CSE 6740: Computational Data Analysis

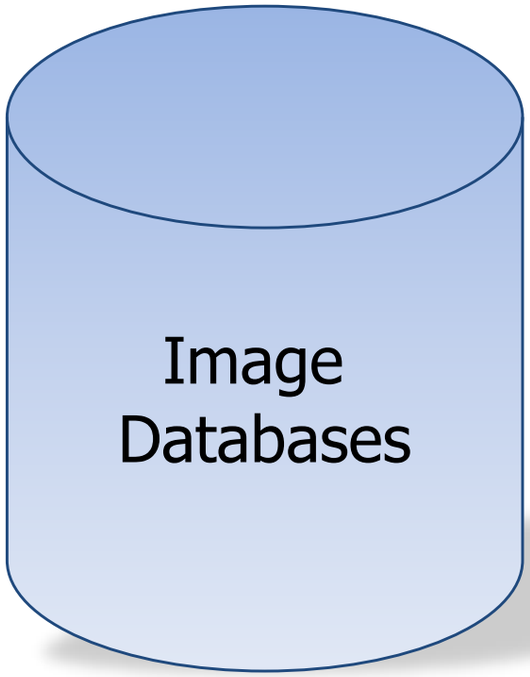
Spring 2026

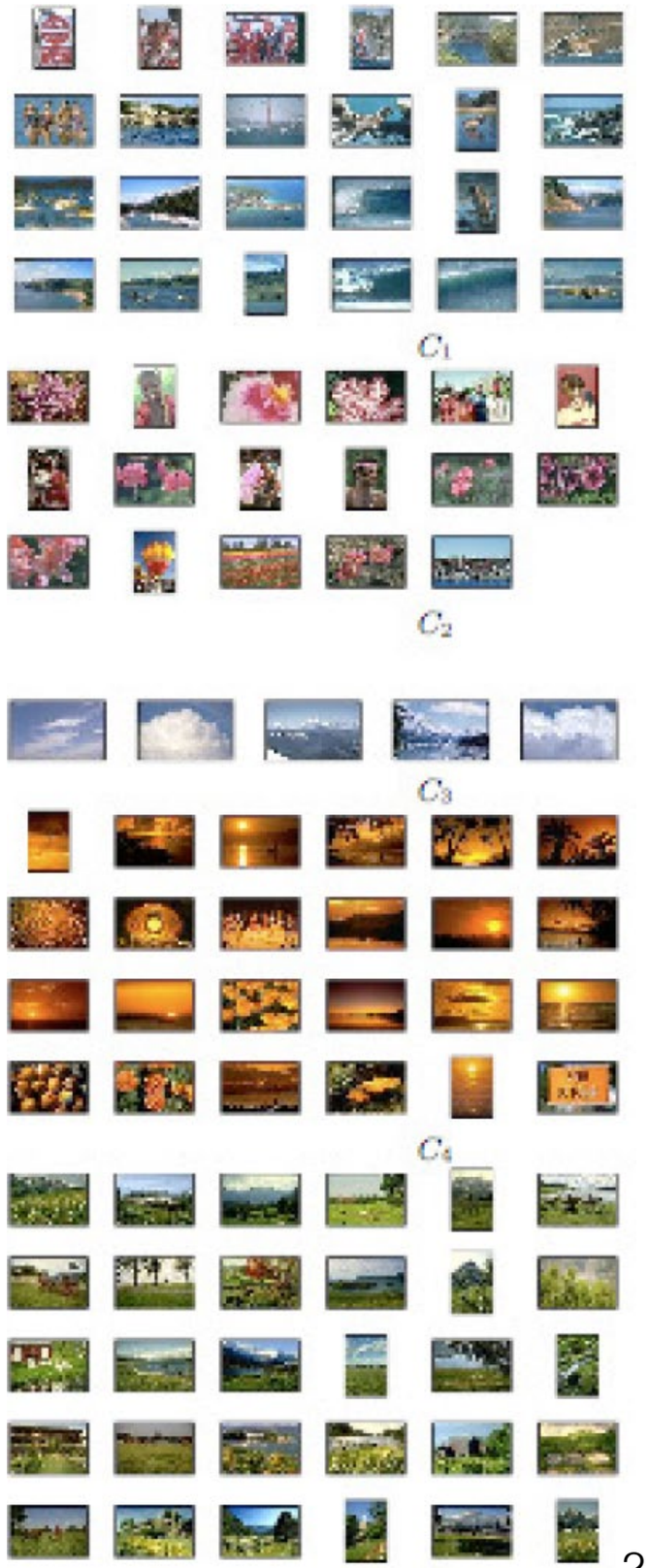
Clustering

Anqi Wu

01/15

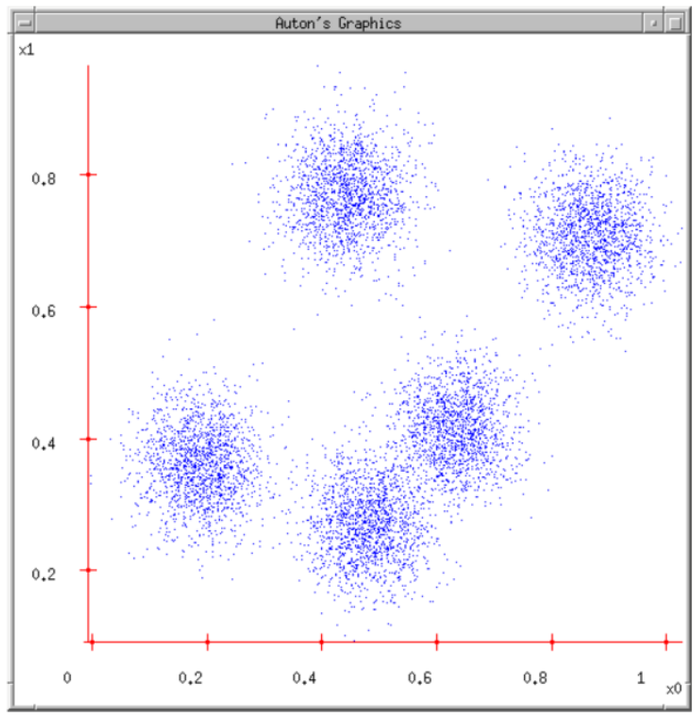
