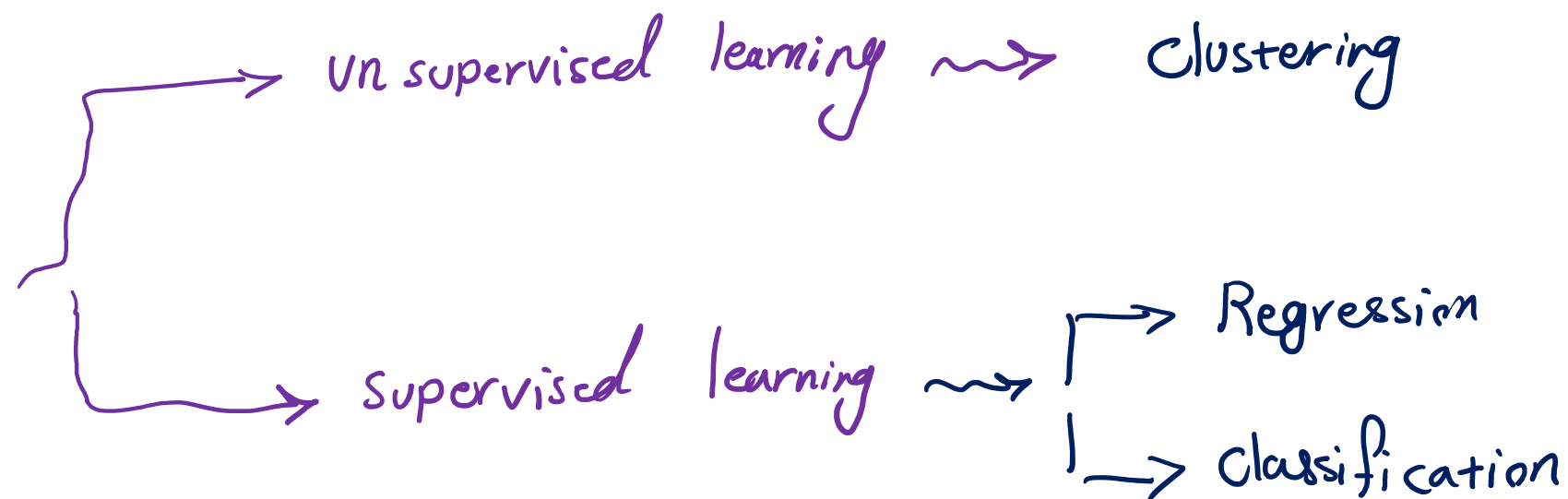


Lin algebra \rightsquigarrow Prob & Stats \rightsquigarrow Info theory \rightsquigarrow Optimization

Toolbox \rightsquigarrow our main python friend \rightsquigarrow np

$X_{n \times d}$ \rightsquigarrow training data

~~$Y_{n \times 1}$~~ \rightsquigarrow training labels or target values




Clustering Analysis and K-Means

Mahdi Roozbahani
Georgia Tech

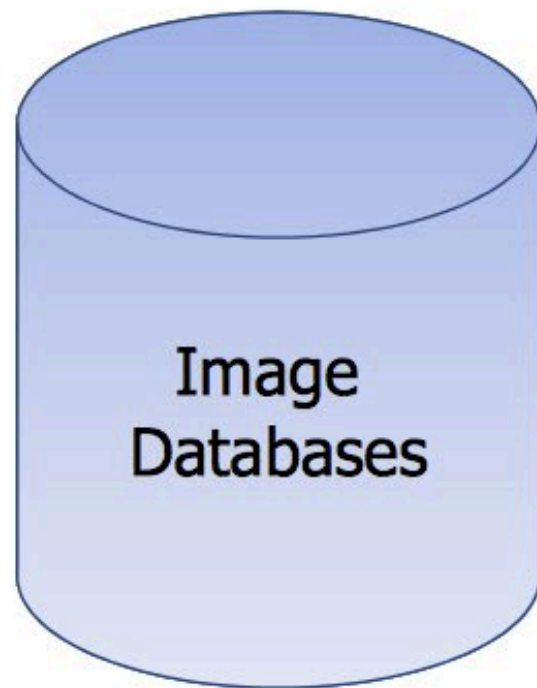
60+ hours on 16 GPU nvidia CUDA cluster.



Outline

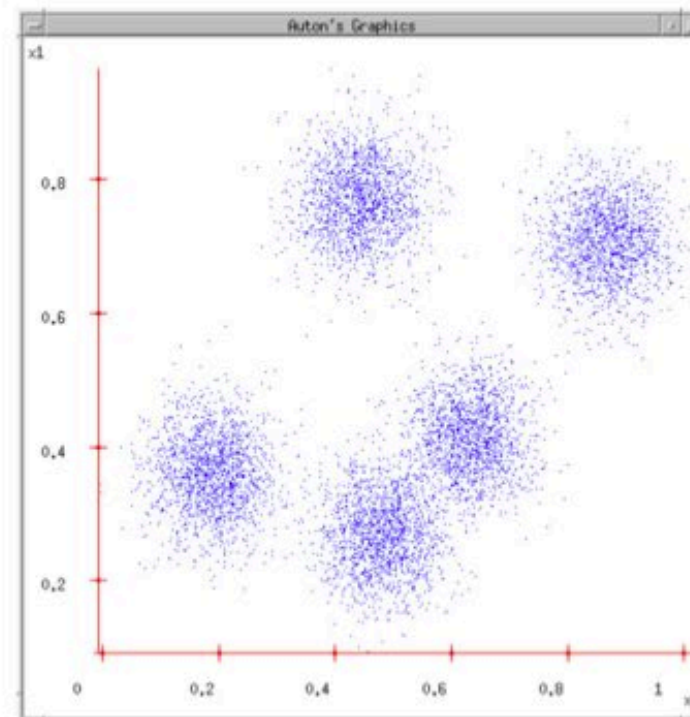
- Clustering 
- Distance Function
- K-Means Algorithm
- Analysis of K-Means

Clustering Images



Goal of clustering:

