Clustering

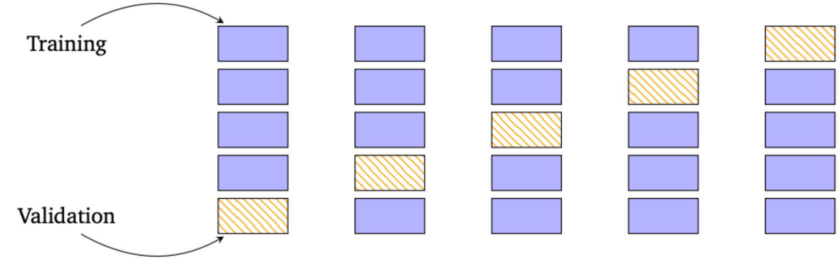
Rahul Shome

slides by Jo Ciucă

comp3670@anu.edu.au

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

Cross-validation Review



- Formally, we partition our dataset into two sets $\mathcal{D} = \mathcal{R} \cup \mathcal{V}$, such that they do not overlap, i.e., $\mathcal{R} \cap \mathcal{V} = \phi$
- We train on our model on $\mathcal{R}$ (training set)
- We evaluate our model on $\mathcal{V}$ (validation set)
- We have K partitions. In each partition k:
  - The training set $\mathcal{R}^{(k)}$ produces a predictor $f^{(k)}$
  - $f^{(k)}$ is applied to the validation set $\mathcal{V}^{(k)}$ to compute the empirical risk $\mathrm{R}\left(f^{(k)}, \mathcal{V}^{(k)}\right)$

- All the empirical risks are averaged to approximate the expected g

$$\mathbb{E}_V[R(f, \mathcal{V})] \approx \frac{1}{K} \sum_{k=1}^{K} \mathrm{R}\left(f^{(k)}, \mathcal{V}^{(k)}\right)$$

Cross-validation – key insights

- The training set is limited -- not producing the best $f^{(k)}$
- The validation set is limited – producing an inaccurate estimation of $\mathrm{R}\left(f^{(k)}, \mathcal{V}^{(k)}\right)$
- After averaging, the results are stable and indicative
- An extreme: leave-one-out cross-validation, where the validation set only contains one example.

- A potential drawback – computation cost
  - The training can be time-consuming
  - Difficult to evaluate many model hyperparameters.
- This problem can be solved by parallel computing, given enough computational resources

Outline

- Unsupervised learning
- Clustering
- K-means clustering
- Discussion

Supervised Learning

Input: Data **X** and label **y**

Goal: Learn how to map **X** to **y**

Examples: Classification, regression

Supervised Learning

Input: Data **X** and label **y**

Goal: Learn how to map **X** to **y**

Examples: Classification, regression

Unsupervised Learning

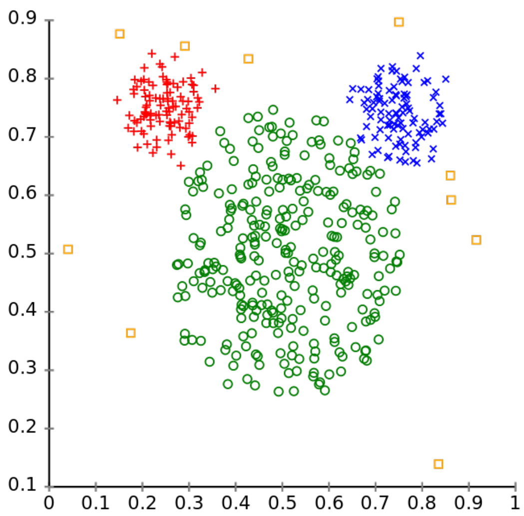
Input: Just data **X**, no labels

Goal: Learn *underlying structure* of the data

Examples: Clustering, Dimensionality reduction

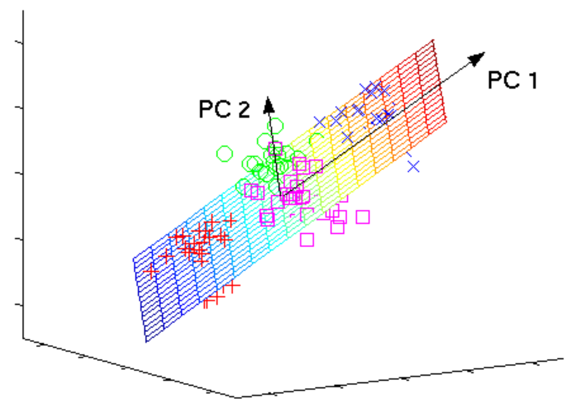
Unsupervised Learning

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



Clustering

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



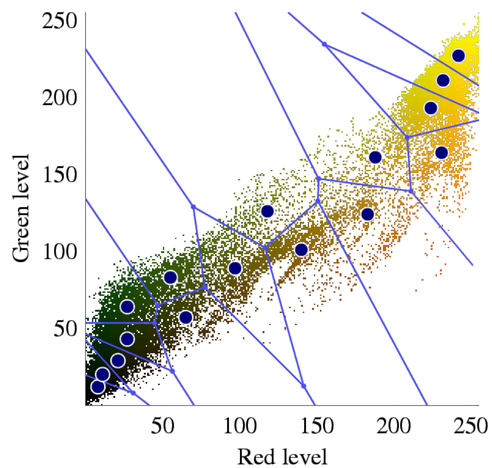
Subspace Learning

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

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).

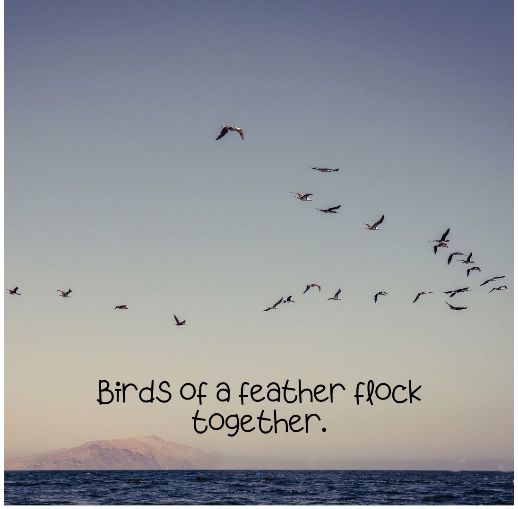
1. From unlabeled data, learn a good feature $T: \mathbb{R}^d \rightarrow \mathbb{R}^m$.
2. To labeled data, apply a transformation $T: \mathbb{R}^d \rightarrow \mathbb{R}^m$.

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

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

What is clustering?

What is Clustering?

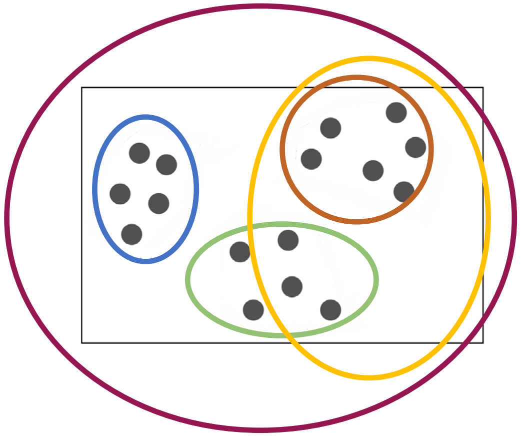


- Unsupervised learning - no information from teacher
- The process of partitioning a set of data into a set of meaningful (hopefully) sub-classes, called clusters

- Cluster:
  - collection of data points that are “similar” to one another and collectively should be treated as a group (*it is not the group we learned in linear algebra*)
  - as a collection, are sufficiently different from other groups

Basic Clustering Methodology

- Hierarchical Algorithms

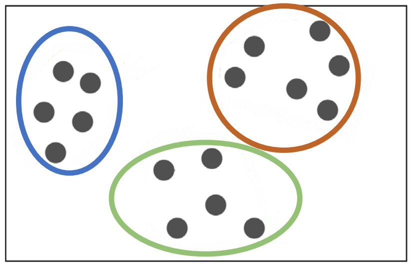


Cluster the clusters

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

- Flat Algorithms



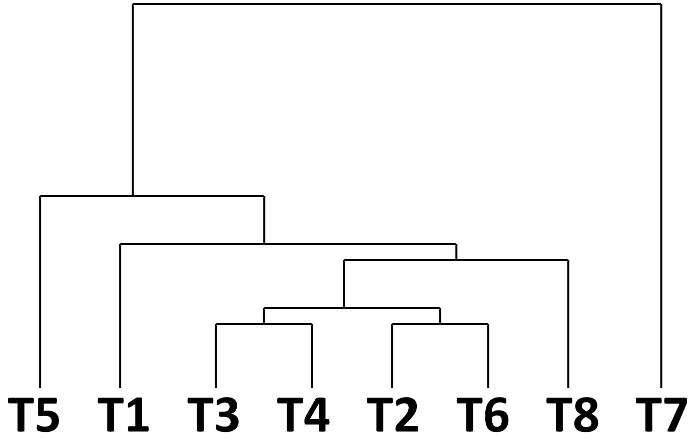
Usually start with a random (partial) partitioning of points into groups

