ML
- unsupervised $\rightsquigarrow$ we only focus on $\underline{X}$ $\rightsquigarrow$ clustering
- supervised $\rightsquigarrow$ we have both X & Y $\rightsquigarrow$ Regression & Classification

$$\underline{X_{n\times d}} \qquad \underline{Y_{n\times 1}}$$

X $\downarrow$ datapoints

Y: labels or target values or outputs

Clustering Analysis and K-Means

Mahdi Roozbahani
Georgia Tech

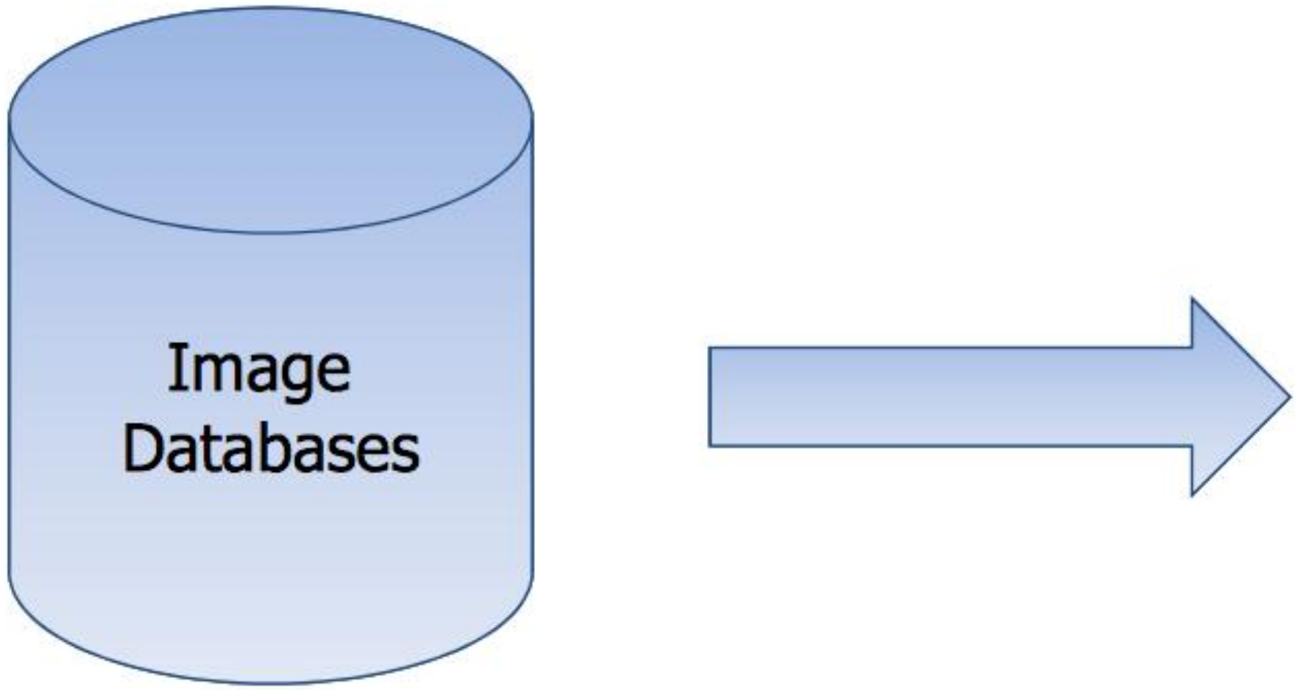
Some of the slides are inspired based on slides from Chao Zhang, and Le Song.

60+ hours on 16 GPU nvidia CUDA cluster.

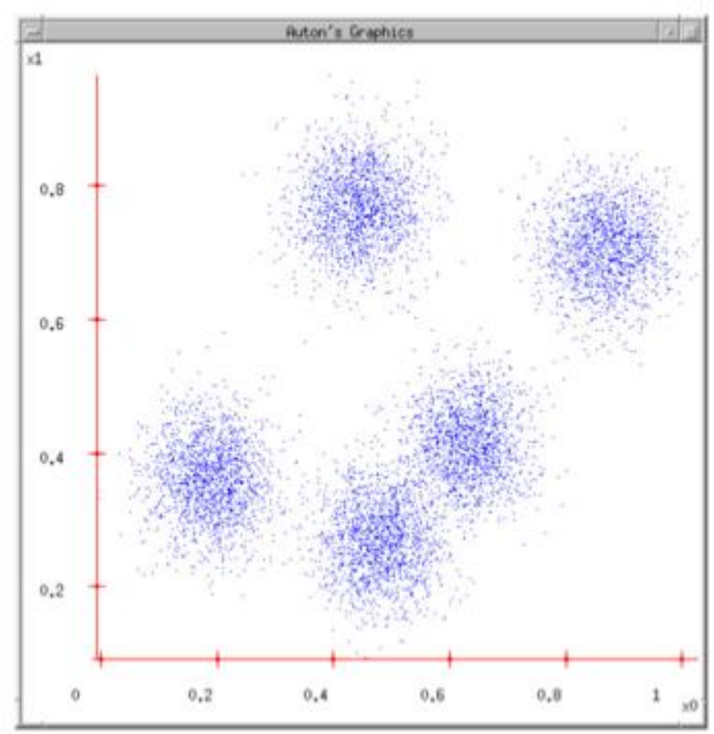
Outline

- Clustering
- Distance Function
- K-Means Algorithm
- Analysis of K-Means

Clustering Images



Goal of clustering:
Divide object into groups, and objects within a group are more similar than those outside the group



Clustering Other Objects

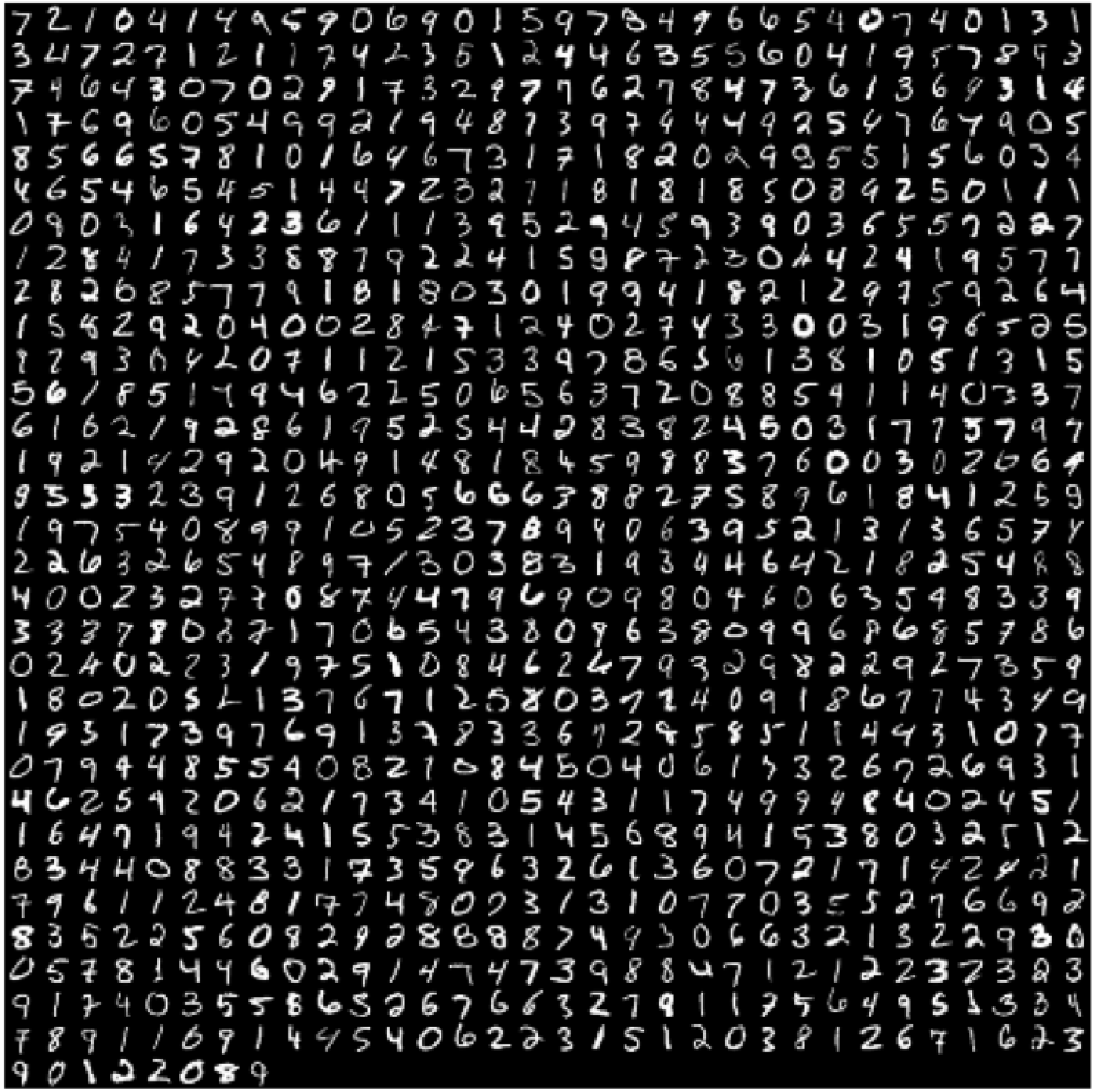
Australia | St. Helena & Dependencies | Anguilla | South Georgia & South Sandwich Islands | U.K. | Serbia & Montenegro | France | Niger | India | Ireland | Brazil