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

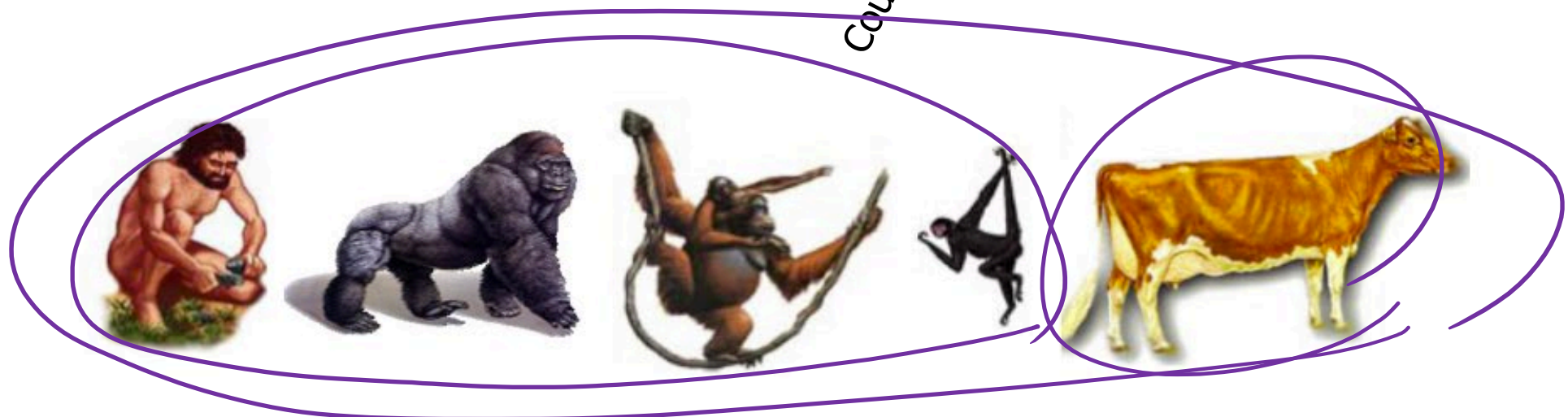


Clustering Other Objects



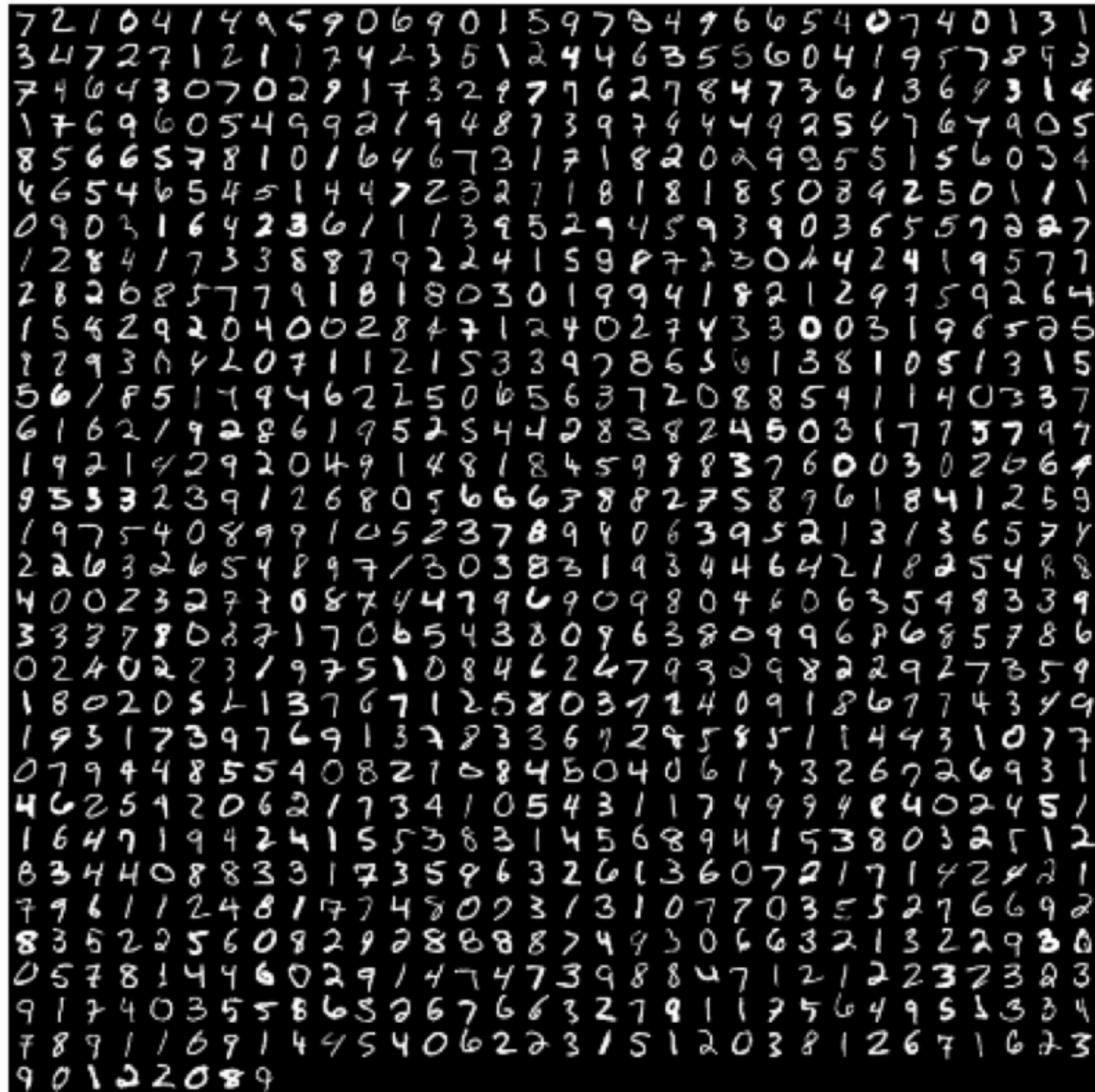
Piotr
 Belarusian
Pyotr
 Azerbaijani
Petros
 Greek
Pietro
 Italian
Pedro
 Portuguese
Pierre
 French
Piero
 Italian
Peter
 Dutch
Peder
 Danish
 Couldn't find it – Finish?
 Peka
 Irish
 Peadar

Linguistic Similarity

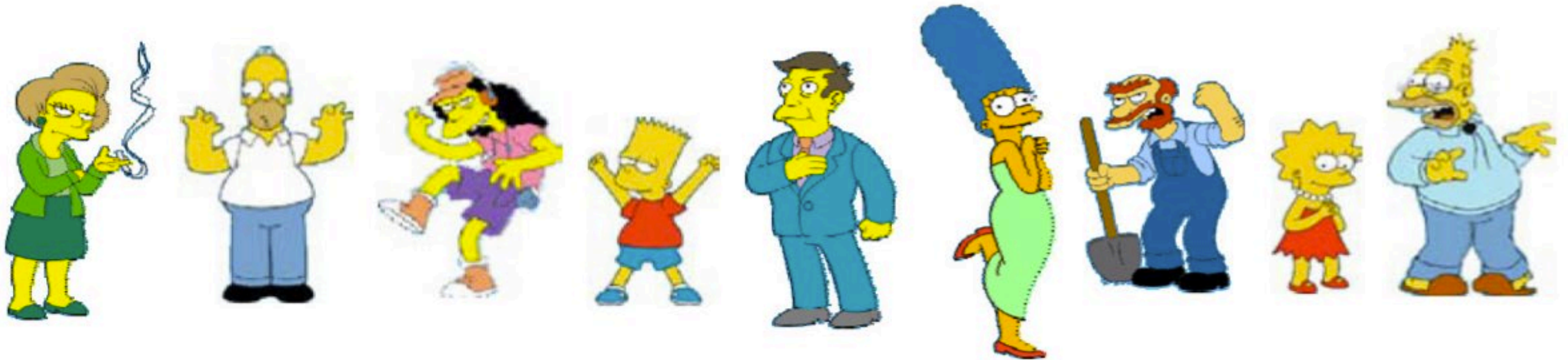


Clustering Hand Digits

0 1 2 3 4 5 6 7 8 9

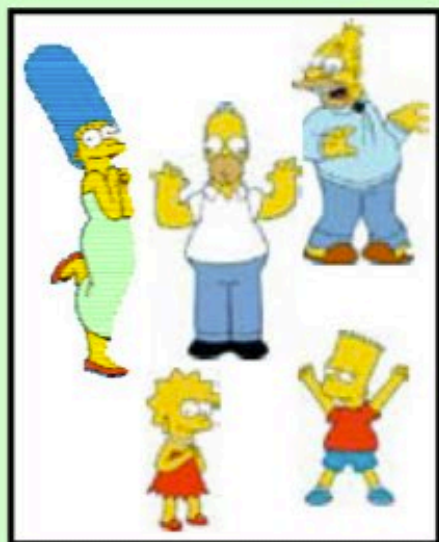


Clustering is Subjective



What is consider similar/dissimilar?

