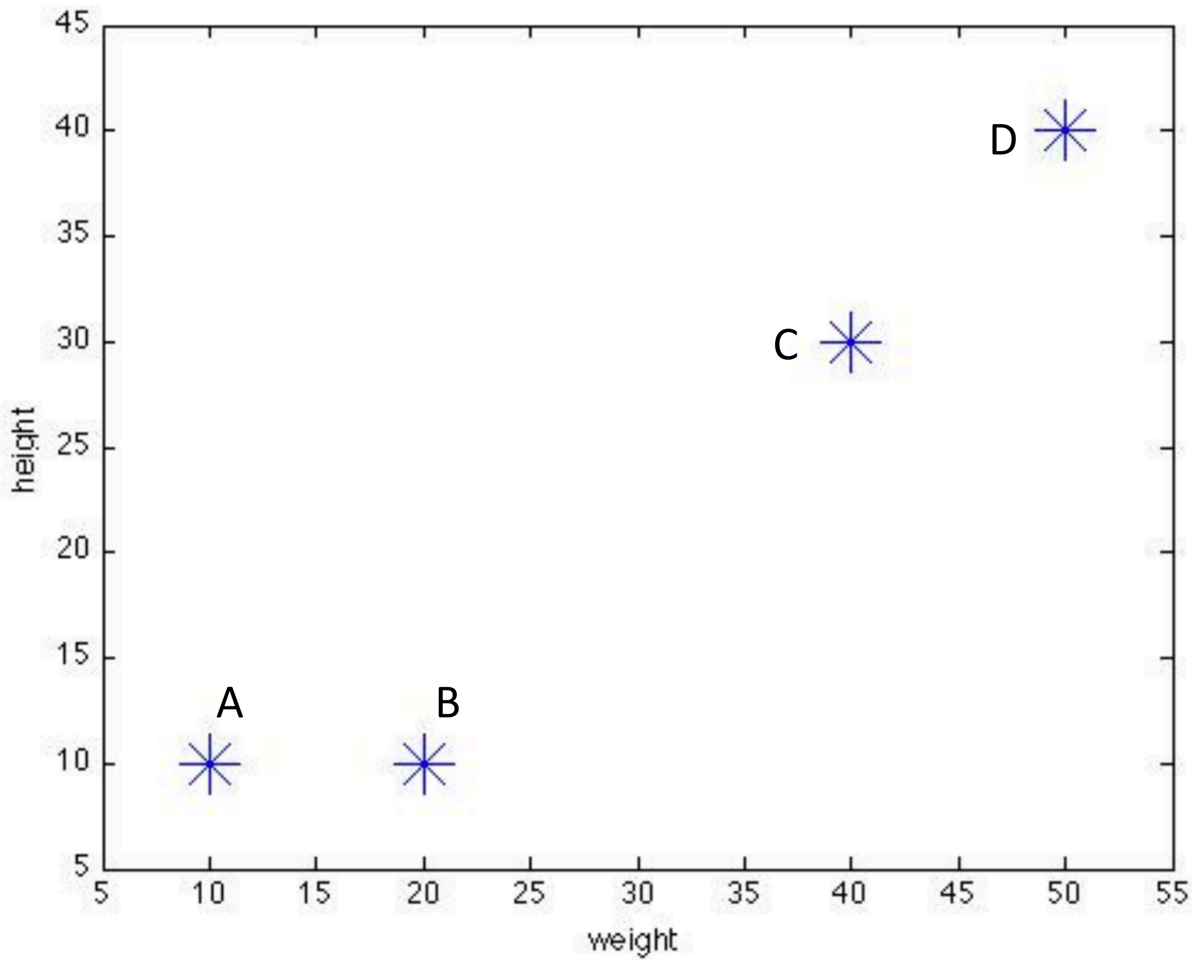
- Training loss always decreases in each step (coordinate descent).
- Converges to local minimum, not necessarily global minimum.



Repeat algorithm over many initial points, and pick the configuration with the smallest training loss.