Piotr Belarusian
Pyotr Azerbaijani
Petros Greek
Pietro Italian
Pedro Portuguese
Pierre French
Piero Italian
Peter Dutch
Peder Danish
Peka Couldn't find it – Finish?
Peadar Irish

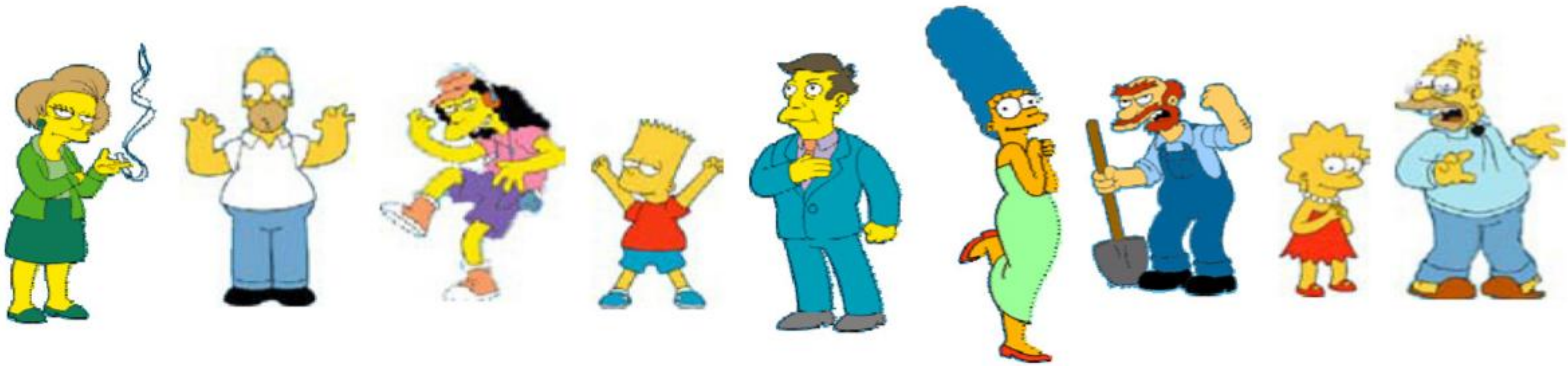
Linguistic Similarity

Clustering Hand Digits

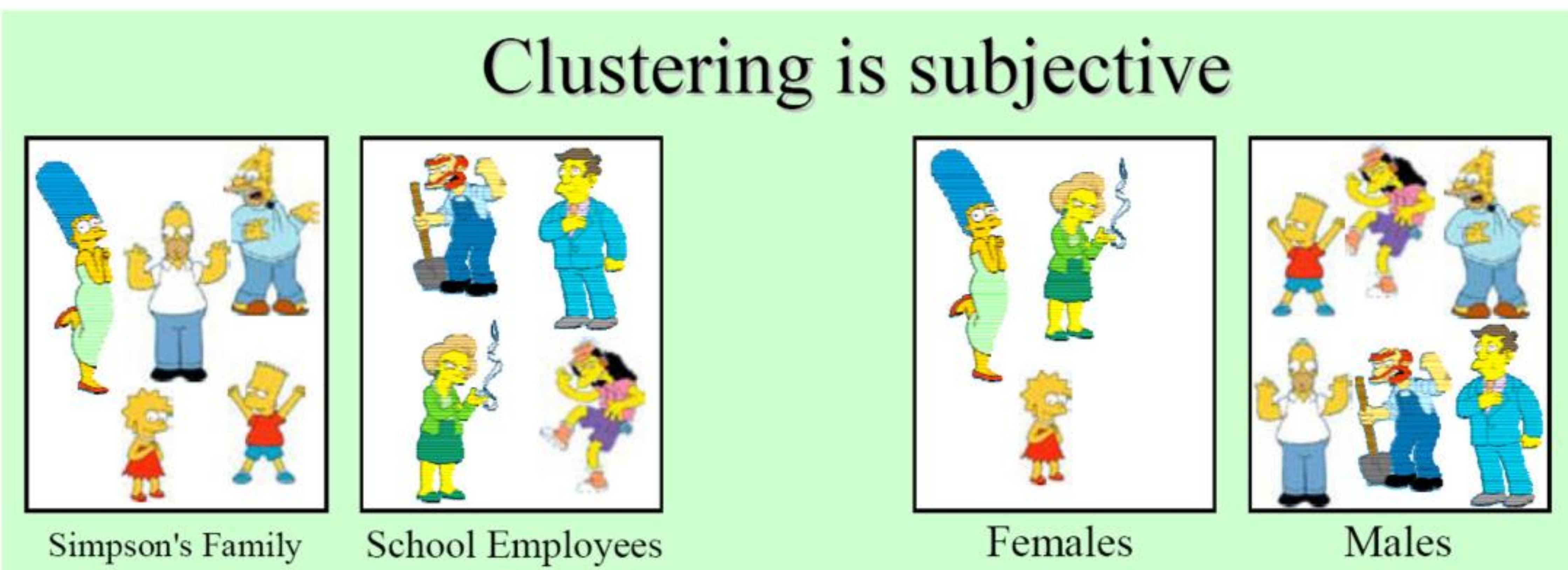
0 1 2 3 4 5 6 7 8 9



Clustering is Subjective



What is consider similar/dissimilar?



Are they similar or not?

So What is Clustering in General?

- You pick your similarity/dissimilarity function
- The algorithm figures out the grouping of objects based on the chosen similarity/dissimilarity function
  - Points within a cluster is similar
  - Points across clusters are not so similar
- Issues for clustering
  - How to represent objects? (Vector space? Normalization?)
  - What is a similarity/dissimilarity function for your data?
  - What are the algorithm steps?

Outline

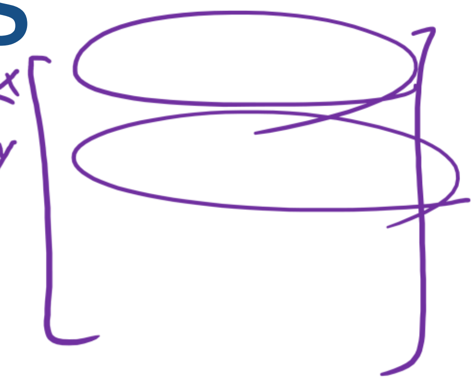
- Clustering
- Distance Function
- K-Means Algorithm
- Analysis of K-Means

Properties of Similarity Function

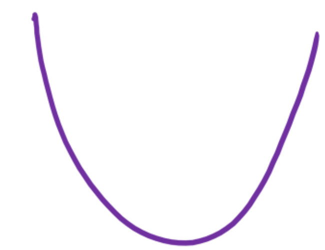
- Desired properties of dissimilarity function
  - Symmetry: $d(x, y) = d(y, x)$
    - *Otherwise you could claim "Alex looks like Bob, but Bob looks nothing like Alex"*
  - Positive separability: $d(x, y) = 0$, if and only if $x = y$
    - *Otherwise there are objects that are different, but you cannot tell apart*
  - Triangular inequality: $d(x, y) \leq d(x, z) + d(z, y)$
    - *Otherwise you could claim "Alex is very like Bob, and Alex is very like Carl, but Bob is very unlike Carl"*

Distance Functions for Vectors

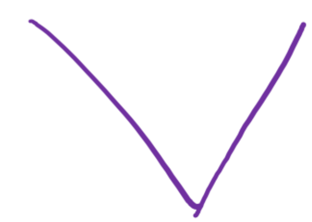
X = y



- Suppose two data points, both in R^d
  - $x = (x_1, x_2, \ldots, x_d)$
  - $y = (y_1, y_2, \ldots, y_d)$

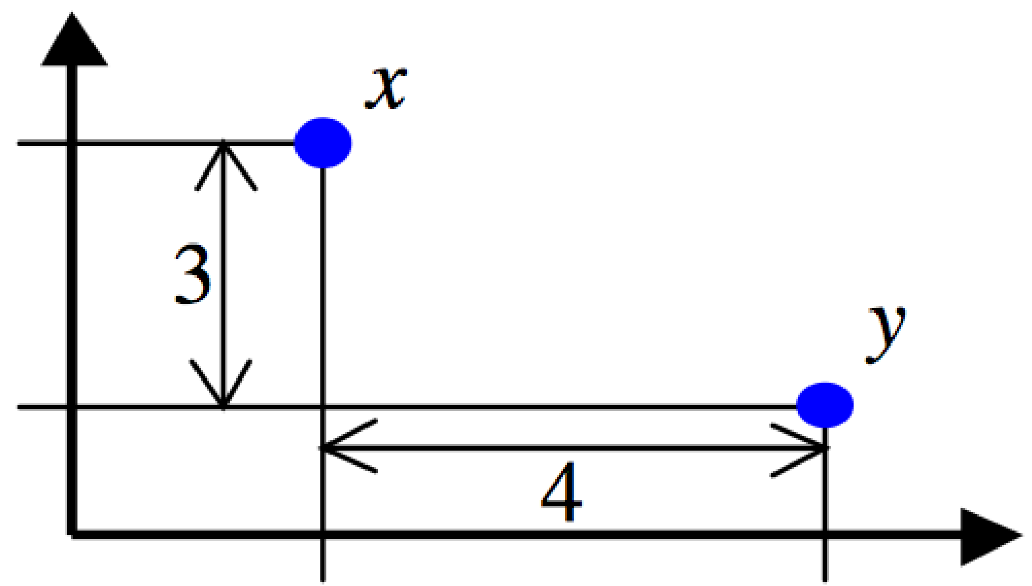


- Euclidean distance: $d(x, y) = \sqrt{\sum_{i=1}^{d} (x_i - y_i)^2}$
- Minkowski distance: $d(x, y) = \sqrt[p]{\sum_{i=1}^{d} (x_i - y_i)^p}$
  - Euclidean distance: $p = 2$