Refine Iteratively

K-Means

Hierarchical Agglomerative

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

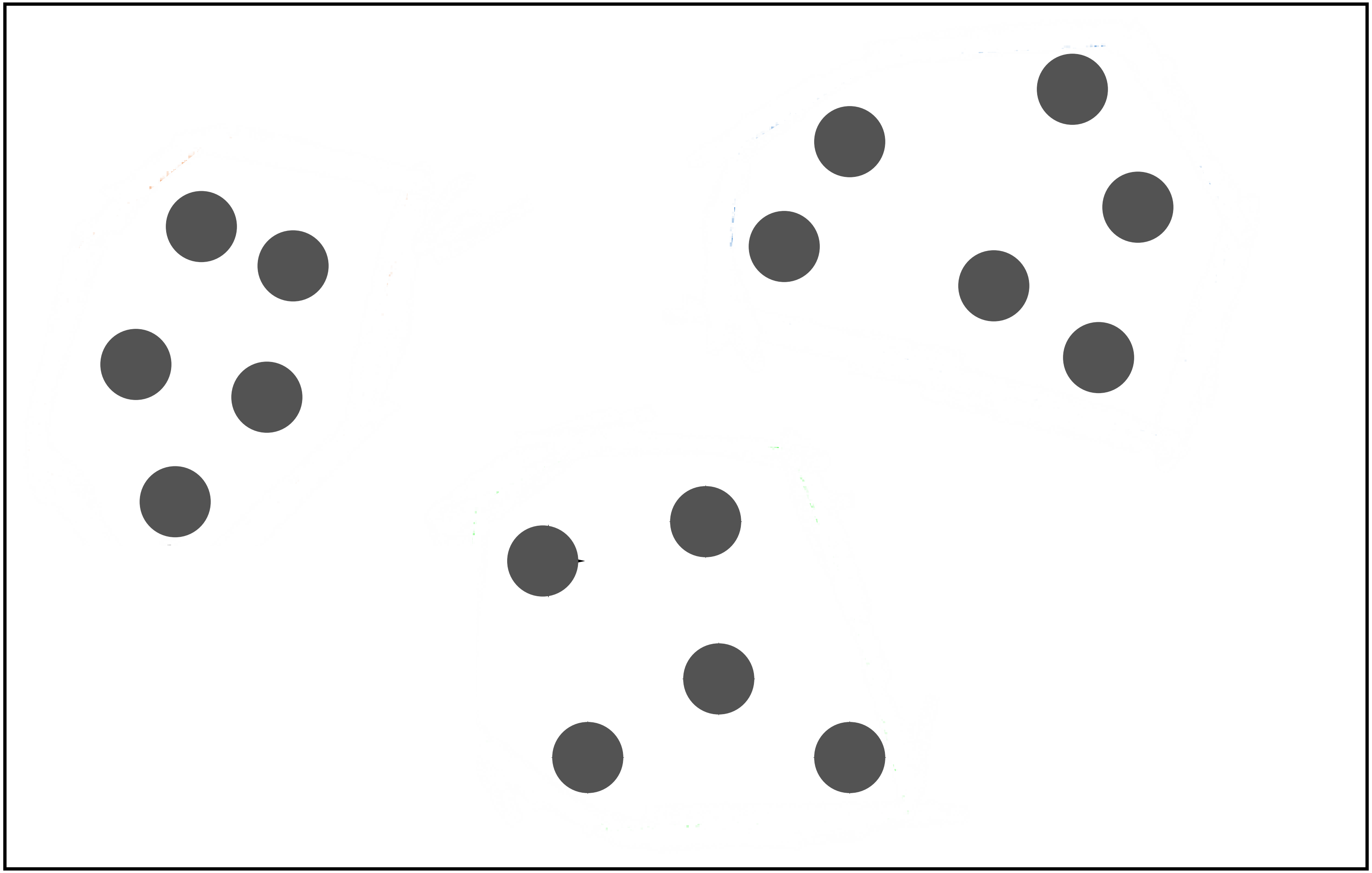


Hierarchical Agglomerative

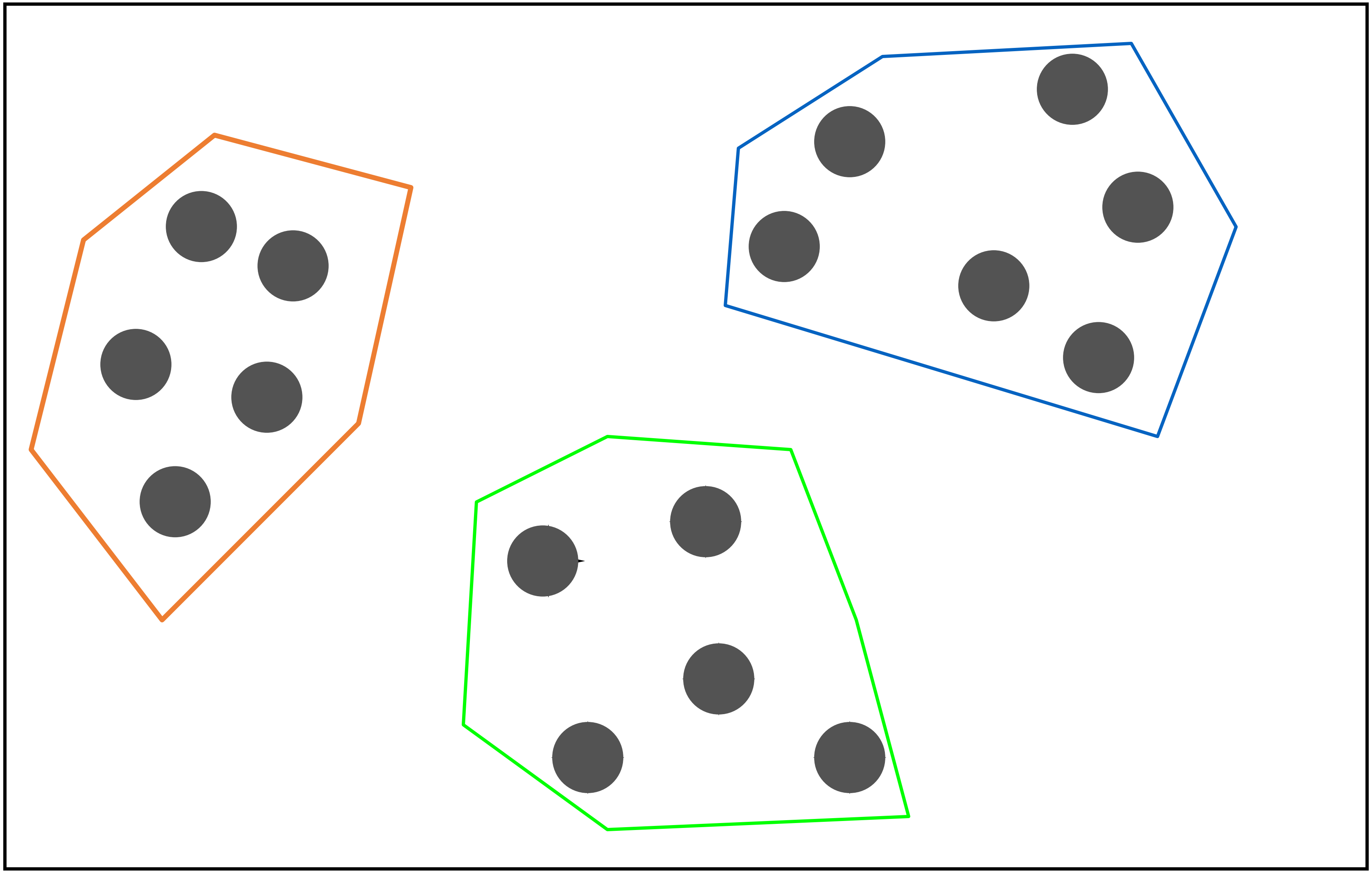
- Compute distance matrix
- Put each data point in its own cluster
- Find most similar pair of clusters
  - merge pairs of clusters (show merger in dendogram)
  - update distance matrix
  - repeat until all data points are in one cluster