An example – kmeans clustering

- Suppose we have 4 boxes of different sizes and we want to divide them into 2 classes
- Each box represents one point with two attributes (w,h):



- Initial centers: suppose we choose points A and B as the initial centers, so c1 = (10, 10) and c2 = (20, 10)
- Object - centre distance: calculate the Euclidean distance between cluster centres and the objects. For example, the distance of object C from the first center is:

$$\sqrt{(40-10)^2+(30-10)^2}=36.06$$

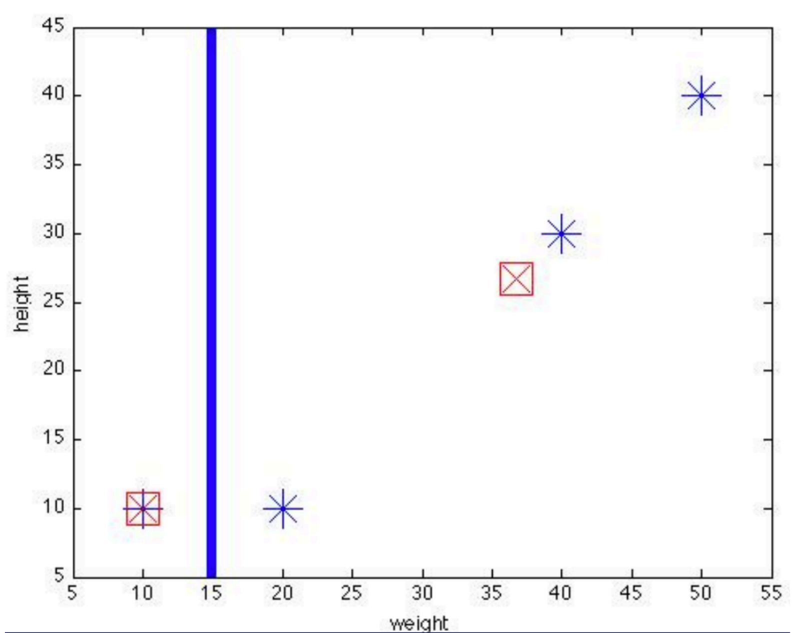
- We obtain the following distance matrix:

Centre 1	0	10	36.06	50
Centre 2	10	0	28.28	43.43

- Object clustering: We assign each object to one of the clusters based on the minimum distance from the centre:

Centre 1	1	0	0	0
Centre 2	0	1	1	1

- Determine centres: Based on the group membership, we compute the new centers
- $c_1 = (10, 10), c_2 = \left(\frac{20+40+50}{3}, \frac{10+30+40}{3}\right) = (36.7, 26.7)$



- Recompute the object-centre distances: We compute the distances of each data point from the new centres:

Centre 1	0	10	36.06	50
Centre 2	31.4	23.6	4.7	18.9

- Object clustering: We reassign the objects to the clusters based on the minimum distance from the centre:

Centre 1	1	1	0	0
Centre 2	0	0	1	1

- Determine the new centres:

$$c_1 = \left(\frac{10+20}{2}, \frac{10+10}{2}\right) = (15, 10)$$

$$c_2 = \left(\frac{40+50}{2}, \frac{30+40}{2}\right) = (45, 35)$$

- Recompute the object-centres distances:

Centre 1	5	5	32	46.1
Centre 2	43	35.4	7.1	7.1

- Object clustering:

Centre 1	1	1	0	0
Centre 2	0	0	1	1

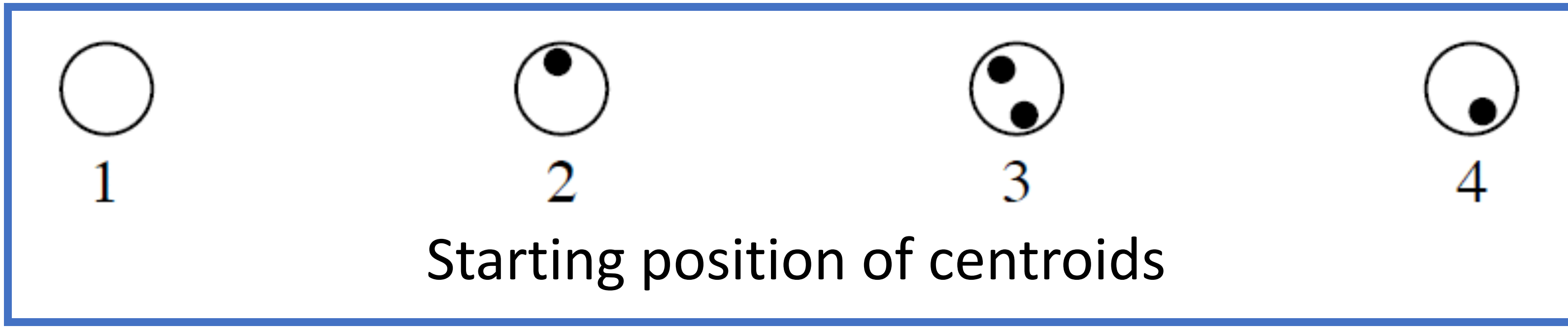
- The cluster membership did not change from one iteration to another and so the k-means computation terminates.