  - Manhattan distance: $p = 1, d(x, y) = \sum_{i=1}^{d} |x_i - y_i|$
  - "inf"-distance: $p = \infty, d(x, y) = \max_{i=1}^{d} |x_i - y_i|$

Example

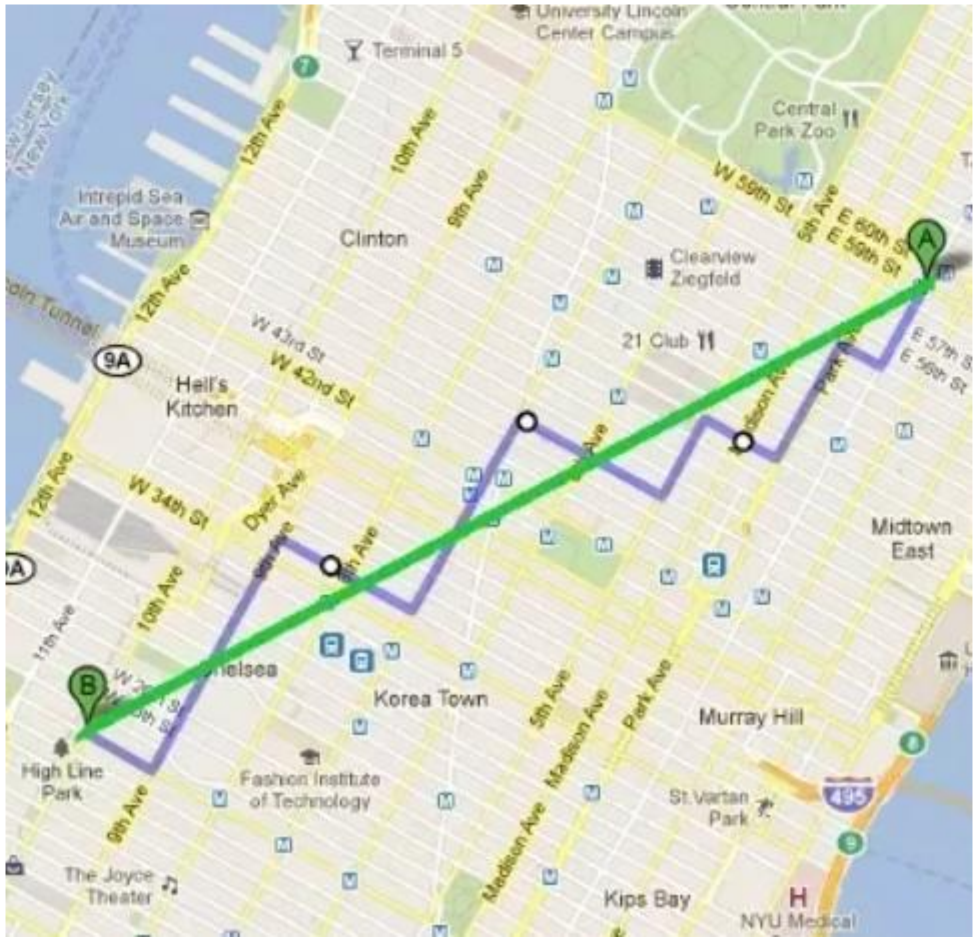


- Euclidean distance: $\sqrt{4^2 + 3^2} = 5$

- Manhattan distance: $4 + 3 = 7$

- "inf"-distance: $max\{4{,}3\} = 4$

Some problems with Euclidean distance



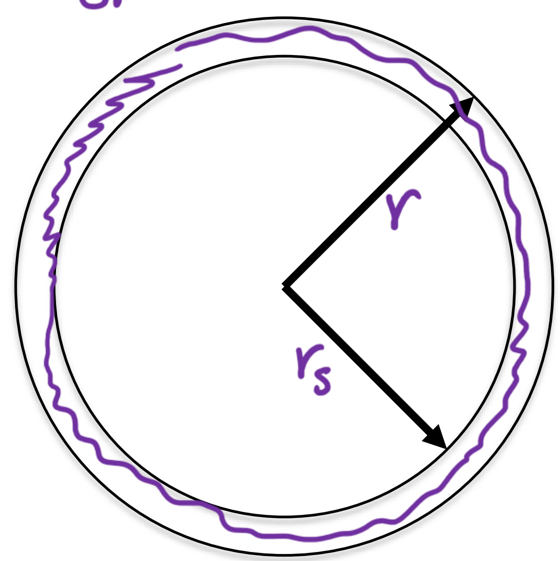
$V_T = \frac{4}{3}\pi r^d = c r^d$

$V_I = c r_s^d$

$V_{shell} = V_T - V_I = c r^d - c r_s^d$

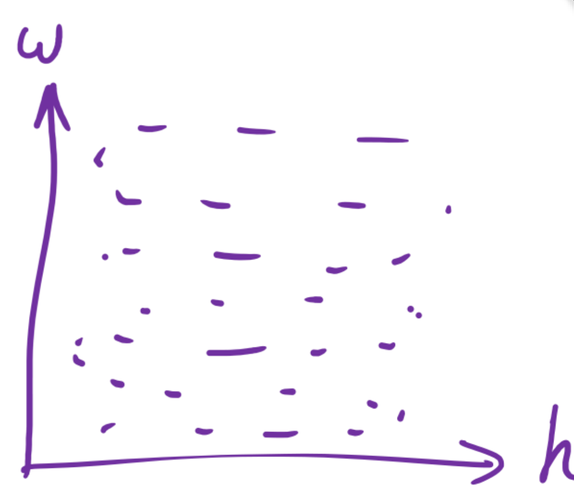
$\frac{V_{shell}}{V_{Total}} = \frac{c r^d - c r_s^d}{c r^d} = 1 - \left(\frac{r_s}{r}\right)^d$

$0 < \frac{r_s}{r} \leq 1$



$d \to \infty \rightsquigarrow \left(\frac{r_s}{r}\right)^d \rightsquigarrow 0$

$V_{shell} \approx V_{Total}$



Curse of dimensionality

Hamming Distance

- Manhattan distance is also called *Hamming distance* when all features are binary
  - Count the number of difference between two binary vectors
  - Example, $x, y \in \{0,1\}^{17}$

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
x	0	1	1	0	0	1	0	0	1	0	0	1	1	1	0	0	1
y	0	1	1	1	0	0	0	0	1	1	1	1	1	1	0	1	1

$$d(x, y) = 5$$

Edit Distance

- Transform one of the objects into the other, and measure how much effort it takes

$$
\begin{array}{lcccccccccc}
x & \mathtt{I} & \mathtt{N} & \mathtt{T} & \mathtt{E} & * & \mathtt{N} & \mathtt{T} & \mathtt{I} & \mathtt{O} & \mathtt{N} \\
 & | & | & | & | & | & | & | & | & | & | \\
y & * & \mathtt{E} & \mathtt{X} & \mathtt{E} & \mathtt{C} & \mathtt{U} & \mathtt{T} & \mathtt{I} & \mathtt{O} & \mathtt{N} \\
 & \mathtt{d} & \mathtt{s} & \mathtt{s} & & \mathtt{i} & \mathtt{s} & & & &
\end{array}
$$

d: deletion (cost 5)
s: substitution (cost 1)
i: insertion (cost 2)