K-Means

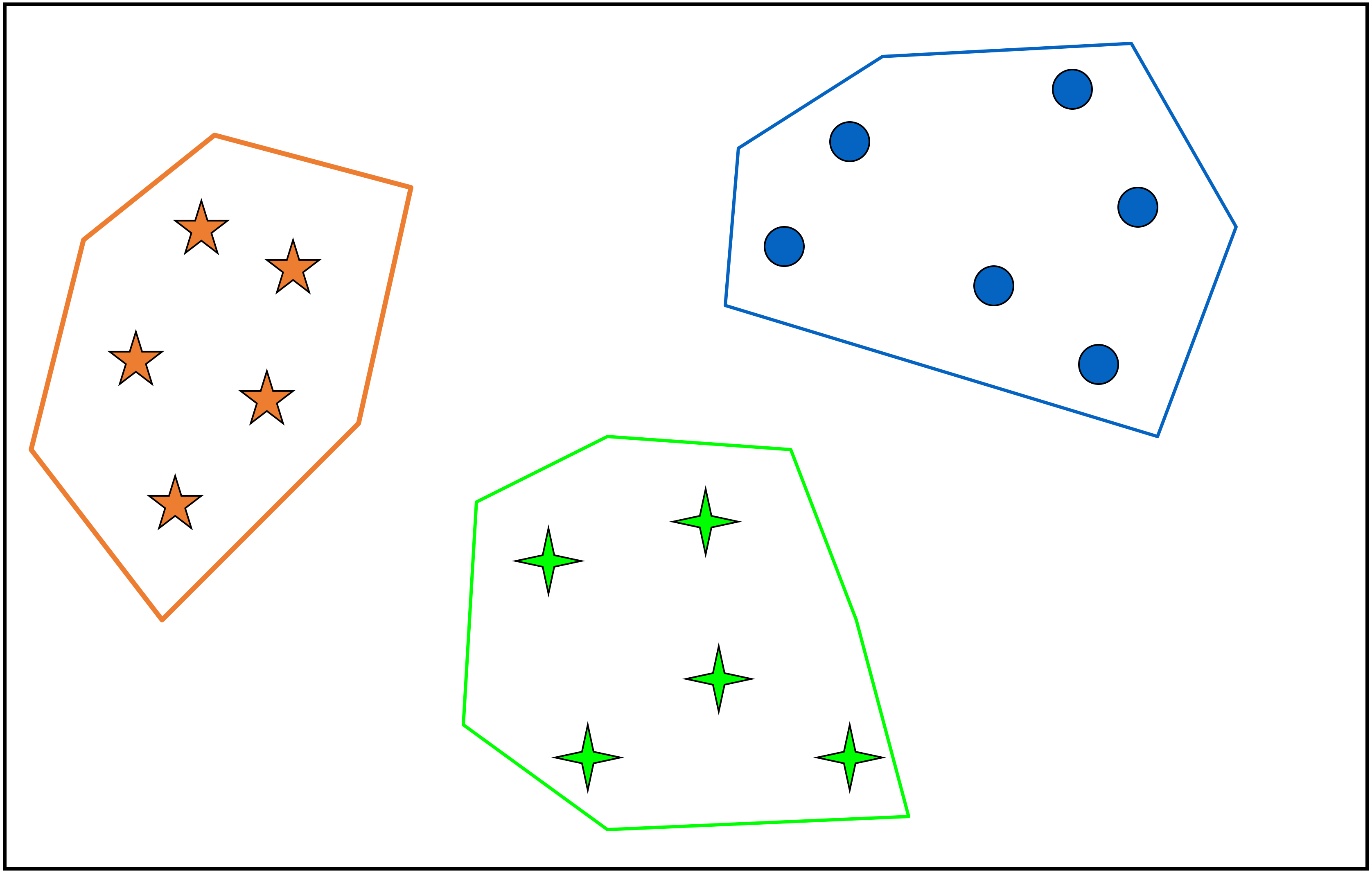
Clusters



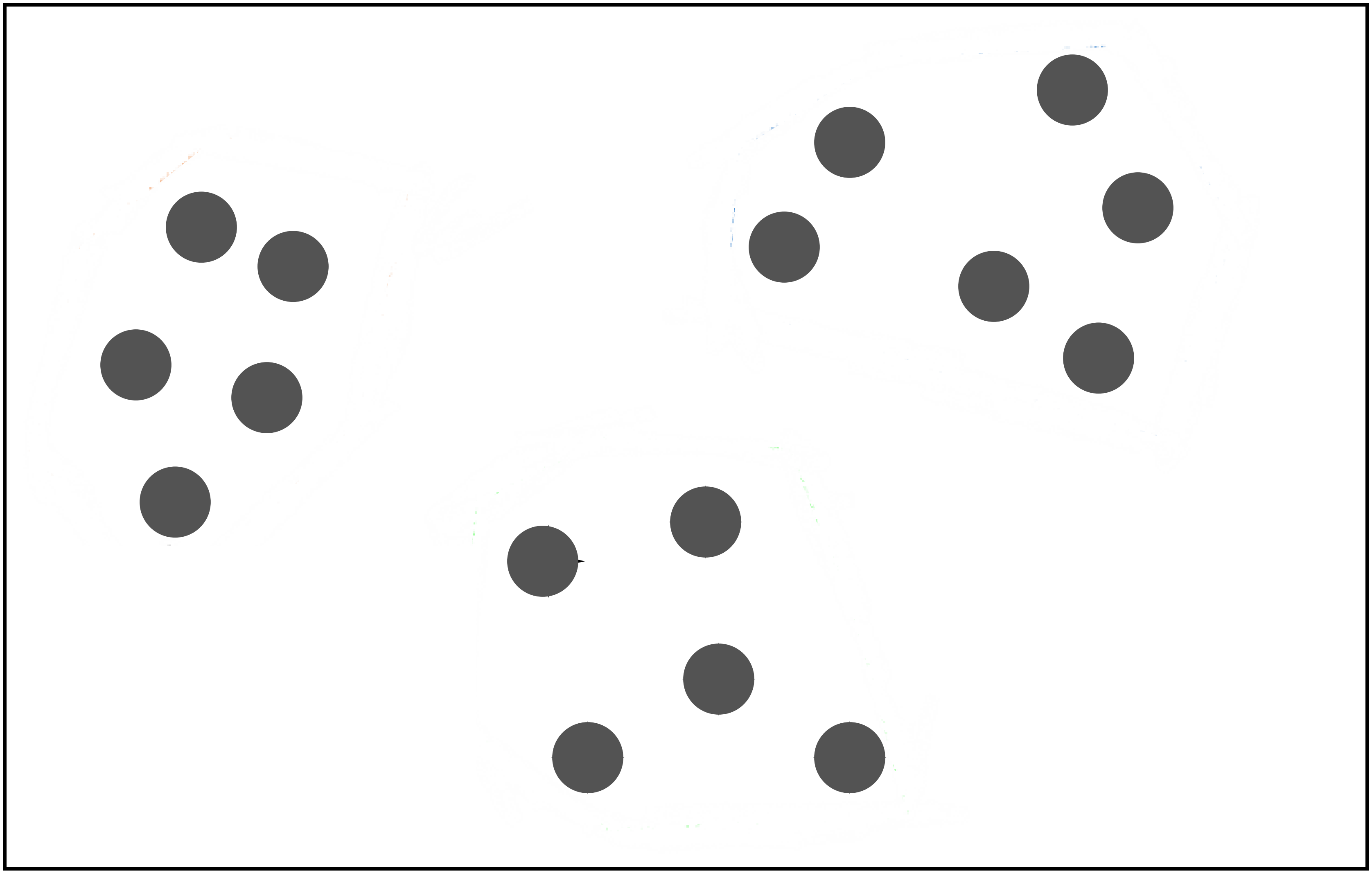
What your brain sees



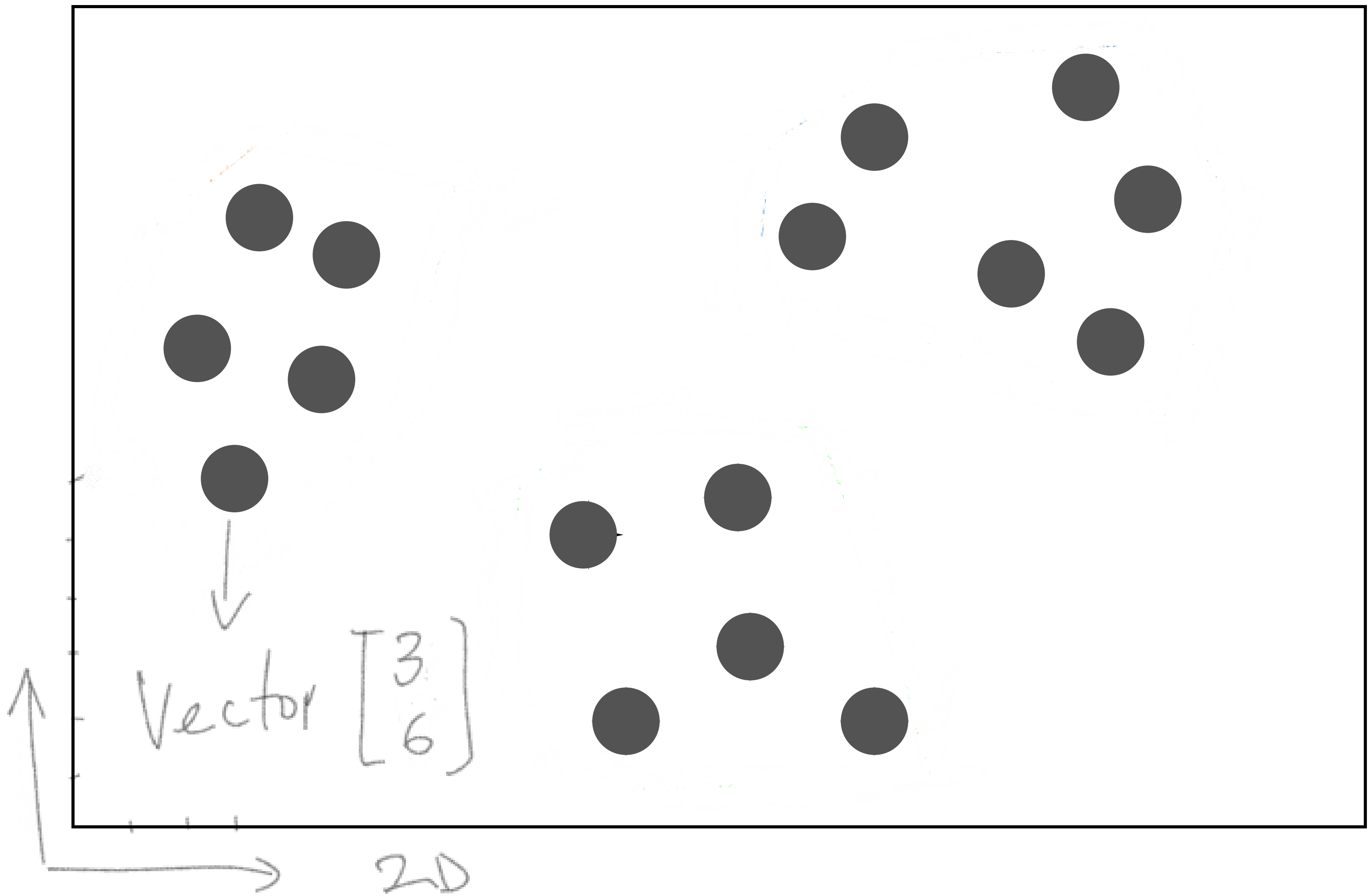
What your brain sees



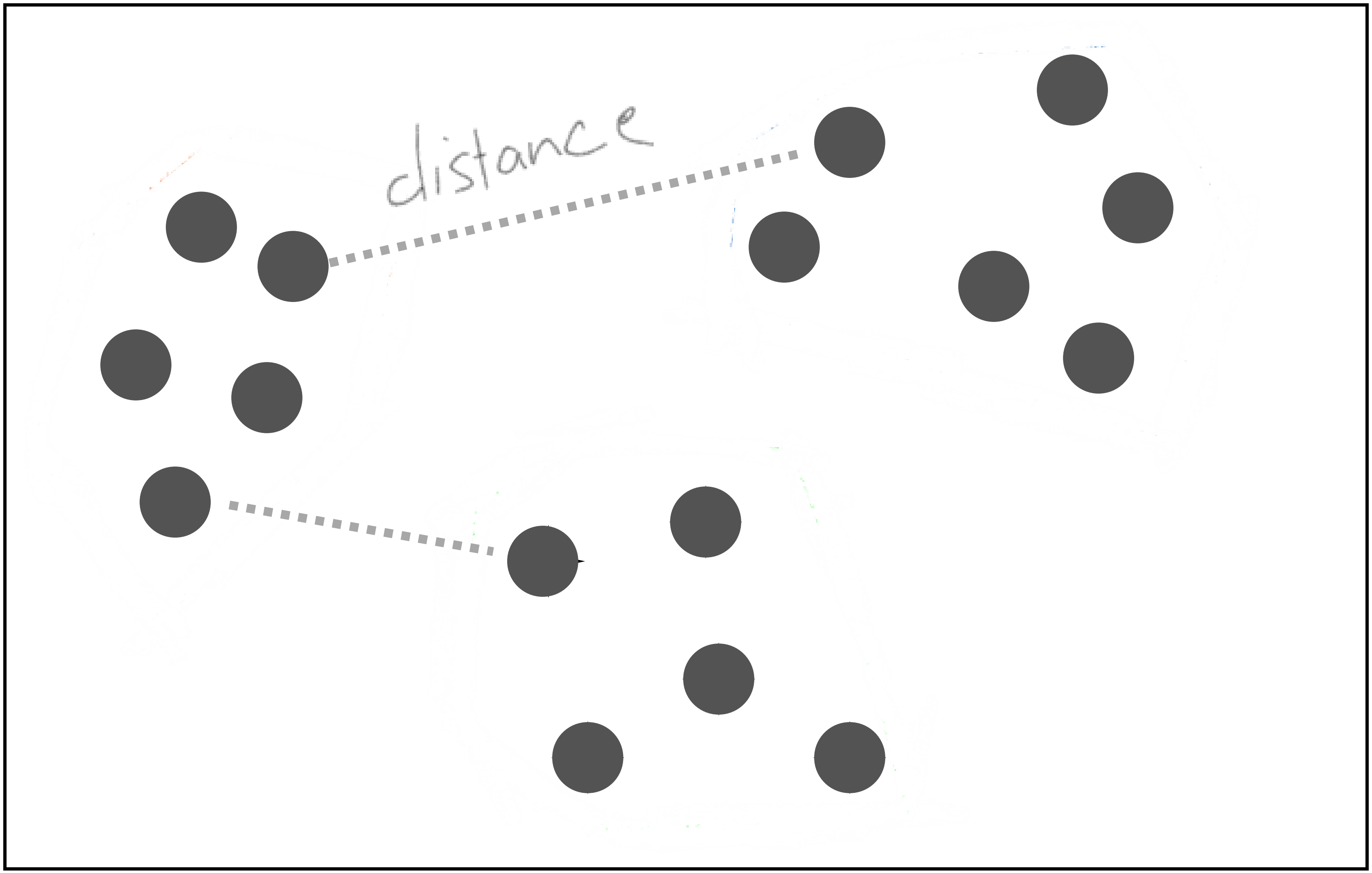
What the machine sees



What the machine sees



What the machine sees

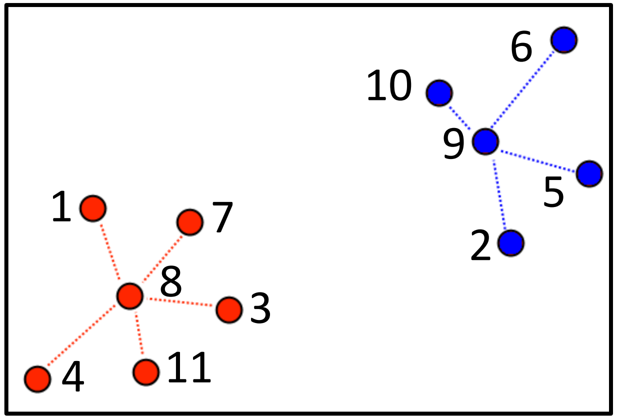


How to Specify a Cluster

- By listing all its elements

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

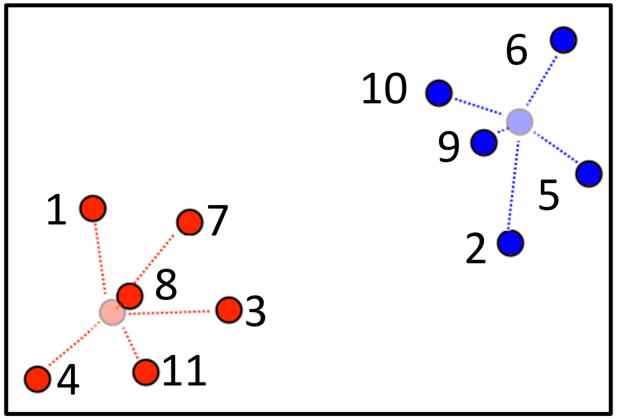
$$\mathcal{C}_2 = \{2,5,6,9,10\}$$



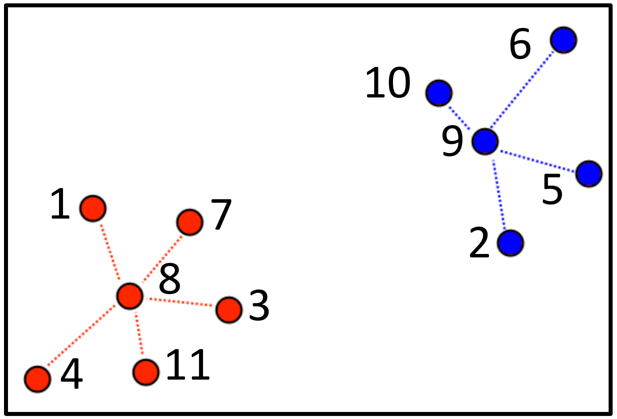
How to Specify a Cluster

- 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

Formalising the Problem

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

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 $j \neq k$.

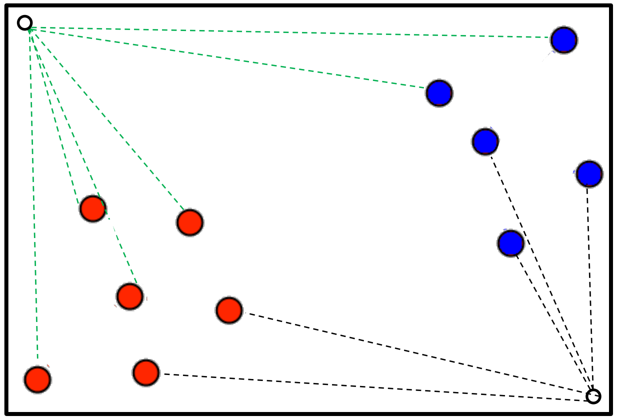
$\boldsymbol{R} = \{r_{nk}\} \in \{0,1\}^{n \times k}, n = 1, \dots, N, k = 1, \dots, K$

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

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