Clustering is subjective



Simpson's Family



School Employees



Females



Males


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

- Suppose two data points, both in R^d

→ • $x = (x_1, x_2, \dots, x_d)$

→ • $y = (y_1, y_2, \dots, y_d)$

$$\|x - y\|_2^2 = \sum (x_i - y_i)^2$$

- Euclidean distance: $d(x, y) = \sqrt{\sum_{i=1}^d (x_i - y_i)^2} = \|x - y\|_2$

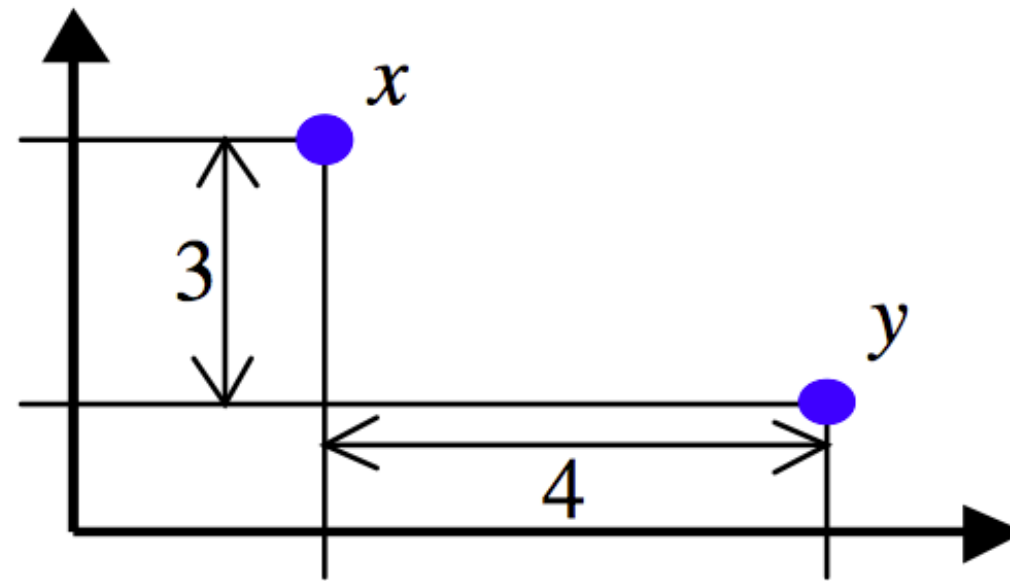
- Minkowski distance: $d(x, y) = \sqrt[p]{\sum_{i=1}^d (x_i - y_i)^p}$

- 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

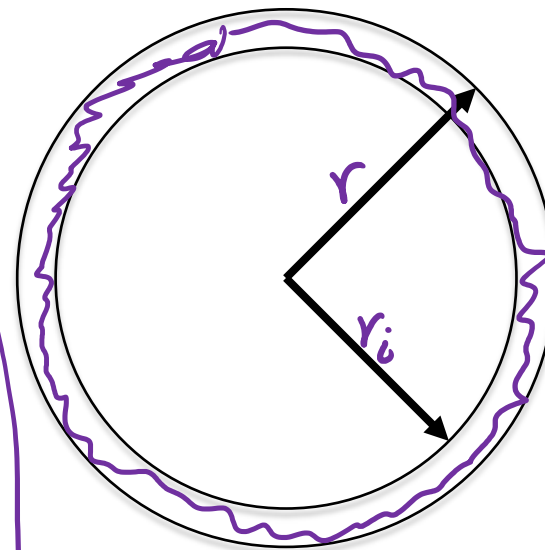


- Euclidean distance: $\sqrt{4^2 + 3^2} = 5$
- Manhattan distance: $4 + 3 = 7$
- “inf”-distance: $\max\{4, 3\} = 4$

Some problems with Euclidean distance



$$V = \frac{4}{3} \pi r^3 = cr^d$$



$$V_{\text{shell}} = cr^d - cr_i^d$$

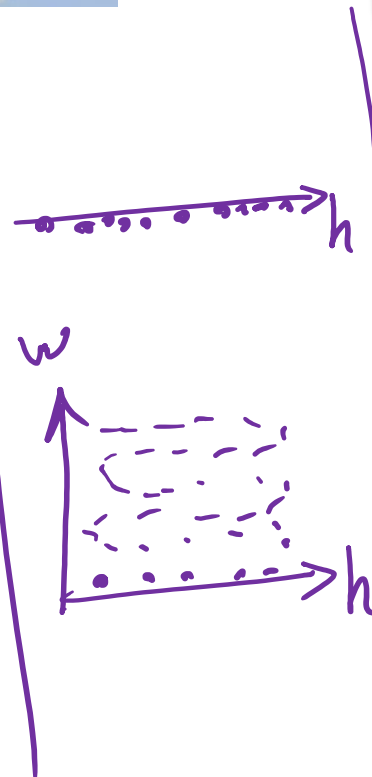
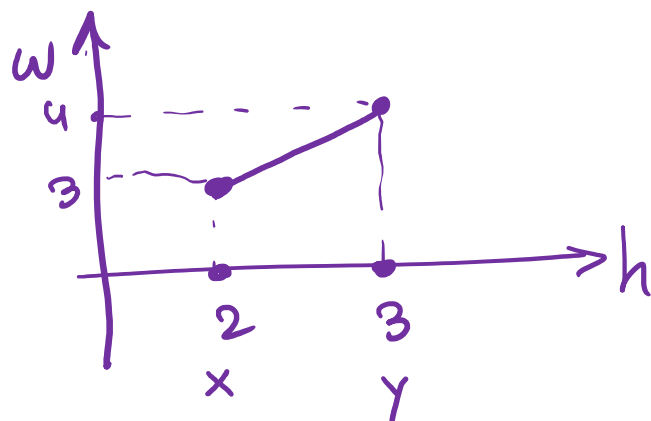
$$V_{\text{total}} = cr^d$$

$$\frac{V_{\text{shell}}}{V_{\text{total}}} = \frac{cr^d - cr_i^d}{cr^d} = 1 - \left(\frac{r_i}{r}\right)^d$$

$$0 < \frac{r_i}{r} < 1$$

$$d \rightarrow \infty \quad V_{\text{shell}} \approx V_{\text{total}}$$

$$X = \begin{bmatrix} h & w \\ 2 & 3 \\ 3 & 4 \end{bmatrix}$$



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

| | | | | | | | | | | |
|-----|---|---|---|---|---|---|---|---|---|---|
| x | I | N | T | E | * | N | T | I | O | N |
| | | | | | | | | | | |
| y | * | E | X | E | C | U | T | I | O | N |
| | d | s | s | | i | s | | | | |

d: deletion (cost 5)

s: substitution (cost 1)

i: insertion (cost 2)