$$d(x, y) = 5 \times 1 + 3 \times 1 + 1 \times 2 = 10$$

Outline

- Clustering
- Distance Function
- K-Means Algorithm
- Analysis of K-Means

K ⇝ the number of clusters

buckets

segments

components

C_1, C_2

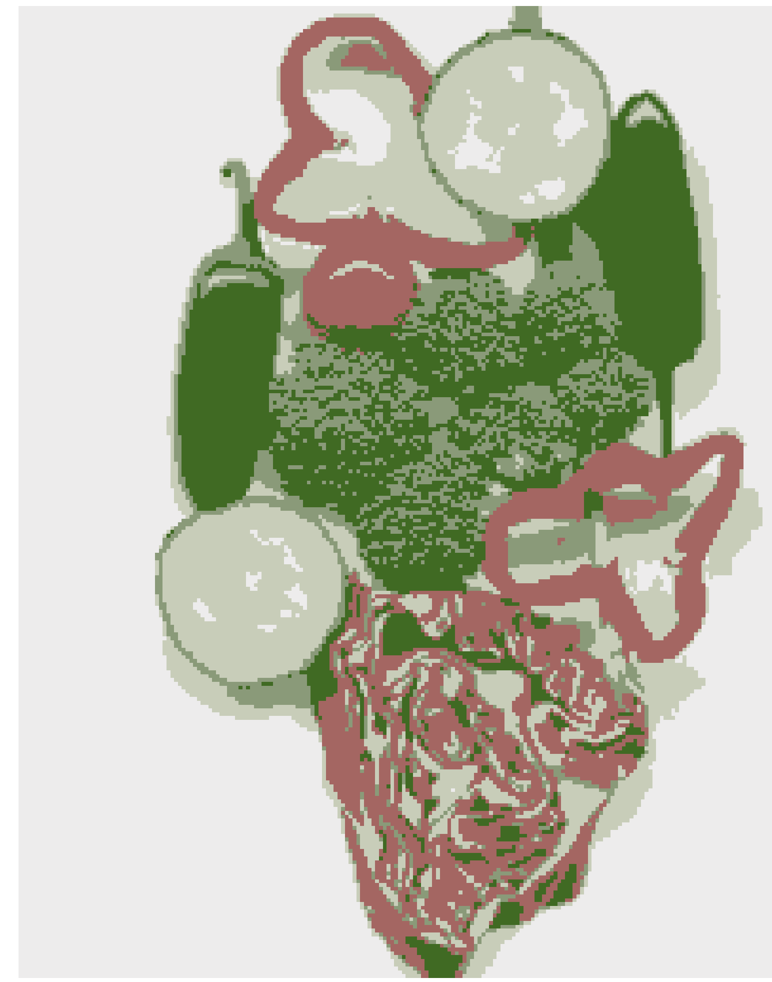
cluster 1

cluster 2

Results of K-Means Clustering:

Image

Clusters on intensity



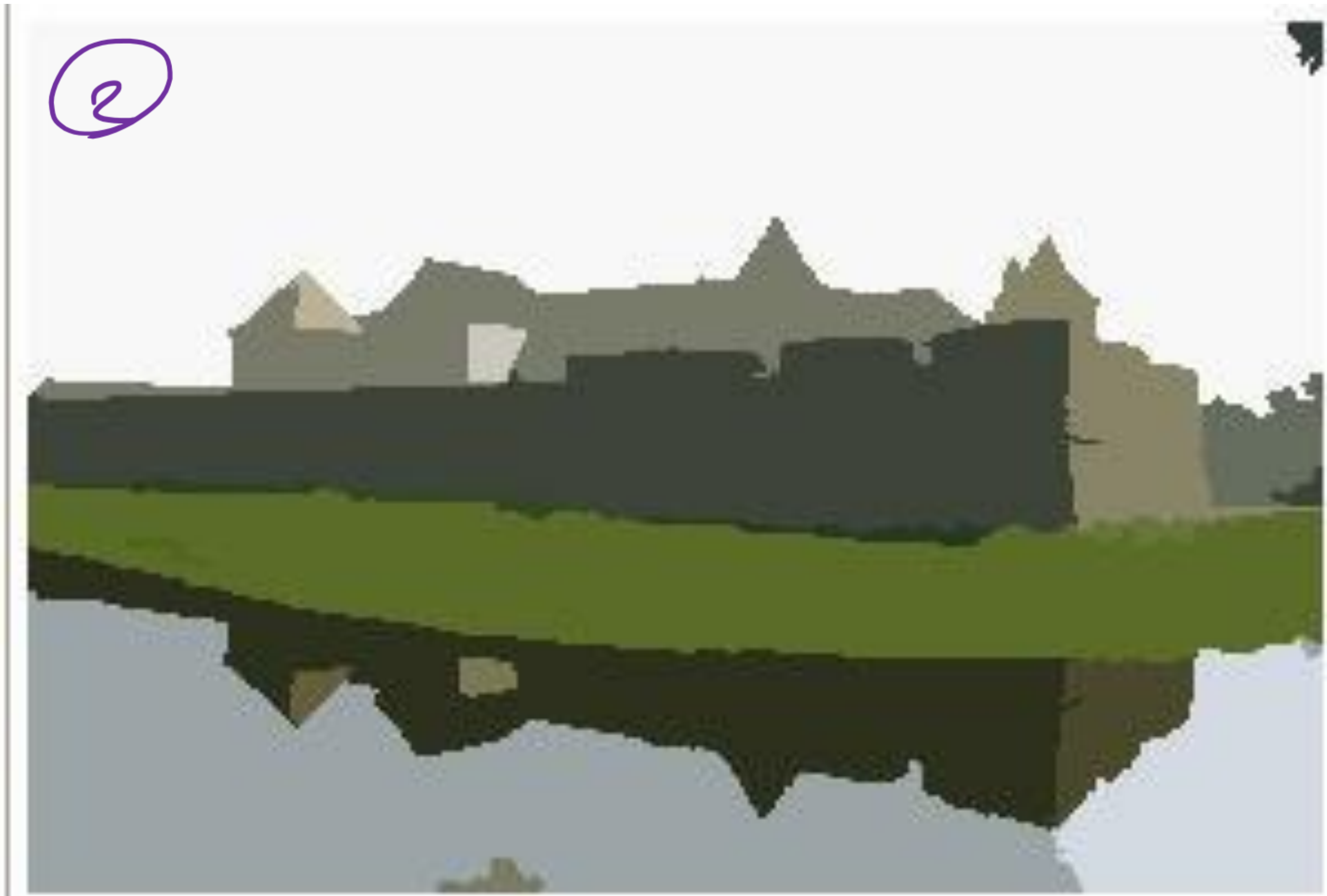
Clusters on color

K-means clustering using intensity alone and color alone

Image

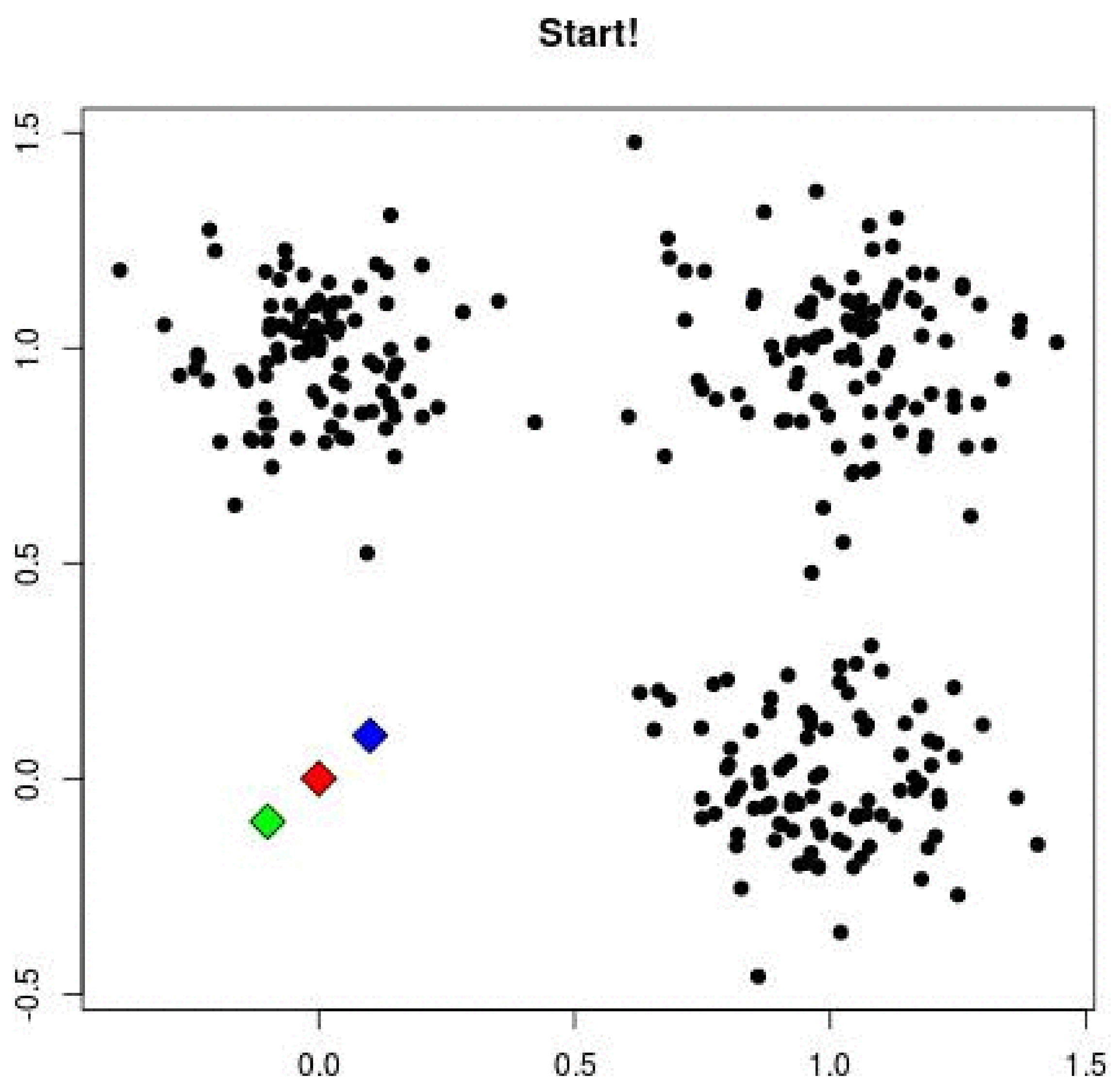
Clusters on color

K-means using color alone, 11 segments (clusters)