Training Loss

Sum of squared distances to closest representative.

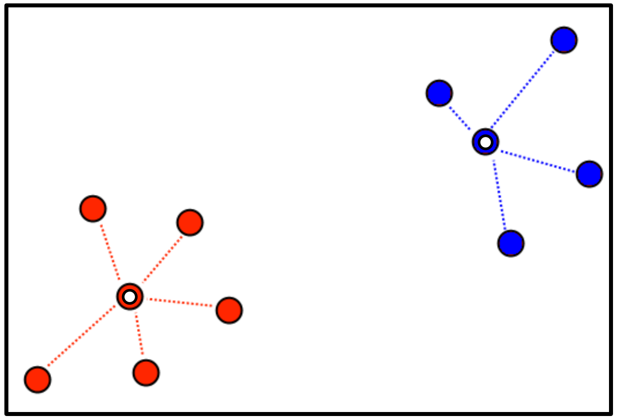


$\text{loss} \approx 11 \times (1)^2 = 11$

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

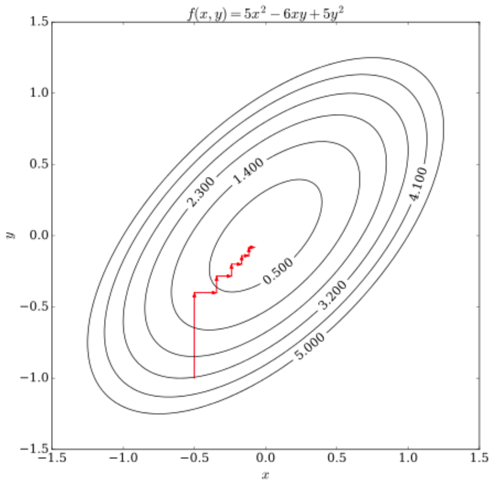
Optimization Algorithm

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

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.



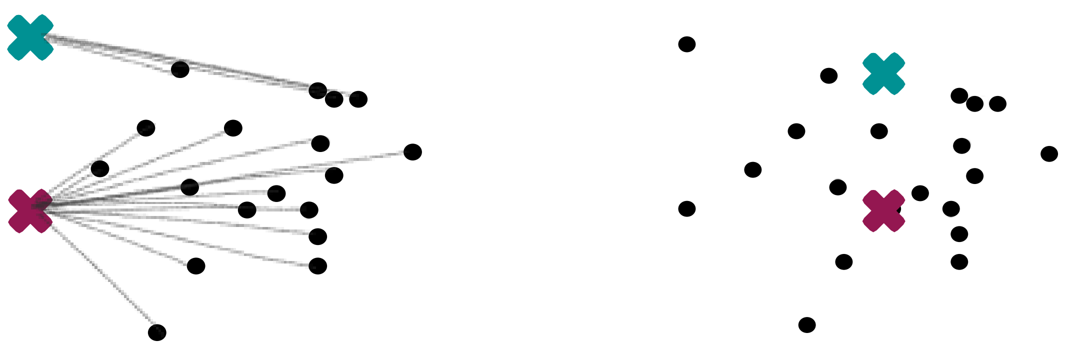
Coordinate descent is an optimization algorithm that successively minimizes along coordinate directions to find the minimum of a function.

Optimization Algorithm

Repeat until convergence:

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

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



Assignment Step **Update Step**

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 $n \in \{1, \dots, N\}$,

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

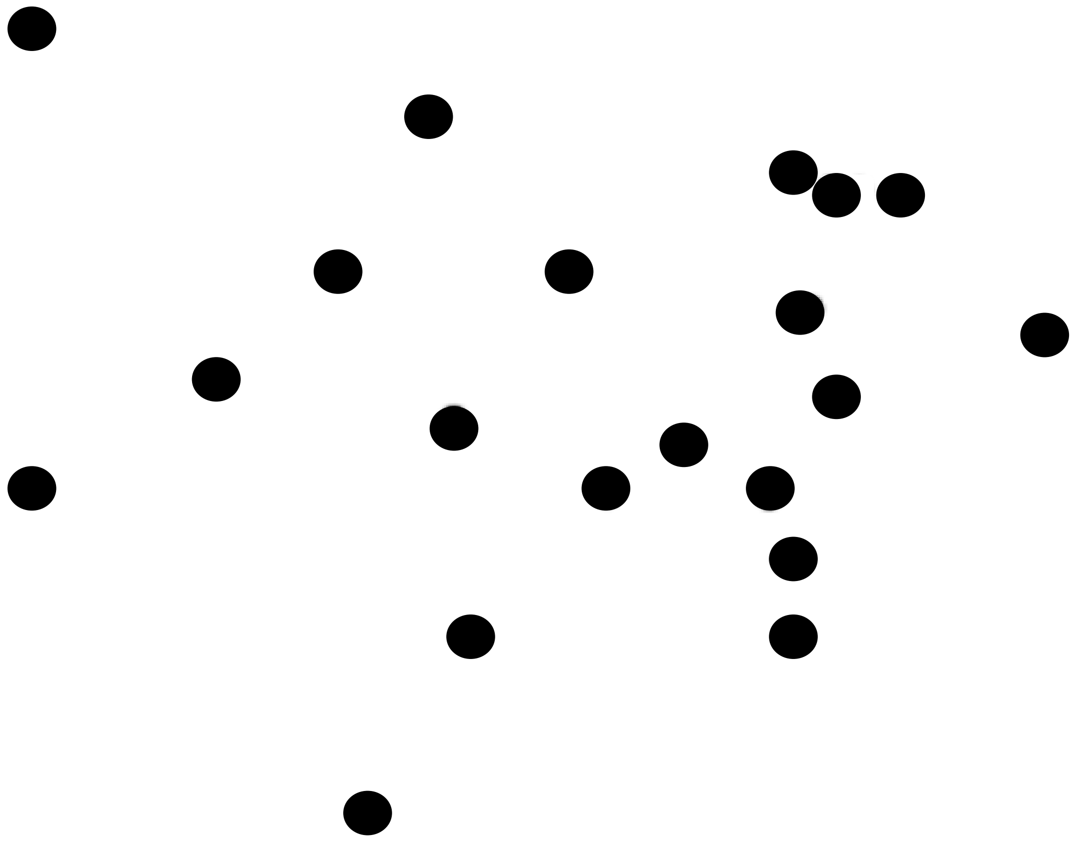
   We assign the n^{th} data point to the closest cluster centre.