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

| | | | | | | | | | |
|----|---|---|----------|---|----------|---|---|---|---|
| x: | I | N | T | E | N | T | J | O | N |
| y: | I | N | S | E | R | T | I | O | N |
| | - | - | <u>s</u> | - | <u>s</u> | - | - | - | - |


$$d(x, y) = 2 \times 1 = 2$$

d: deletion (cost 5)

s: substitution (cost 1)

i: insertion (cost 2)

Outline

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

Results of K-Means Clustering:



Image



Clusters on intensity



Clusters on color

K-means clustering using intensity alone and color alone



Image



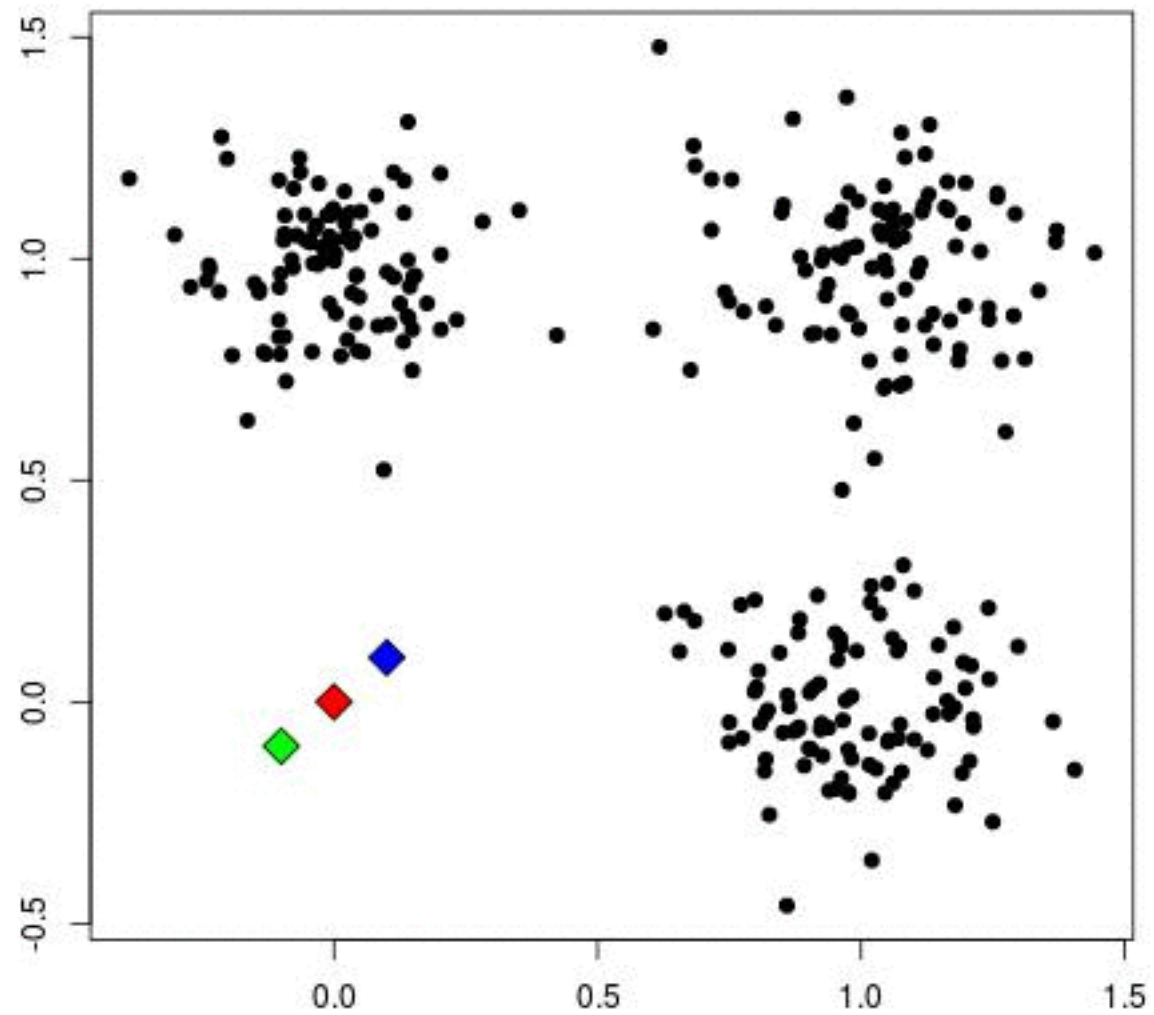
Clusters on color

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



K-Means Algorithm

Start!



[Visualizing K-Means Clustering](#)

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**)

$$\pi(i) = \underset{j=1, \dots, k}{\operatorname{argmin}} \quad \|x_i - c_j\|^2 \quad \rightarrow \text{Expectation}$$