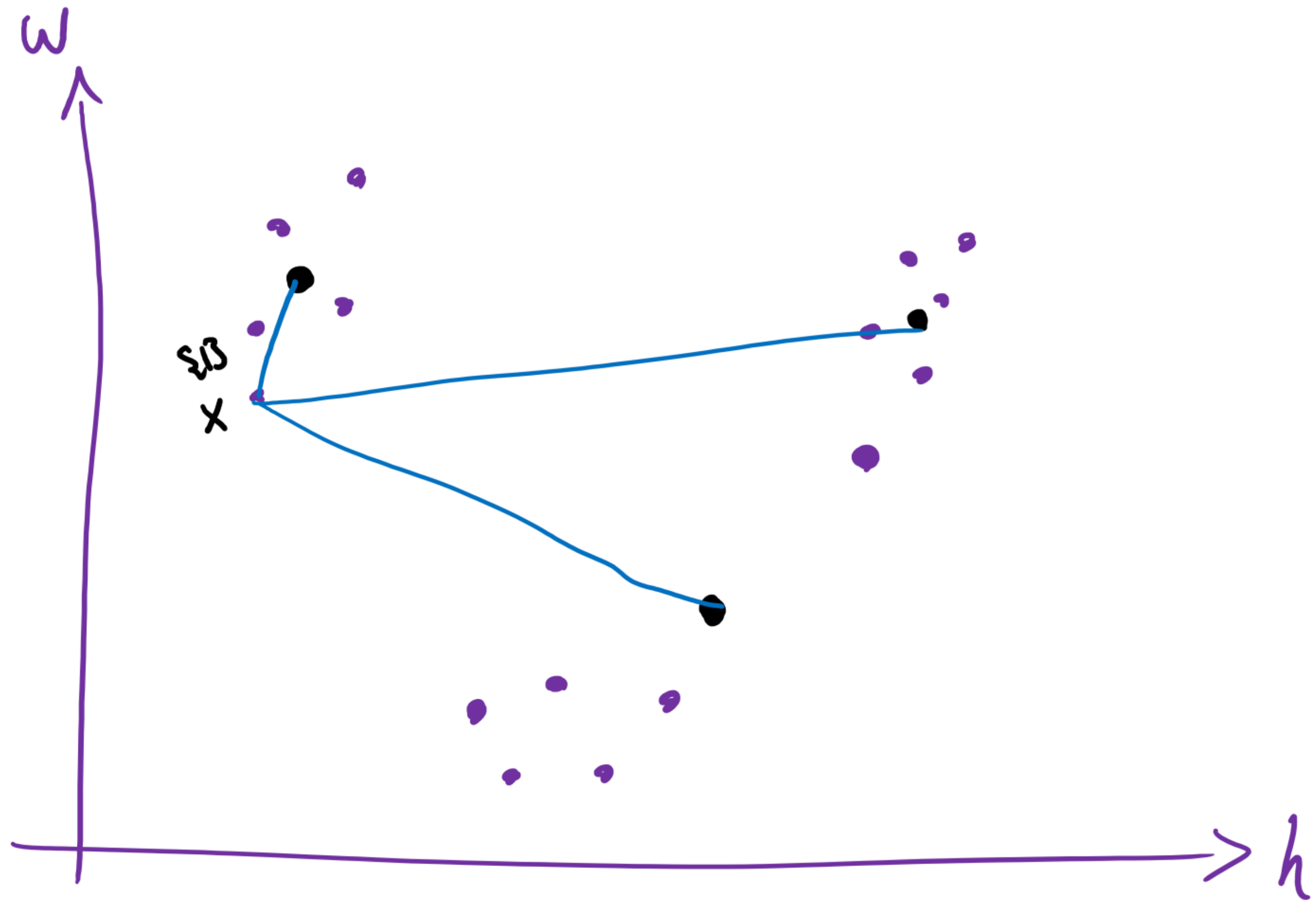


* Pictures from Mean Shift: A Robust Approach toward Feature Space Analysis, by D. Comaniciu and P. Meer http://www.caip.rutgers.edu/~comanici/MSPAMI/msPamiResults.html

K-Means Algorithm

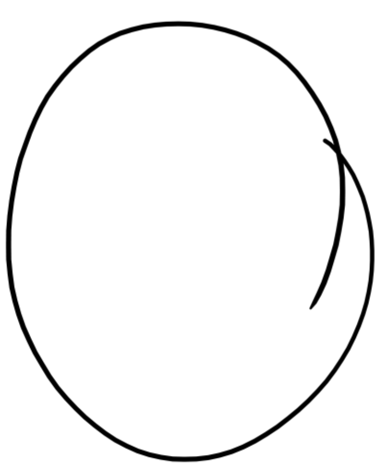


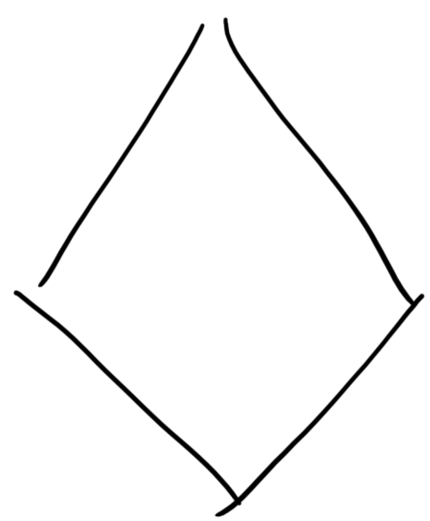
[Visualizing K-Means Clustering]



$$X = \begin{array}{c} \\ x^{\{1\}} \\ \\ \\ x^{\{n\}} \end{array} \begin{array}{c} \begin{array}{cc} h & w \end{array} \\ \left[\begin{array}{cc} & \\ & \\ & \\ & \end{array} \right] \end{array}$$

$$C = \begin{array}{c} \\ c_1 \\ c_2 \\ c_3 \end{array} \begin{array}{c} \begin{array}{cc} h & w \end{array} \\ \left[\begin{array}{cc} & \\ & \\ & \end{array} \right] \end{array}$$





K-Means Algorithm

- Initialize k cluster centers, $\{c_1, c_2, \dots, c_k\}$, randomly

- Do

  - Decide the cluster memberships of each data point, x_i by assigning it to the nearest cluster center (cluster assignment)

$$\pi(i) = argmin_{j=1,\dots,k} \quad \left\|x_i - c_j\right\|^2$$

Expectation

  - Adjust the cluster centers (center adjustment)

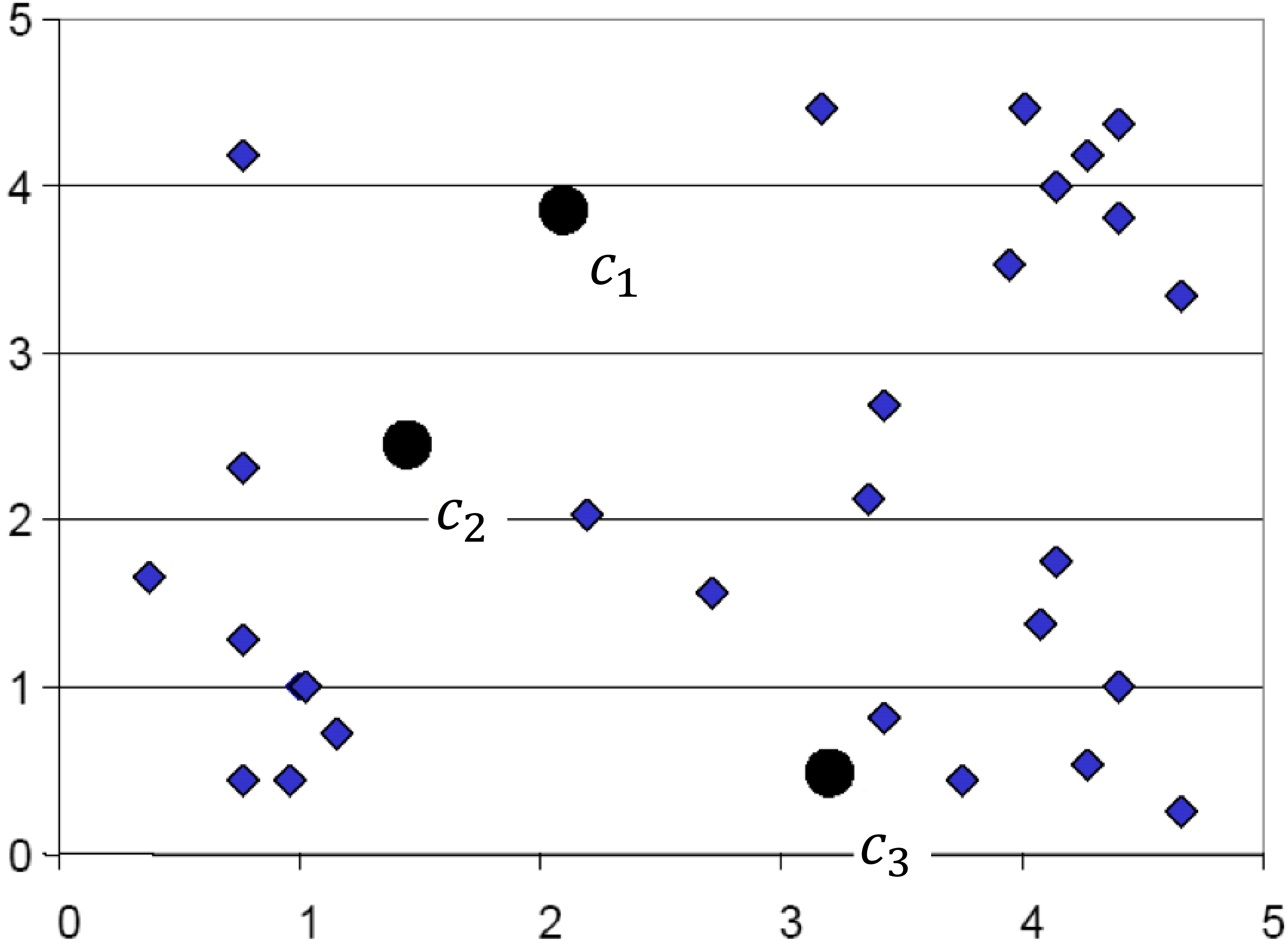
$$c_j = \frac{1}{|\{i: \pi(i) = j\}|} \sum_{i:\pi(i)} x_i$$