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

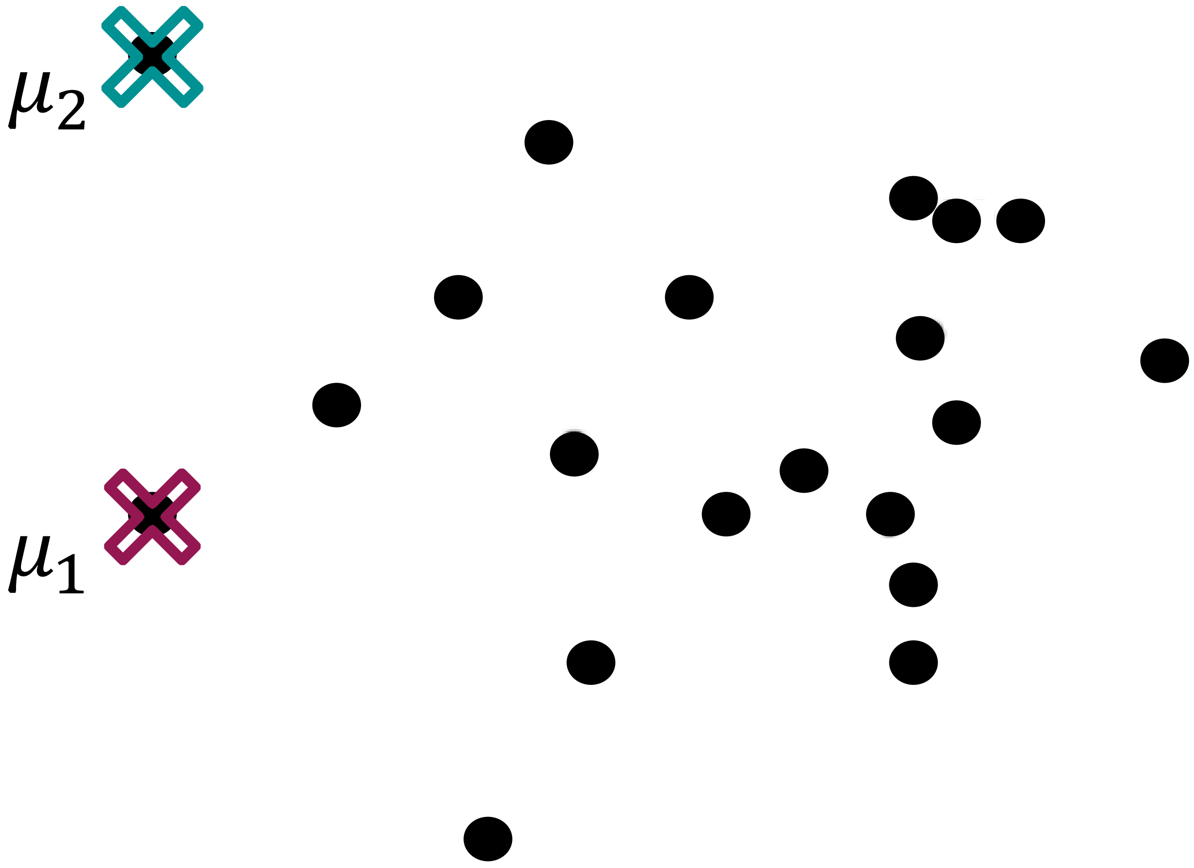
$$\boldsymbol{\mu}_{\boldsymbol{j}} = \frac{\sum_n r_{nj} \boldsymbol{x}_n}{\sum_n r_{nj}}$$

   We recompute cluster means.

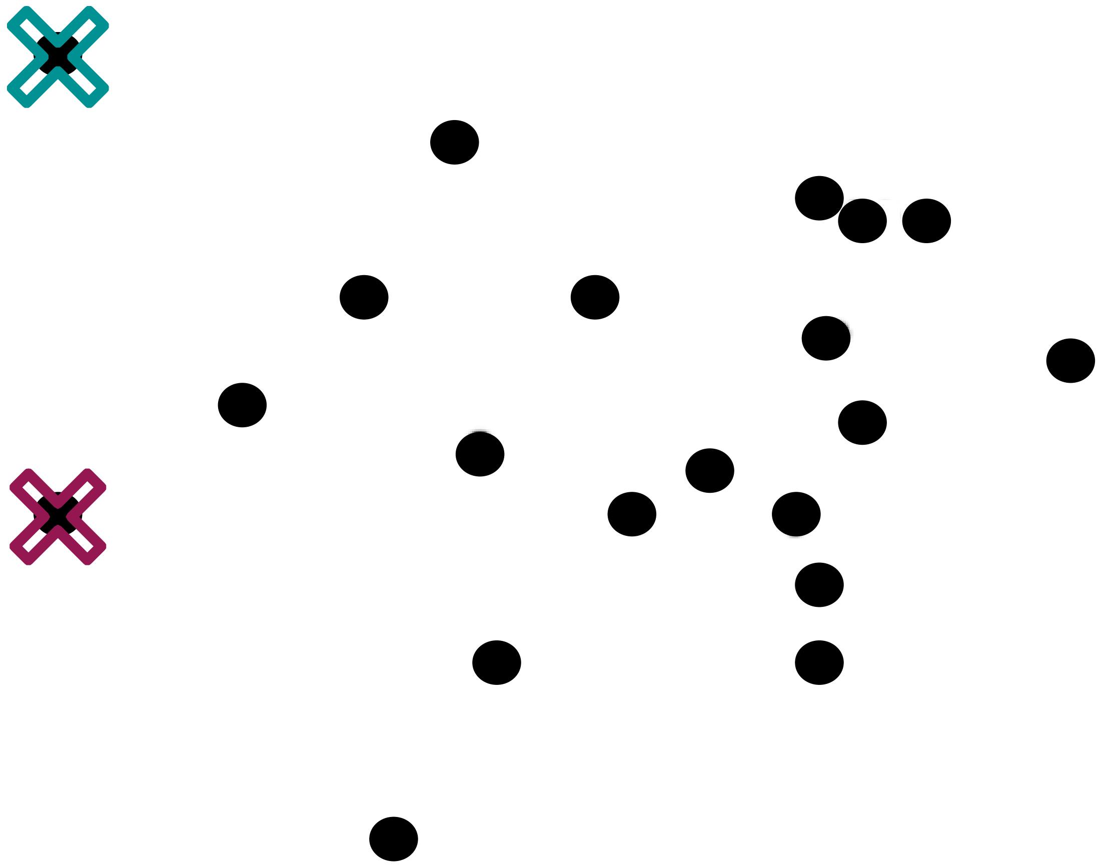
Goal: group data in 2 clusters



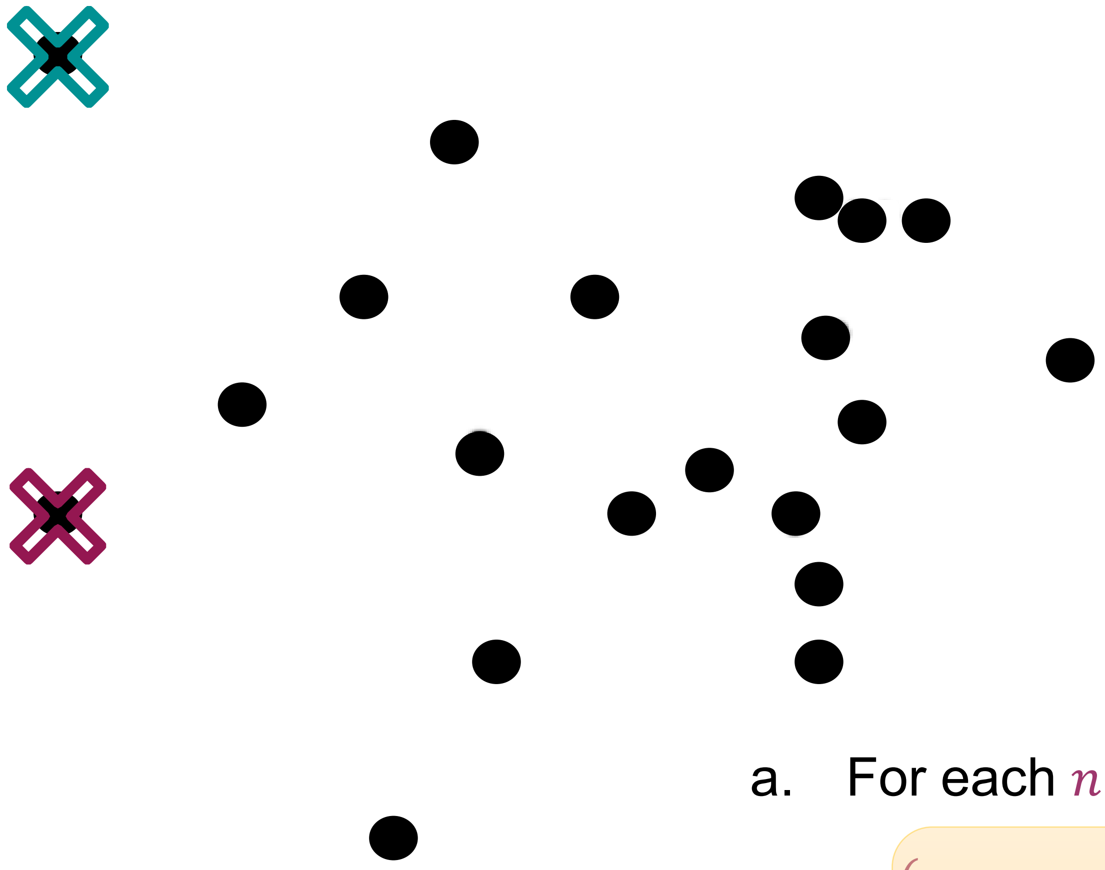
1. Initialize centers from the data.