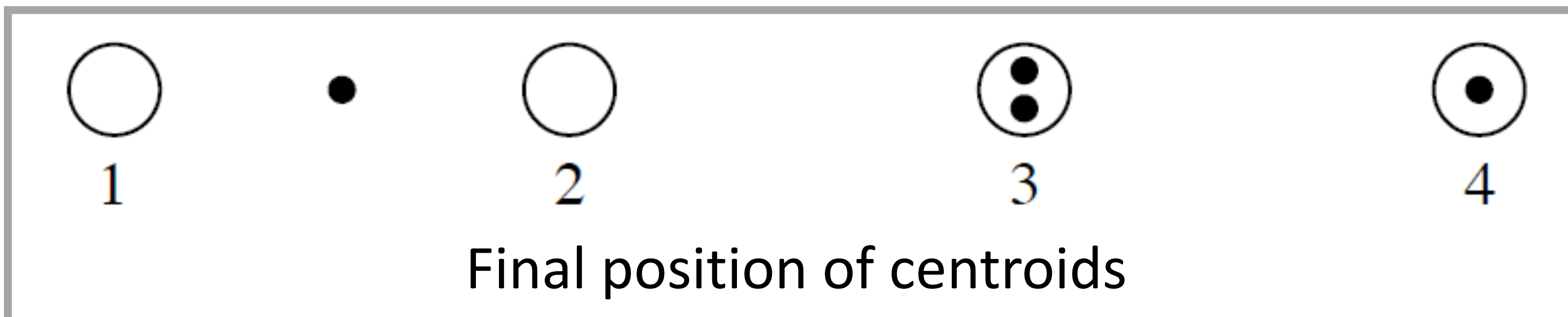
Discussion

Initialization

- Empty clusters
  - Pick data points to initialize clusters
- Bad local minima
  - Initialize many times and pick solution with smallest training loss
  - Pick good starting positions