Maximization

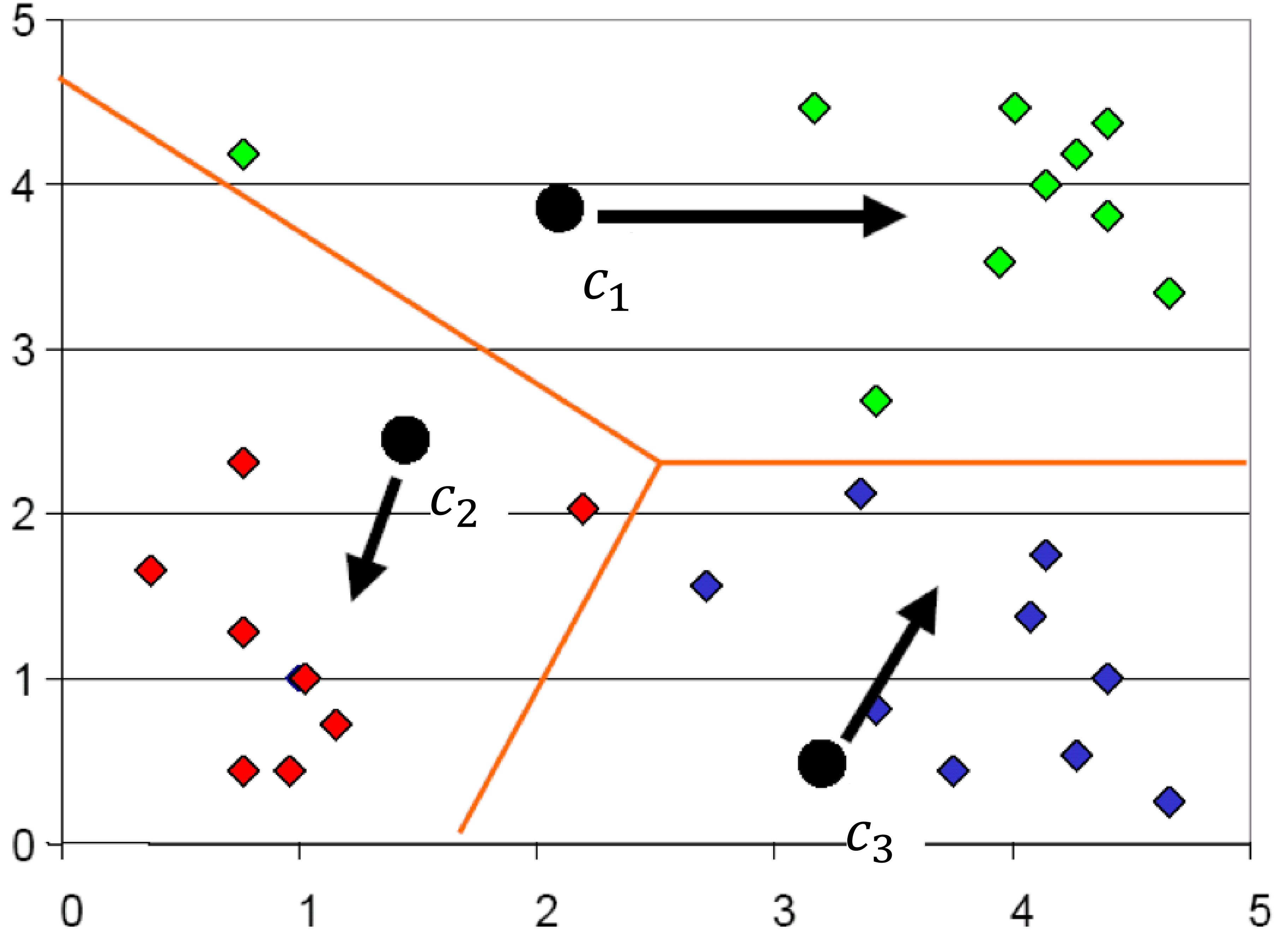
EM algorithm

- While any cluster center has been changed

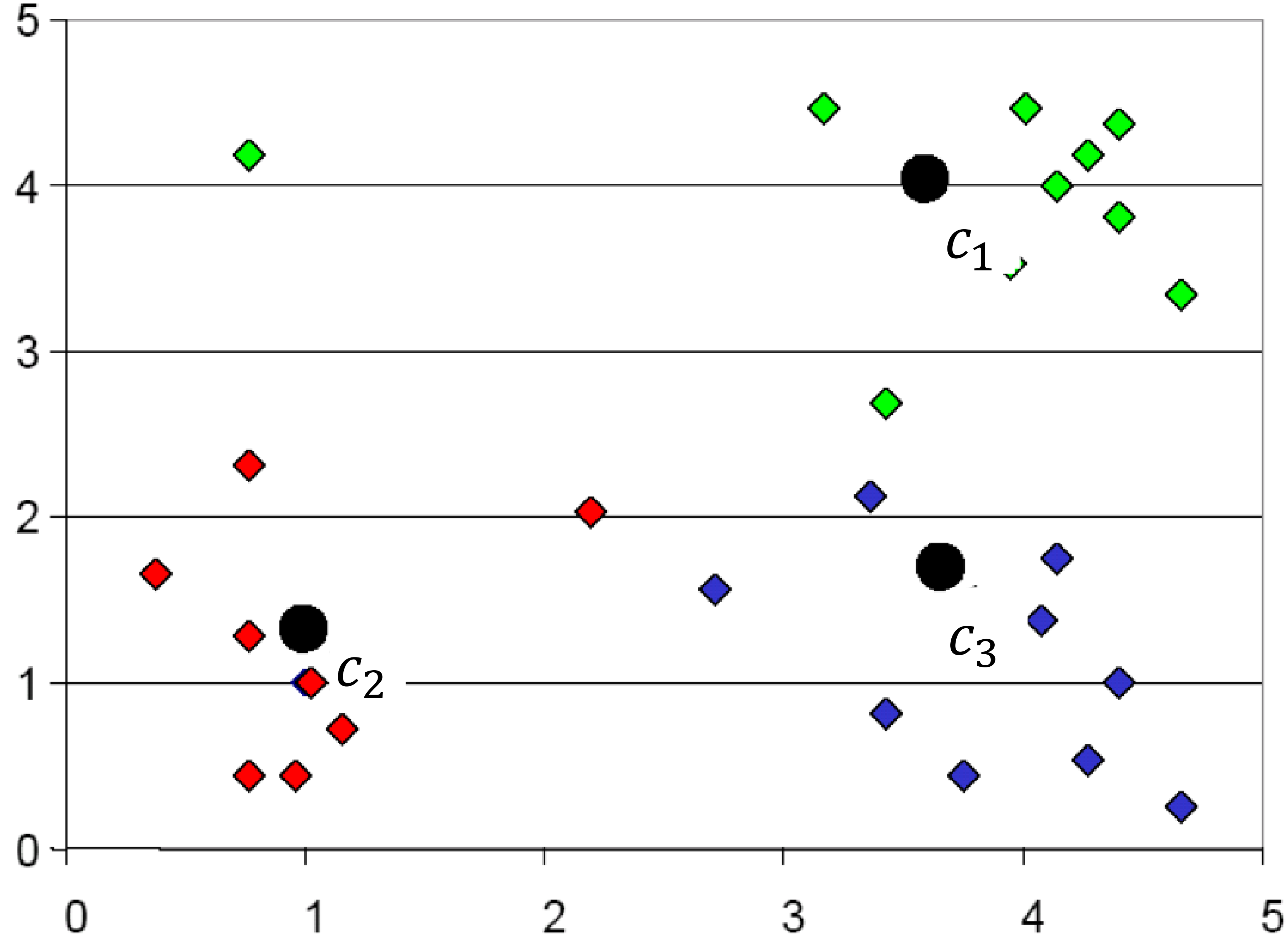
K-Means: Step 1



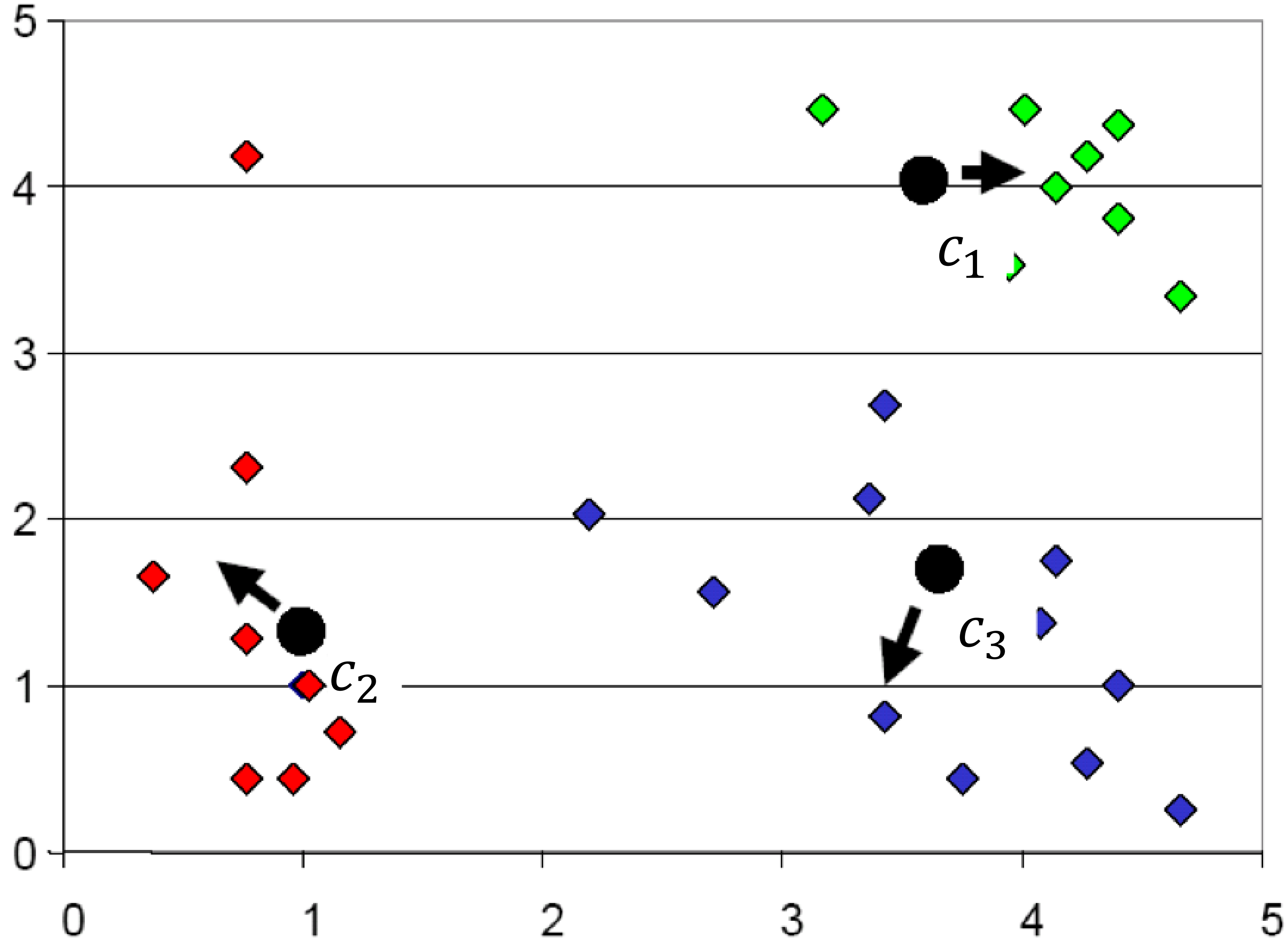
K-Means: Step 2



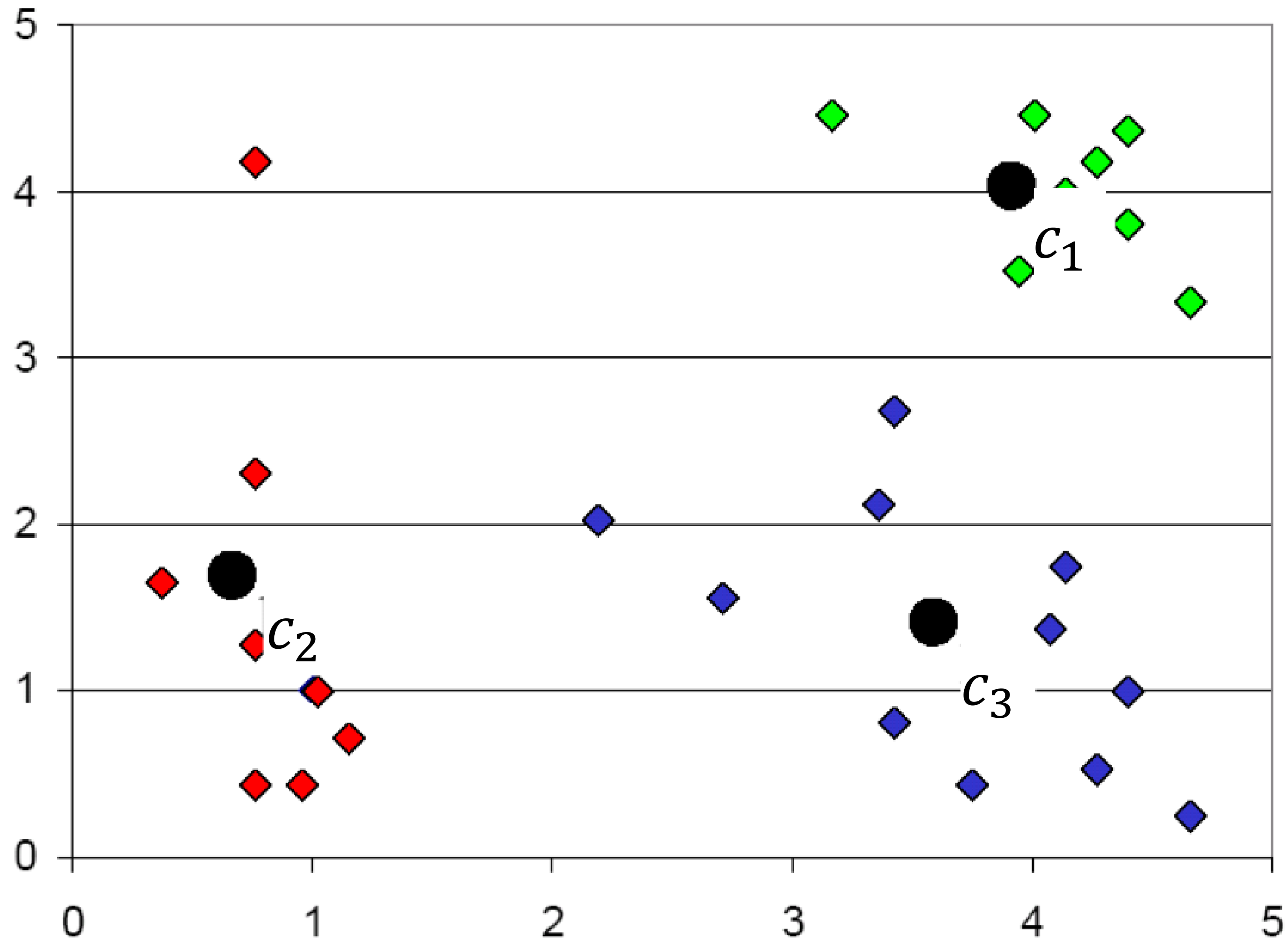
K-Means: Step 3



K-Means: Step 4



K-Means: Step 5



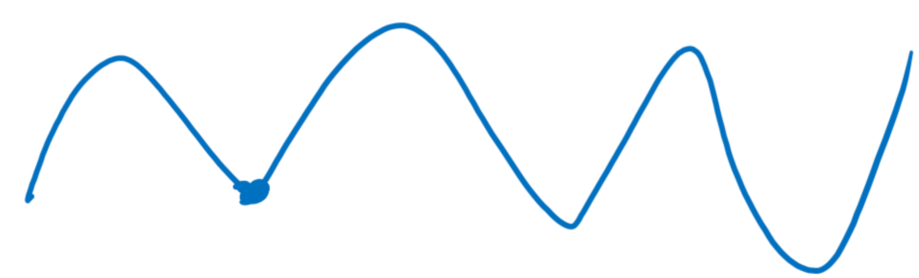
Outline

- Clustering
- Distance Function
- K-Means Algorithm
- Analysis of K-Means

Questions

- Will different initialization lead to different results?
  - Yes
  - No
  - Sometimes

- Will the algorithm always stop after some iteration?
  - Yes
  - No (we have to set a maximum number of iterations)
  - Sometimes



Formal Statement of the Clustering Problem

- Given n data points, $\{x_1, x_2, \dots, x_n\}$ $x \in R^d$
- Find k cluster centers, $\{c_1, c_2, \dots, c_k\}$ $c \in R^d$
- And assign each datapoint i to one cluster, $\pi(i) \in \{1, \dots, k\}$
- Such that the averaged square distances from each datapoint to its respective cluster center is small