“Call centre”: Each point calls to find $\left\| \boldsymbol{x}_n - \boldsymbol{\mu}_j \right\|^2$



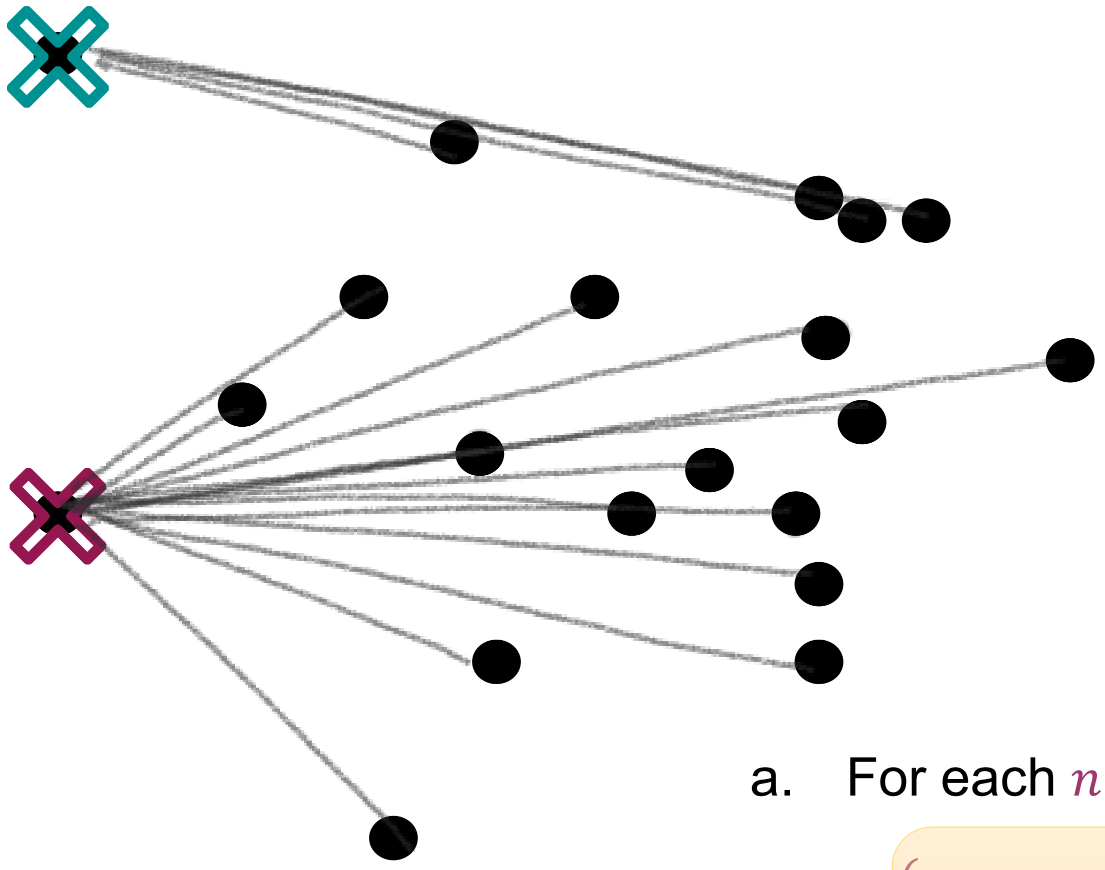
Step 1: Assign points to closest centroid.



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

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

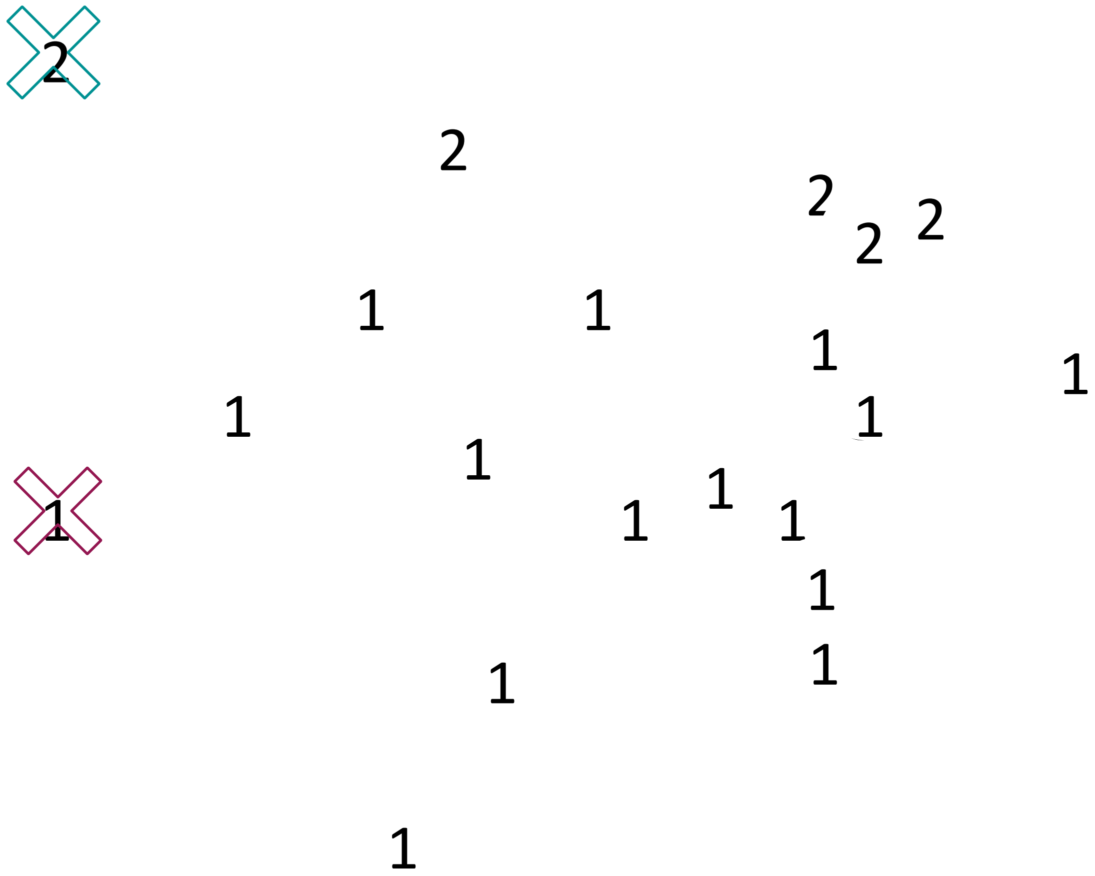
Step 1: Assign points to closest centroid.