↓
number of clusters

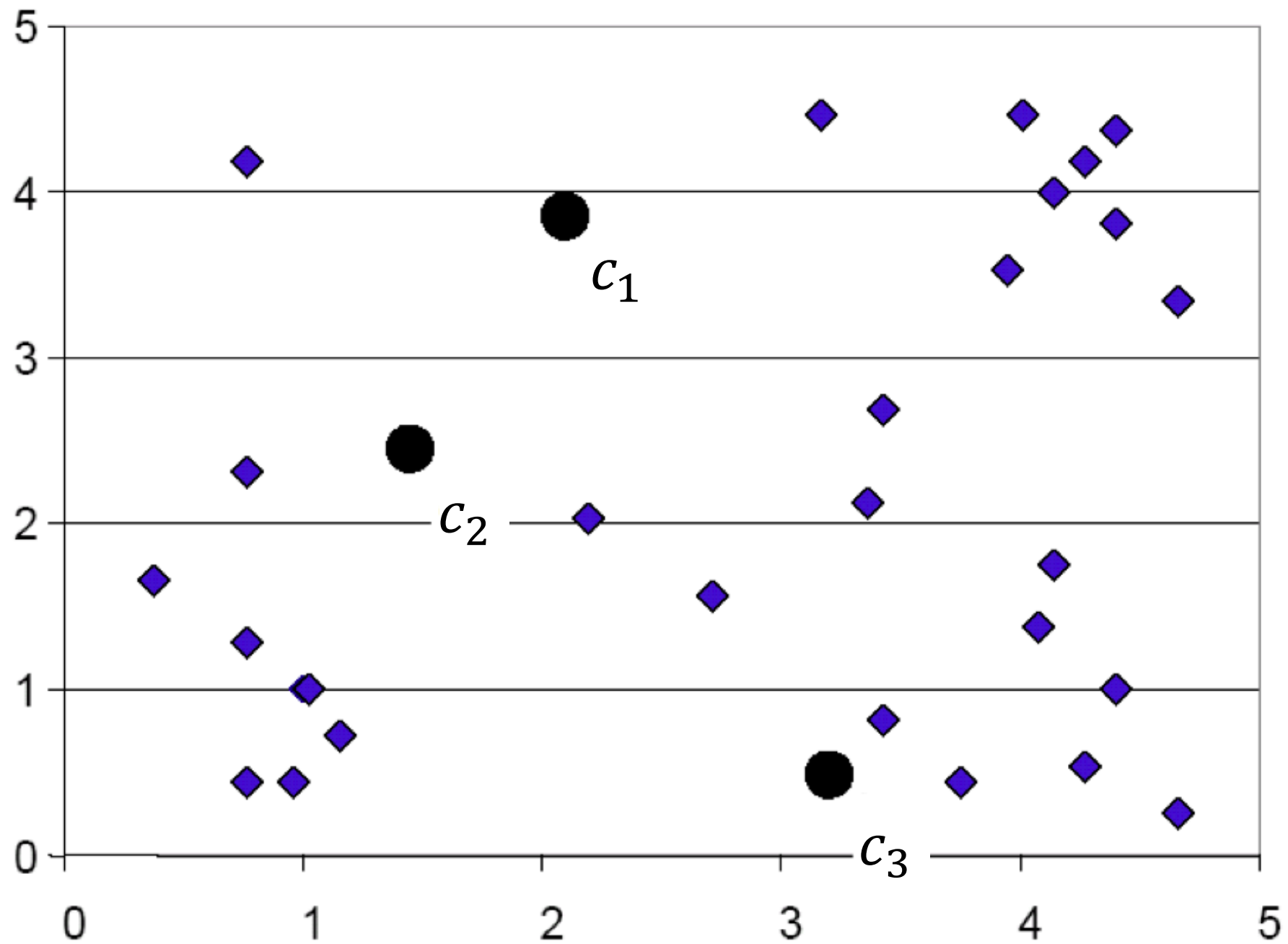
- Adjust the cluster centers (**center adjustment**)

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

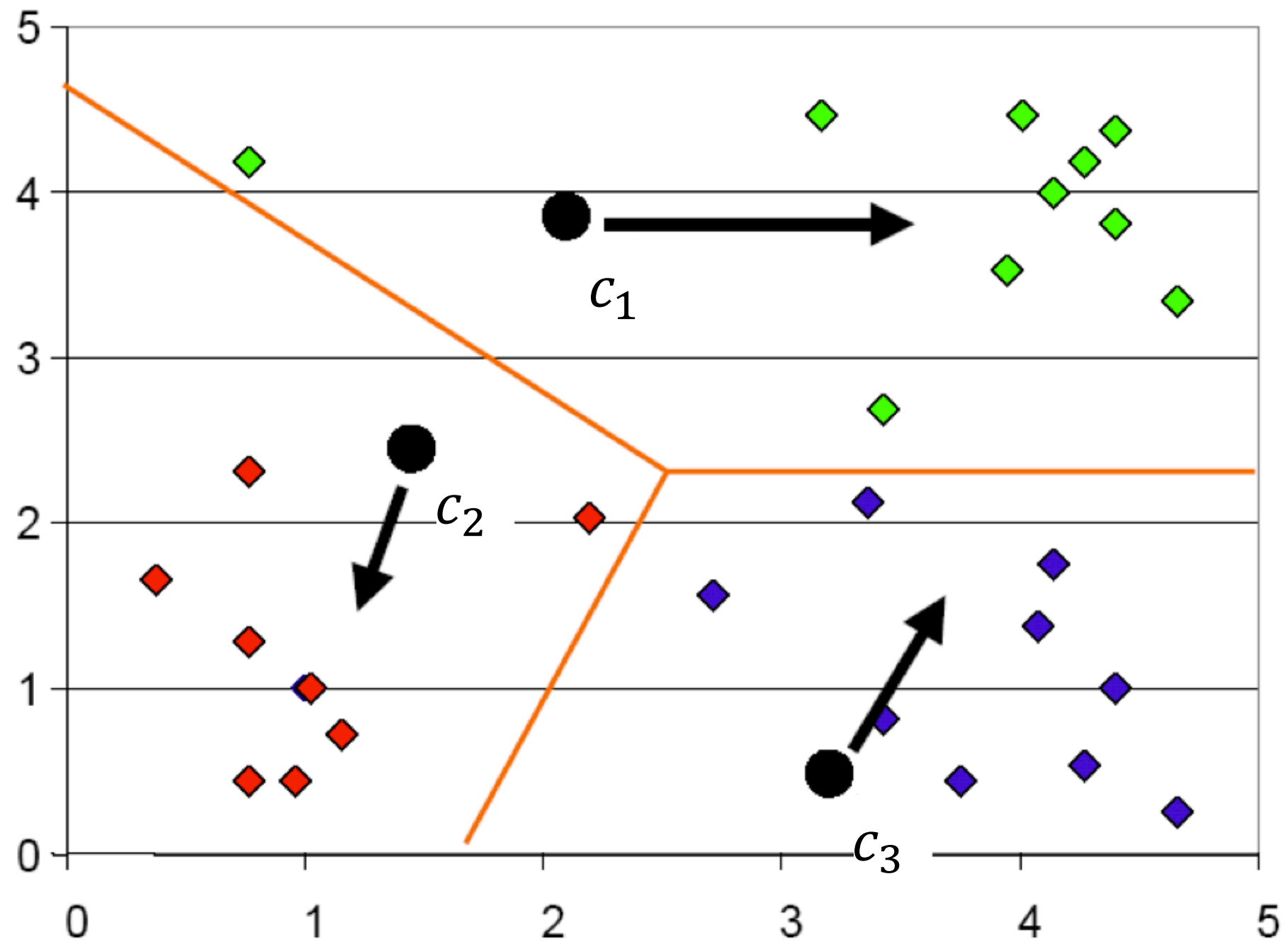
- While any cluster center has been changed