$\text{Sum } c_1 = a_1 + b_1 + d_1$

$\text{Sum } c_2 = a_2 + b_2 + d_2$

$$\text{distortion} = \min_{c,\pi} \sum_{i=1}^{n} \left\| x_i - c_{\pi(i)} \right\|^2$$

$\text{Sum}_{c_3} = a_3 + b_3 + d_3$

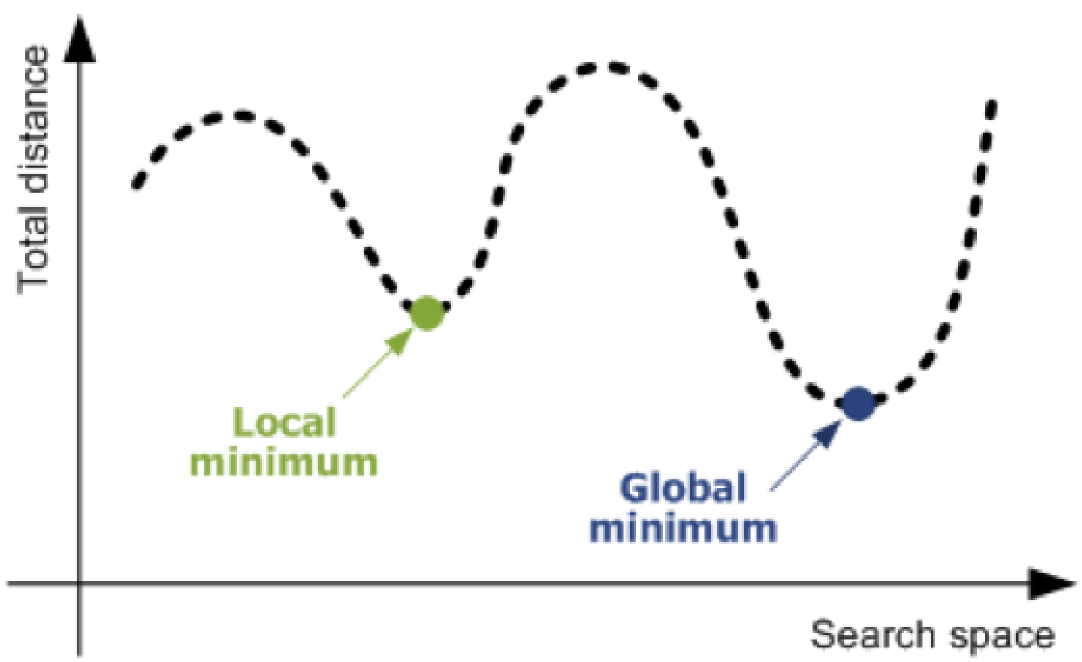
$$= \text{SUM}_{c_1} + \text{SUM}_{c_2} + \text{SUM}_{c_3}$$

Clustering is NP-Hard

- Find k cluster centers, $\{c_1, c_2, \ldots, c_k\}$ $c \in R^d$, and assign each data point i to one cluster, $\pi(i) \in \{1, \ldots, k\}$, to minimize

$$\min_{c,\pi} \sum_{i=1}^{n} \left\| x_i - c_{\pi(i)} \right\|^2$$

- A search problem over the space of discrete assignments
  - For all n data point together, there are k^n possibility
  - The cluster assignment determines cluster centers, and vice versa



- For all n data point together, there are k^n possibility

$X = \{A,B,C\}$

n=3 (data points)

$k^n = 2^3 = 8$

k=2 clusters of two members

Cluster 1	Cluster 2
{ }	A, B, C
A, B, C	{ }
A	B, C
B	A, C
C	A, B
A, B	C
A, C	B
B, C	A

Convergence of K-Means

- Will kmeans objective oscillate?

$$\min_{c,\pi} \sum_{i=1}^{n} \left\| x_i - c_{\pi(i)} \right\|^2$$

- The minimum value of the objective is finite

- Each iteration of kmeans algorithm decrease the objective
  - Cluster assignment step decreases objective
    - $\pi(i) = argmin_{j=1,\ldots,k} \ \left\| x_i - c_{\pi(j)} \right\|^2$ for each data point i
  - Center adjustment step decreases objective
    - $c_i = \frac{1}{|\{i:\pi(i)=j\}|} \sum_{i:\pi(i)=j} x_i = argmin_c \sum_{i:\pi(i)=j} \left\| x_i - c_{\pi(j)} \right\|^2$

Time Complexity

$X = \{x_1, x_2, \ldots, x_d\}$

$Y = \{y_1, y_2, \ldots, y_d\}$

- Assume computing distance between two instances is O(d) where d is the dimensionality of the vectors.

$\|X - Y\|^2 = (x_1 - y_1)^2 + \cdots + (x_d - y_d)^2$

$O(d)$

- Reassigning clusters for all datapoints:
  - O(kn) distance computations (when there is one feature)
  - O(knd) (when there is d features)

$O(nd)$

$O(knd)$

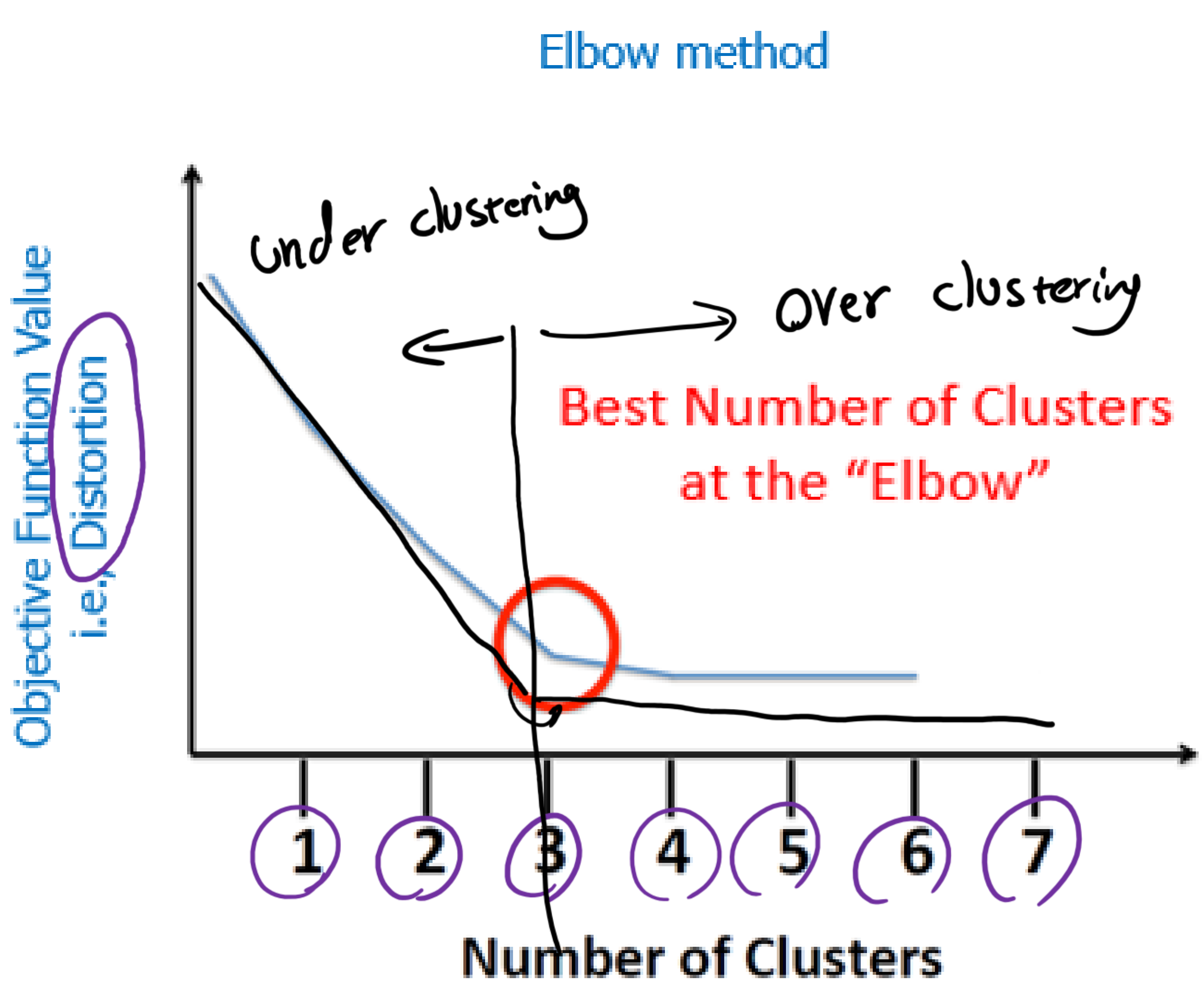
- Computing centroids: Each instance vector gets added once to some centroid (Finding centroid for each feature): O(nd).

- Assume these two steps are each done once for I iterations: O(Iknd).

$O(Iknd)$

Slide credit: Ray Mooney.

How to Choose K?



$a_1+b_1+c_1 + a_2+b_2+c_2$

Distortion score: computing the sum of squared distances from each point to its assigned center

Image credit: Dileka Madushan.