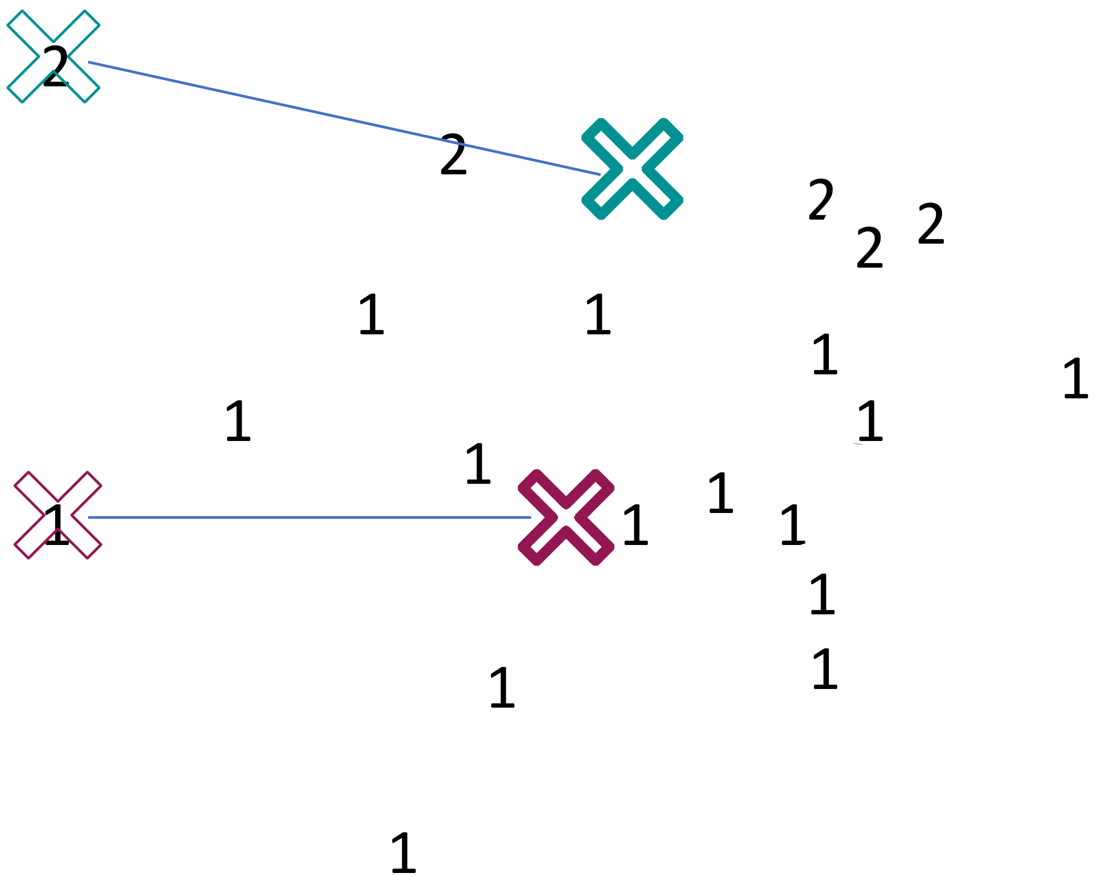
a. For each $n \in \{1, \ldots, N\}$,

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

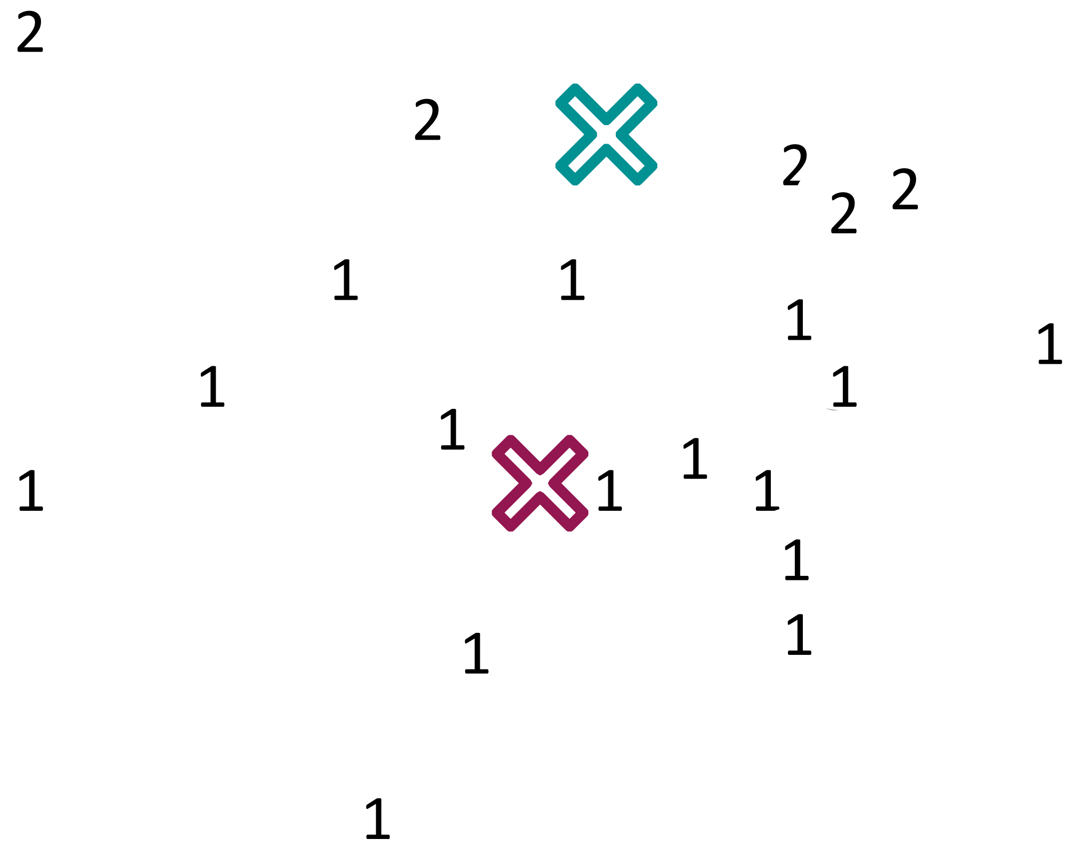
Step 1: Assign points to closest centroid.



Step 2: Compute cluster centres: $\boldsymbol{\mu}_j = \frac{\sum_n r_{nj} \boldsymbol{x}_n}{\sum_n r_{nj}}$



Repeat Step 1: Assign points to closest centroid.



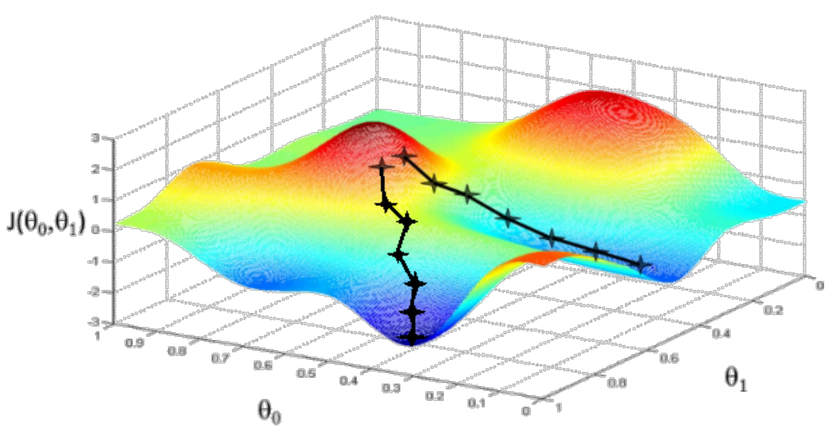
Compute updated cluster centre.

Repeat until convergence.

Break time

Convergence

- Training loss always decreases in each step.
- K-Means is guaranteed to converge.
- Convergence usually fast (less than 10-20 iterations).
- Converges to local minimum, not necessarily global minimum.

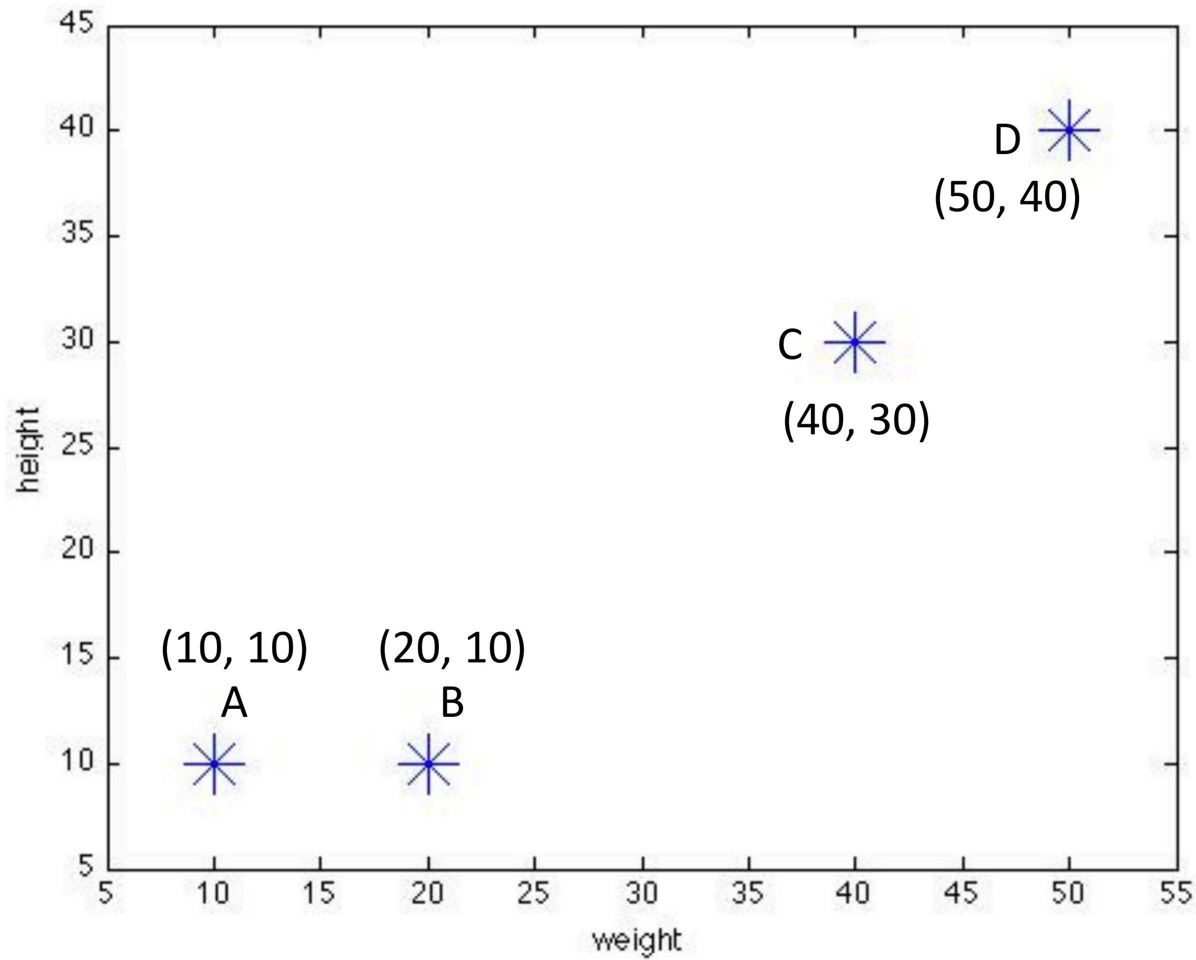


Repeat algorithm over many initial points and pick the configuration with the smallest training loss.

Let’s cluster.

Exercise – k-means clustering

- Suppose we have 4 boxes of different sizes and 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:

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

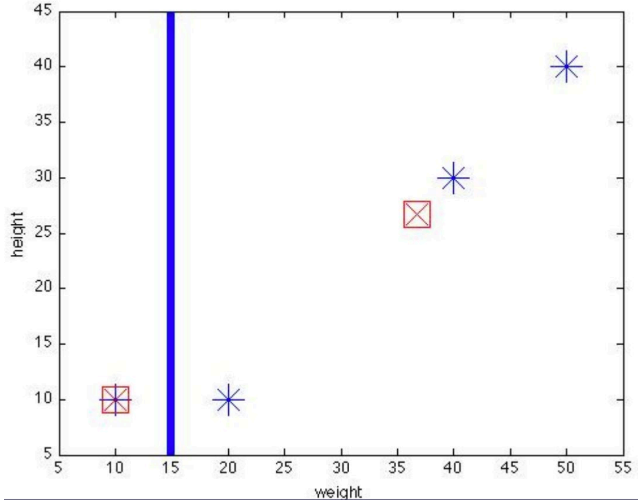
- We obtain the following distance matrix:

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.