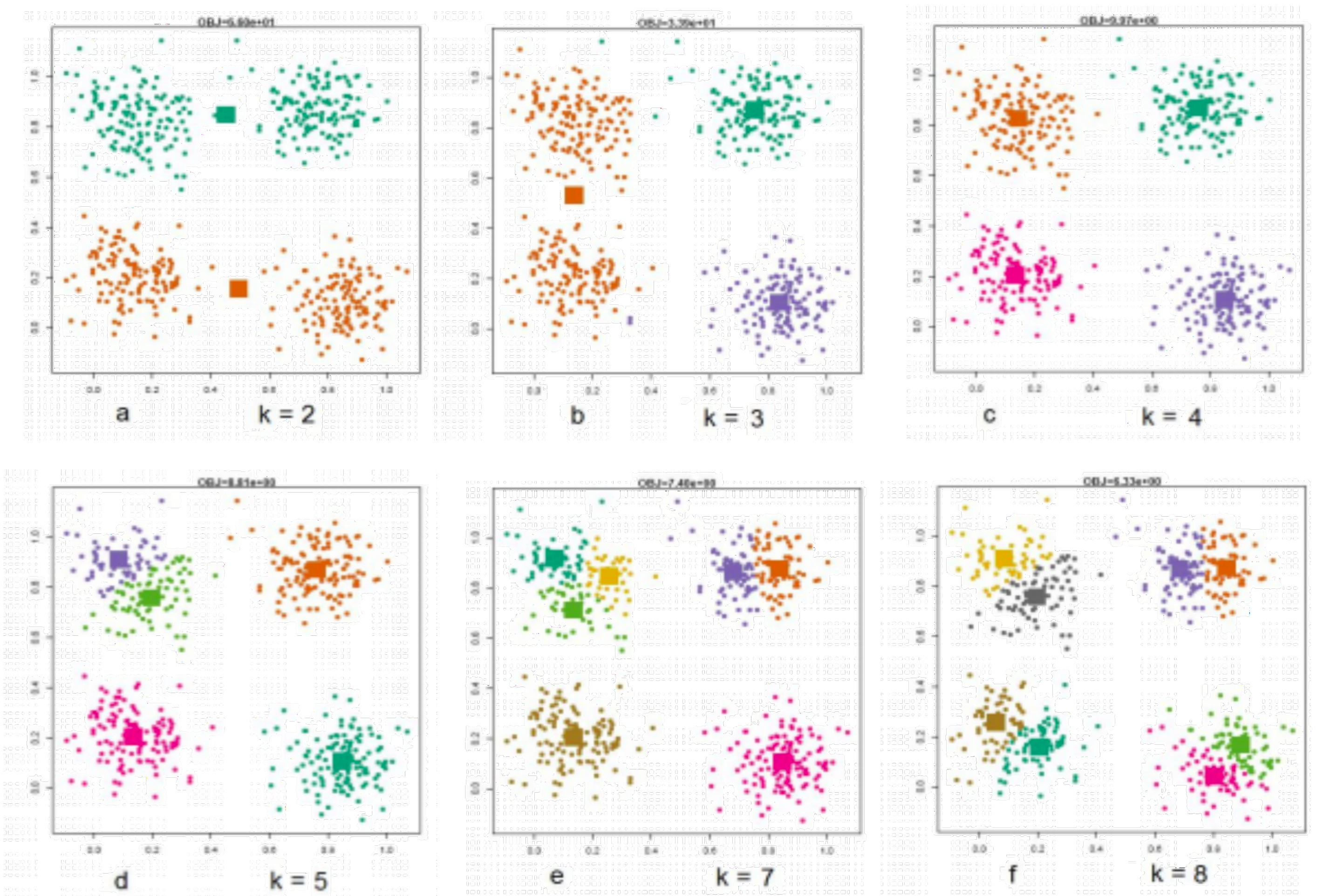
Initialization





Problem. How to choose good starting positions?
Solution. Place them far apart with high probability.

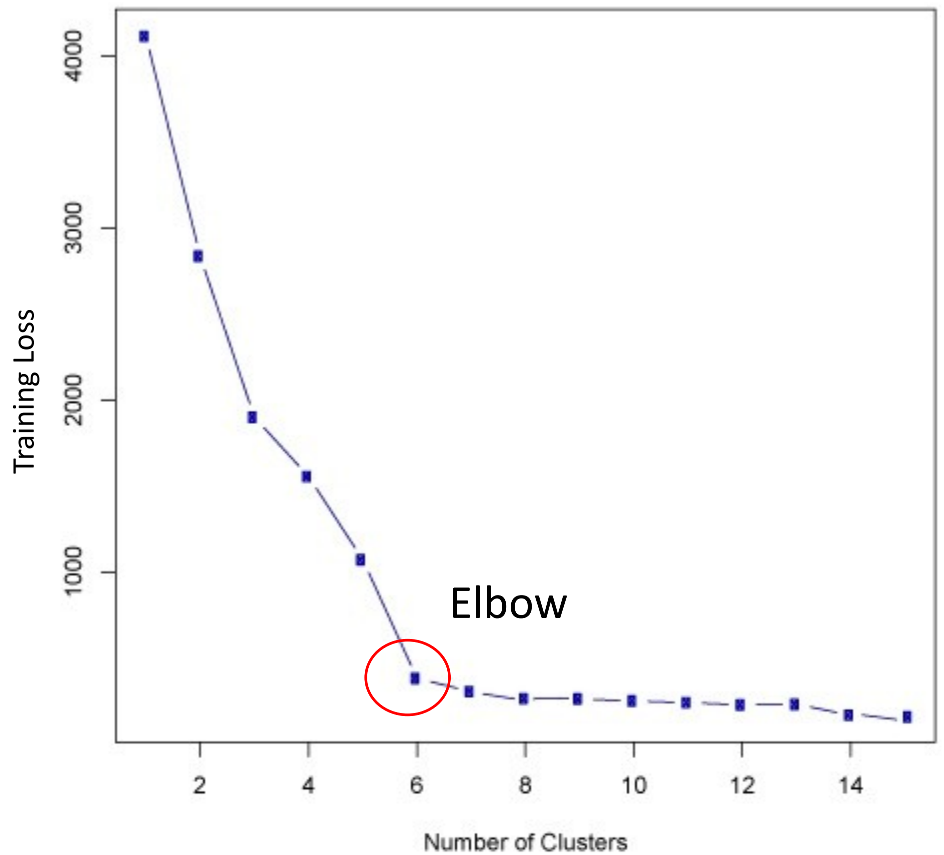
Number of Clusters



Number of Clusters

How do we choose k, the optimal number of clusters?

- Elbow method
  - Training Loss
  - Validation Loss



Check your understanding

- Clustering gives you an idea of how the data is distributed.
- You have 1000 data vectors that are very similar to each other (e.g., Eu distance less than 0.0001). You can only divide them into a few clusters.
- When you use kmeans, you usually obtain a global minimum of the loss function
- When you use kmeans, you can never achieve global minimum of the loss function.
- The number of clusters is a parameter that can be optimized by kmeans.
- In K-fold cross validation, K is a hyperparameter.
- Agglomerative clustering as we introduced in this lecture, has a fixed clustering result given distance measurement and the number of clusters desired, and tie breaker.