EM

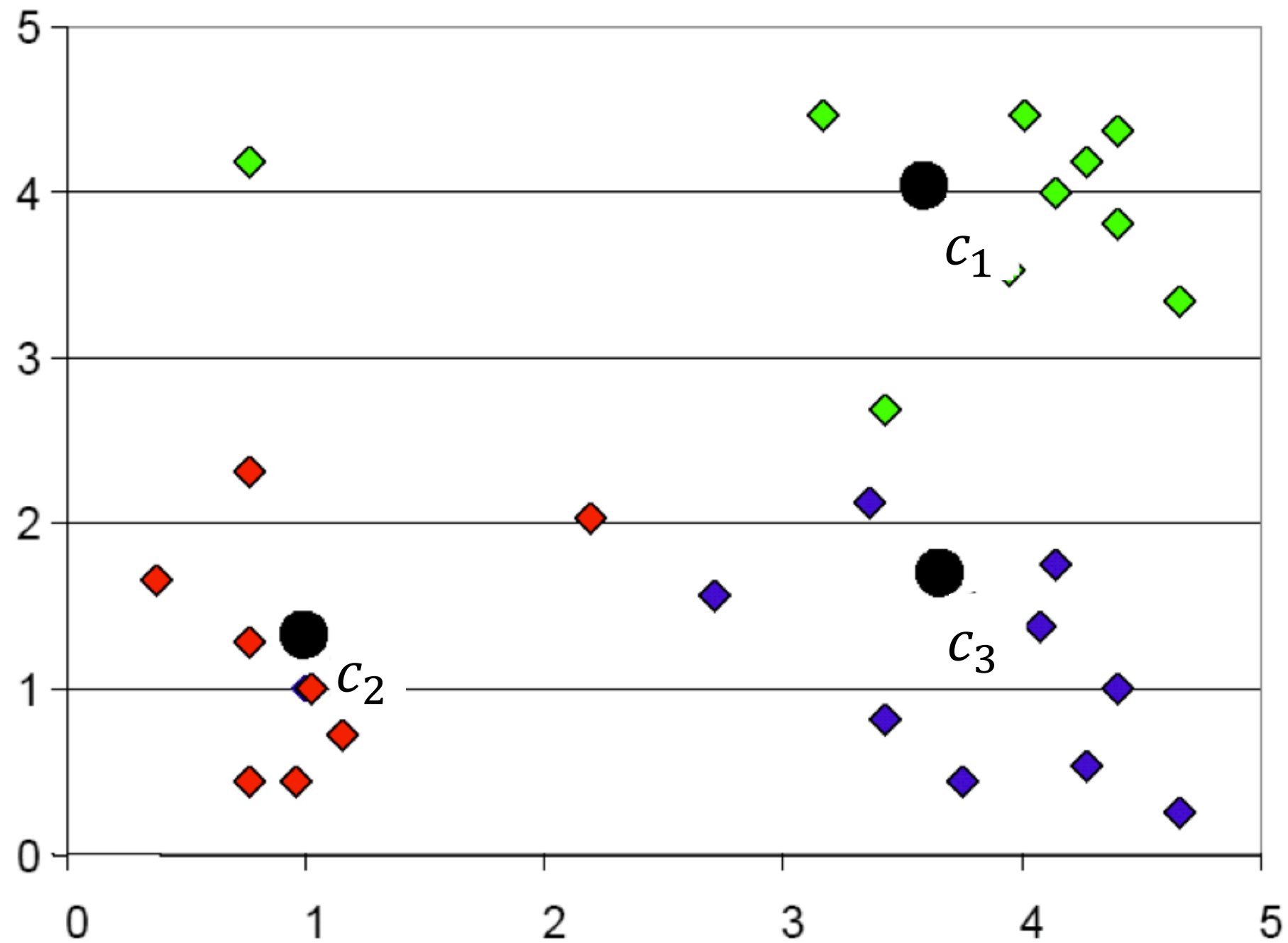
K-Means: Step 1



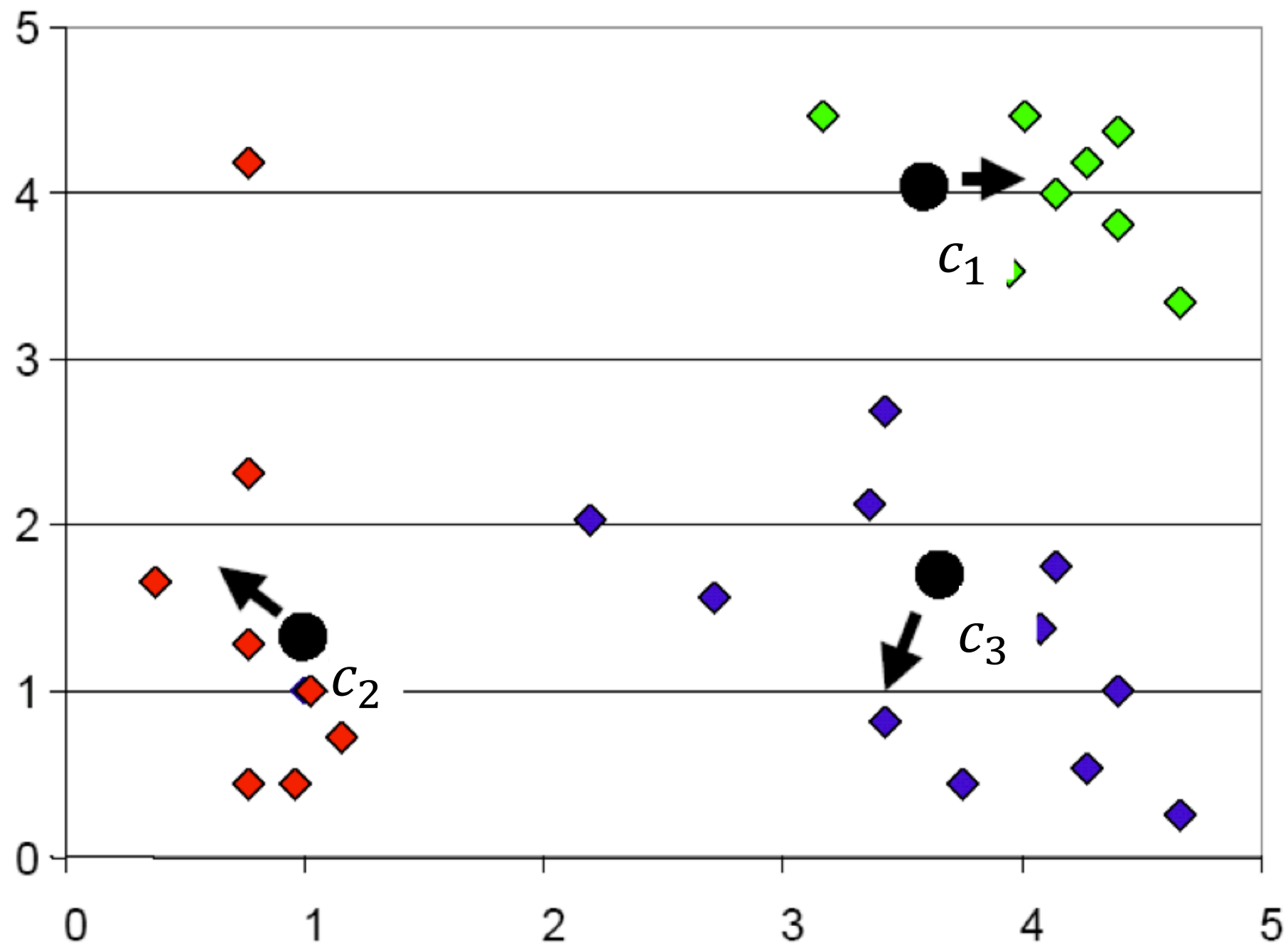
K-Means: Step 2



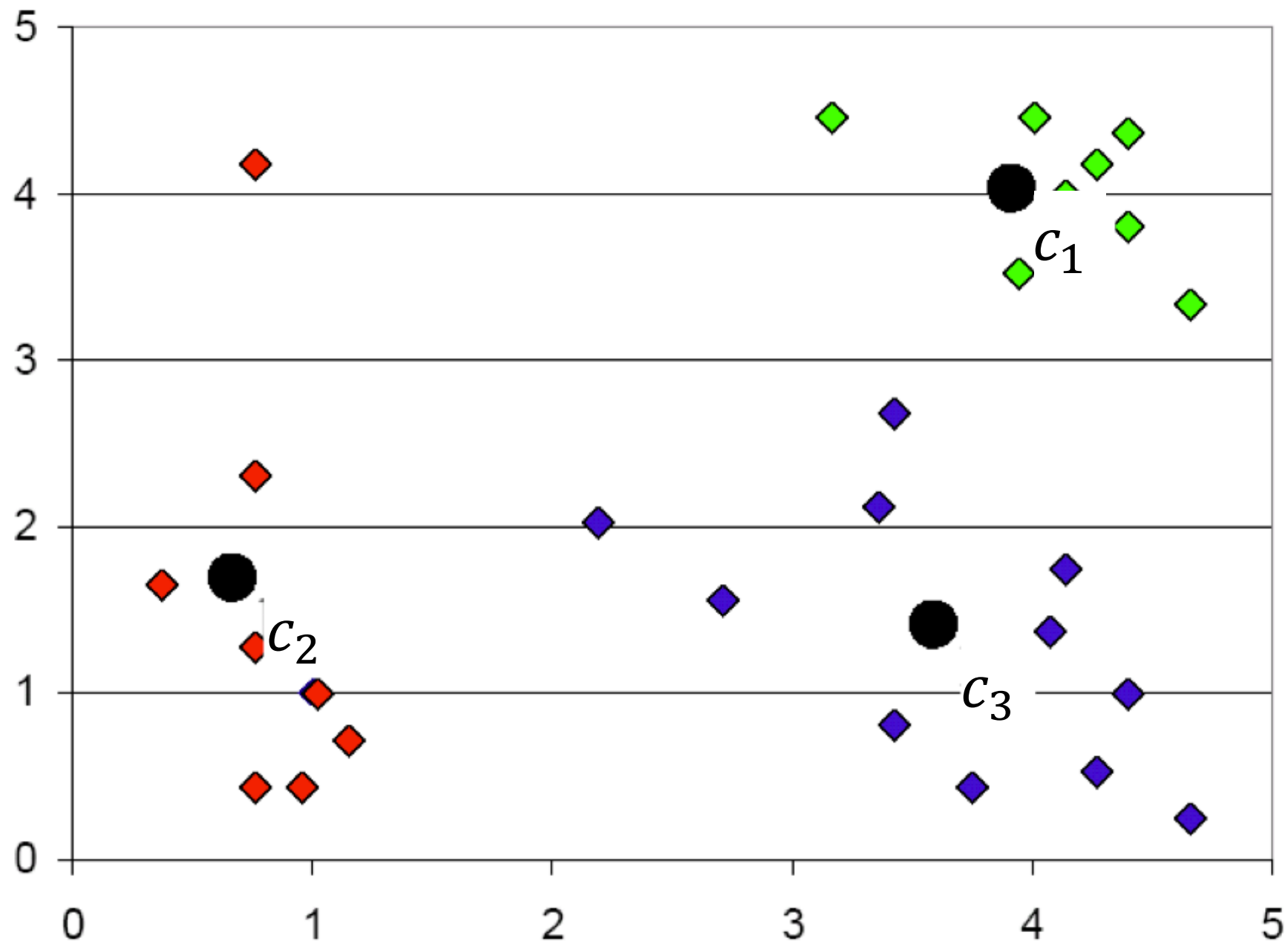
K-Means: Step 3




K-Means: Step 4



K-Means: Step 5



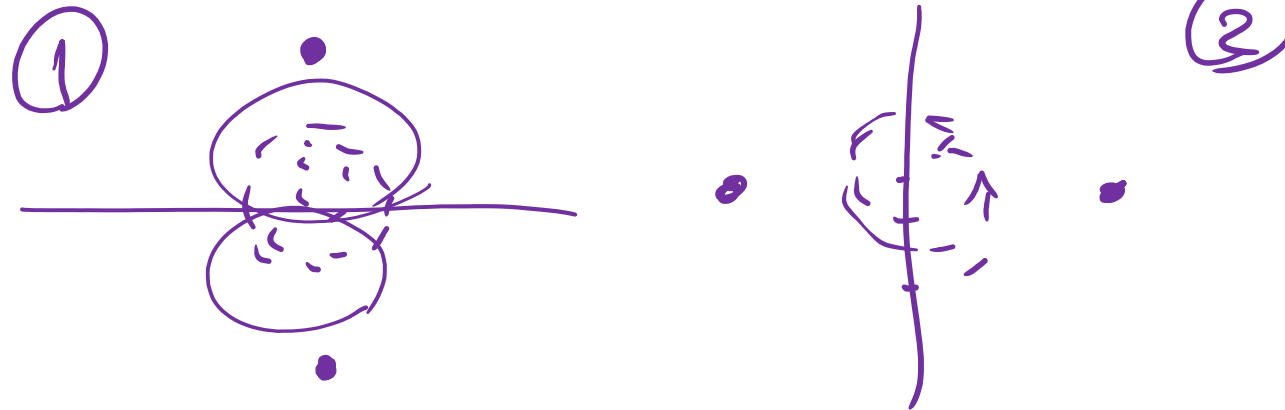
Outline

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

Questions

- Will different initialization lead to different results?

- Yes
- No
- Sometimes




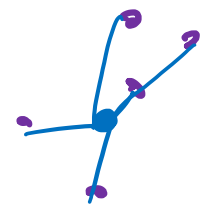
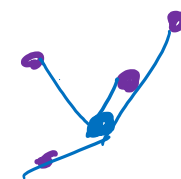
- Will the algorithm always stop after some iteration?