Evaluating K-Means

- **Internal criteria**: the residual sum of squares
- But doesn’t evaluate the actual utility of the K-Means to an application

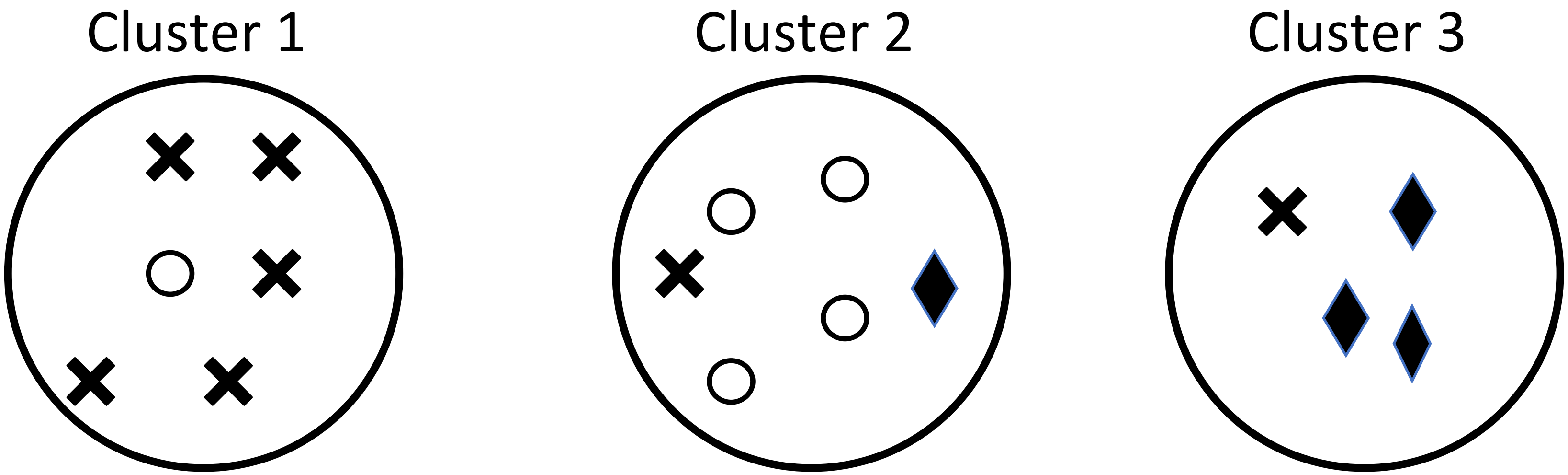
- **Extrinsic criteria**: evaluate with respect to a human-d

Extrinsic criterion: Purity

- If we have a set of truth class labels, we can use purity
- For each cluster k, find the class with the most members in the cluster
- Sum them and divide by the total number of points

$$Purity = \frac{1}{N} \sum_{k=1}^{K} \max_{j} |C_k \cap c_j|$$

Example



$$5 = \max_j |\mathcal{C}_1 \cap c_j|$$ (Cluster 1, class x)

$$4 = \max_j |\mathcal{C}_2 \cap c_j|$$ (Cluster 2, class o)

$$3 = \max_j |\mathcal{C}_3 \cap c_j|$$ (Cluster 3, class ◆)

$$Purity = \frac{1}{17}(5 + 4 + 3) \approx 0.71$$

Discussion

Initialization

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

Initialization

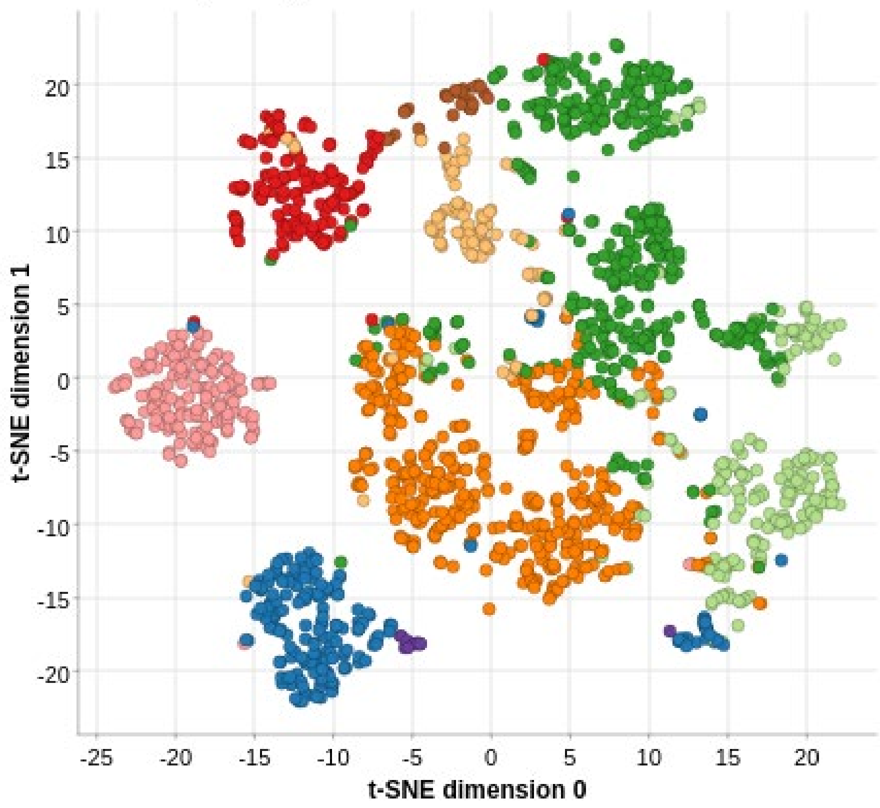
Starting position of centroids

Final position of centroids

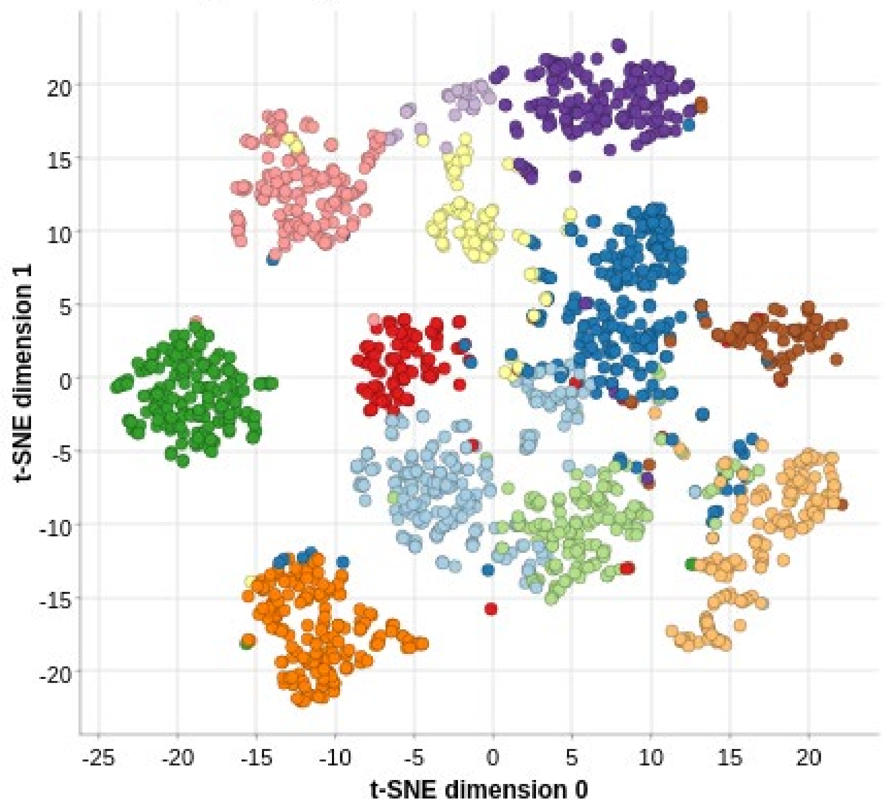
Problem. How to choose good starting positions?
Solution. Place them far apart with high probability.

Number of Clusters

k-means (k=9)



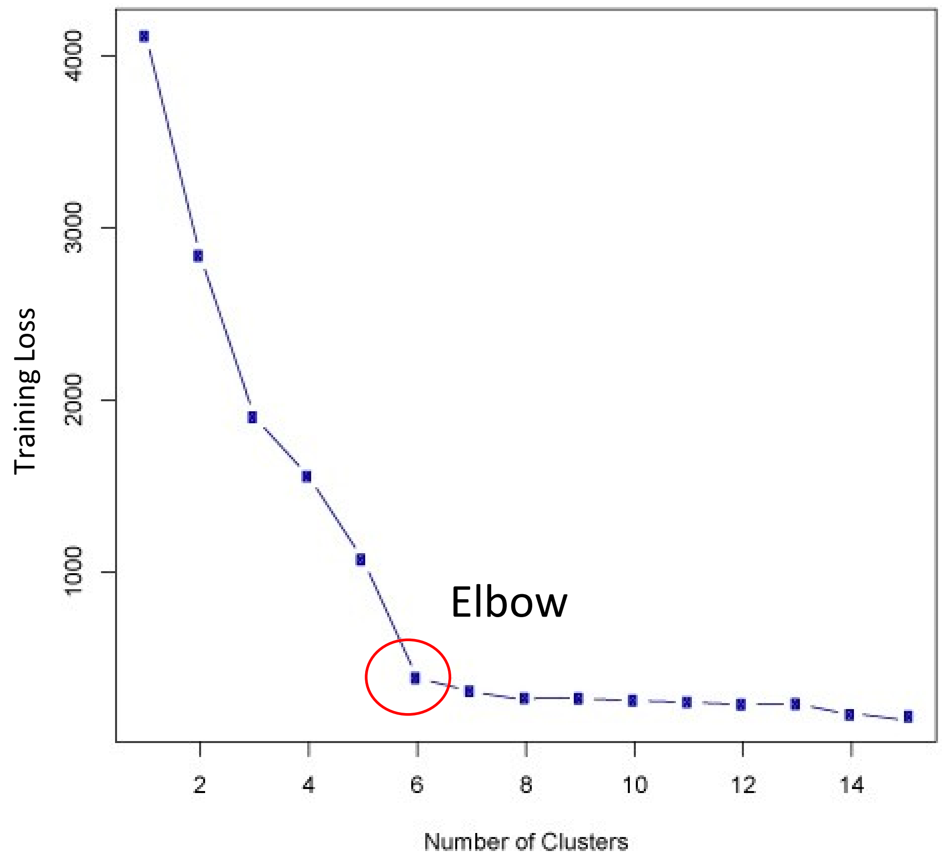
k-means (k=12)



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). We should divide them into a few clusters.
- When you use K-Means, you usually obtain a global minimum of the loss function
- When you use K-Means, you can never achieve global minimum of the loss function.
- The number of clusters is a parameter that can be optimized by K-Means.
- In K-fold cross validation, K is a hyperparameter.
- Clustering is just a different version of classification.

Check your understanding

- Clustering gives you an idea of how the data is distributed. Y
- You have 1000 data vectors that are very similar to each other (e.g., Eu distance less than 0.0001). We should divide them into a few clusters. N
- When you use K-Means, you usually obtain a global minimum of the loss function N
- When you use K-Means, you can never achieve global minimum of the loss function. N
- The number of clusters is a parameter that can be optimized by kmeans. N
- In K-fold cross validation, K is a hyperparameter. N
- Clustering is just a different version of classification. N