- Yes, Local optimal Solution
- 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


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



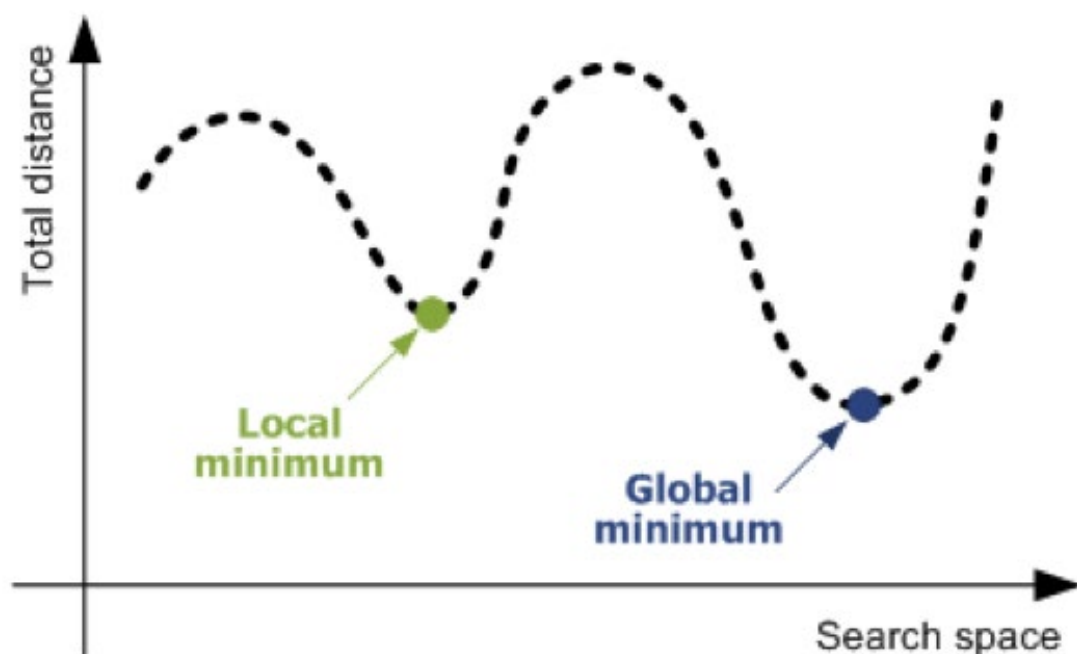
Clustering is NP-Hard

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

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

NP-hard!

- 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
 $2^3 = 8$

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

$n=3$ (data points)

$k=2$ clusters of two members

Cluster 1

$\{ \}$

A, B, C

A

B

C

B, C

A, C

B, C

Cluster 2

A, B, C

$\{ \}$

B, C

A, C

B, C

A

B

C

Convergence of K-Means

- Will kmeans objective oscillate?

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

- The minimum value of the objective is finite
- Each iteration of kmeans algorithm decrease the objective
 - Cluster assignment step decreases objective
 - $\pi(i) = \operatorname{argmin}_{j=1, \dots, k} \|x_i - c_{\pi(j)}\|^2$ for each data point i
 - Center adjustment step decreases objective
 - $c_i = \frac{1}{|\{i: \pi(i)=j\}|} \sum_{i: \pi(i)=j} x_i = \operatorname{argmin}_c \sum_{i: \pi(i)=j} \|x_i - c_{\pi(j)}\|^2$

Time Complexity

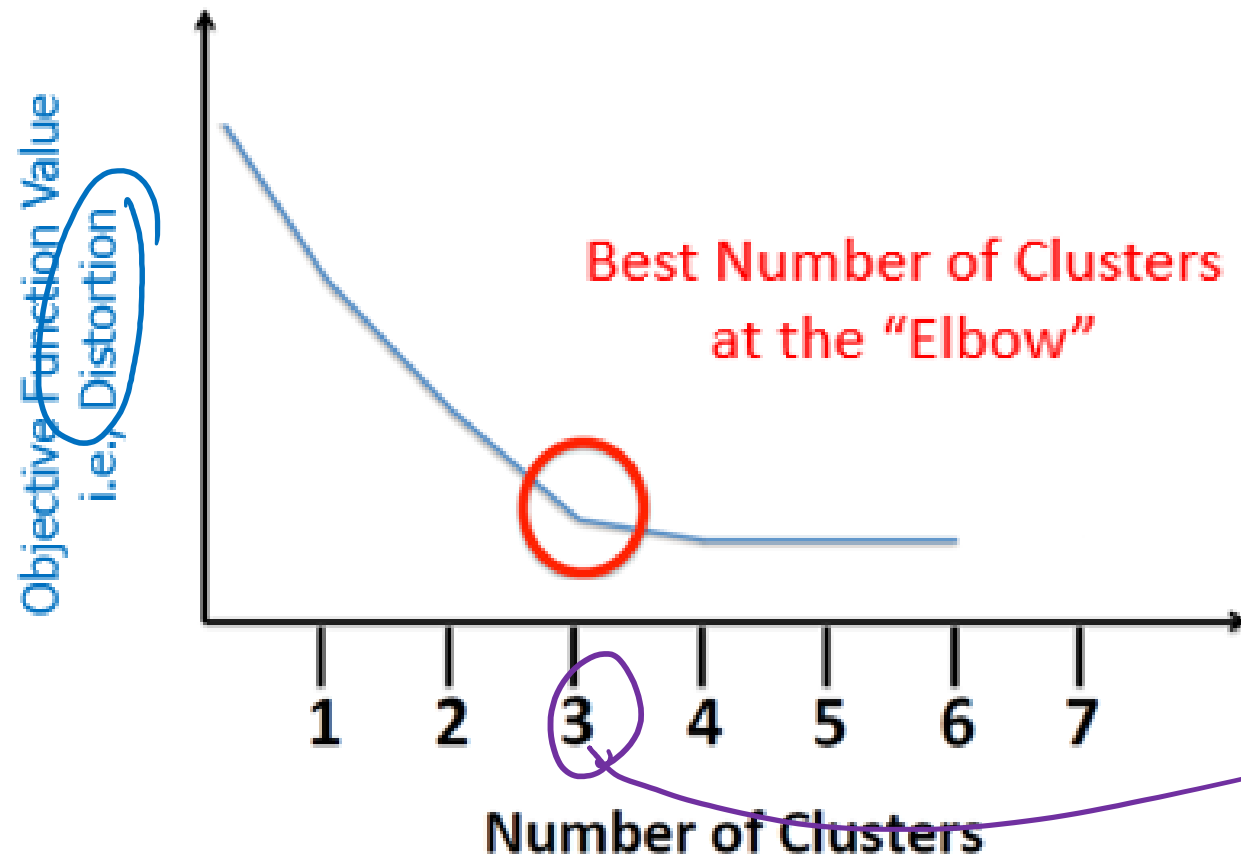
$$X = [x_1 \quad x_d]$$
$$Y = [y_1 \quad y_d]$$
$$\|X - Y\|_2^2 = (x_1 - y_1)^2 + (x_d - y_d)^2$$

- Assume computing distance between two instances is $O(d)$ where d is the dimensionality of the vectors.
- Reassigning clusters for all datapoints:
 - $O(kn)$ distance computations (when there is one feature)
 - $O(knd)$ (when there is d features)
- 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: $O(Iknd)$.

$$\begin{array}{c} O(d) \\ \downarrow \\ O(nd) \\ \downarrow \\ O(knd) \\ \downarrow \\ O(Iknd) \end{array}$$

How to Choose K?

Elbow method



①

$$Sum_1 = a_1 + b_1 + c_1$$

②

$$Sum_2 = a_2 + b_2 + c_2$$

③

$$Sum_3 = a_3 + b_3 + c_3$$

$$Distortion = Sum_1 + Sum_2 + Sum_3$$

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