Clustering images





Goal of clustering:

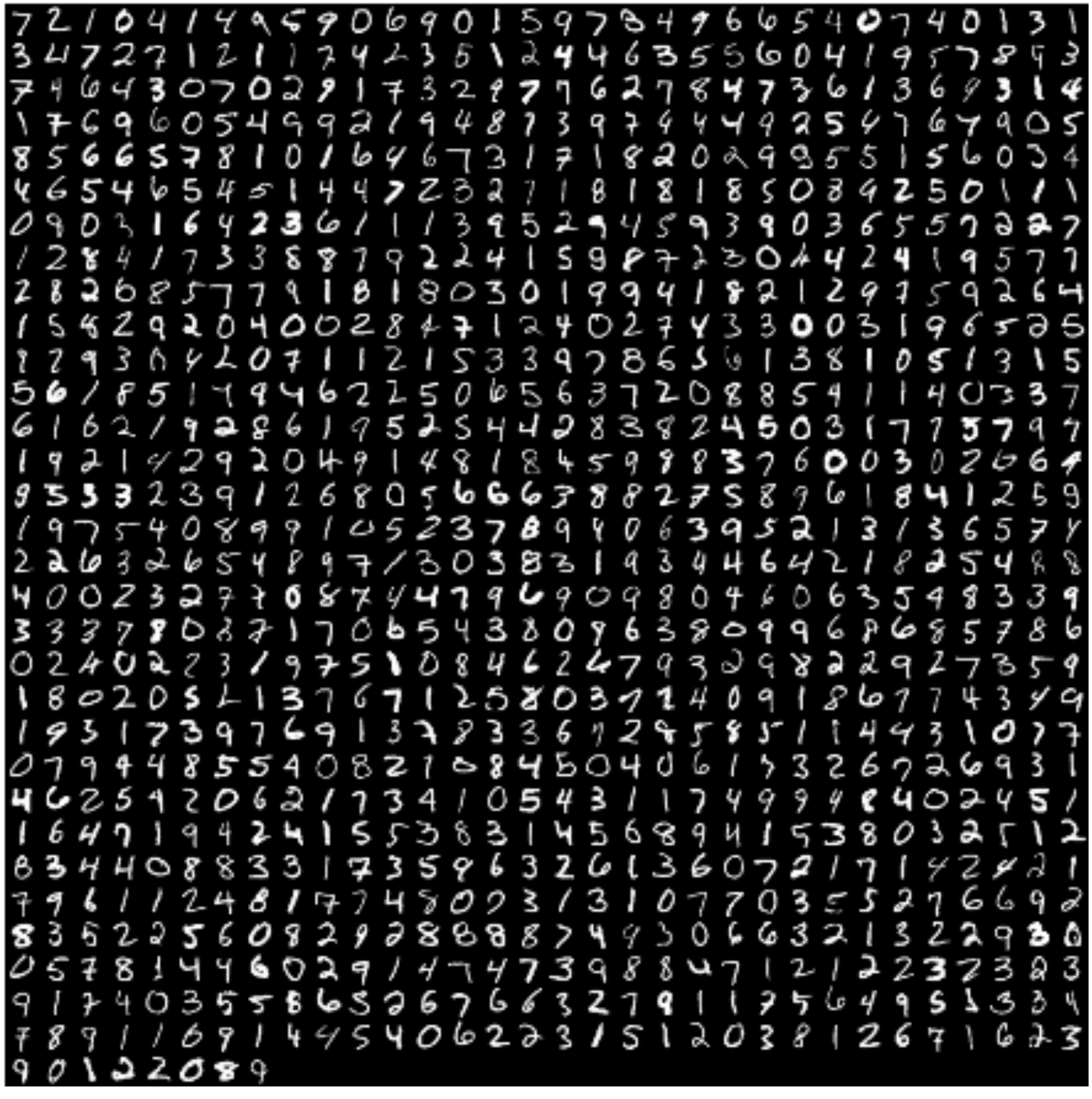
Divide object into groups, and objects within a group are more similar than those outside the group



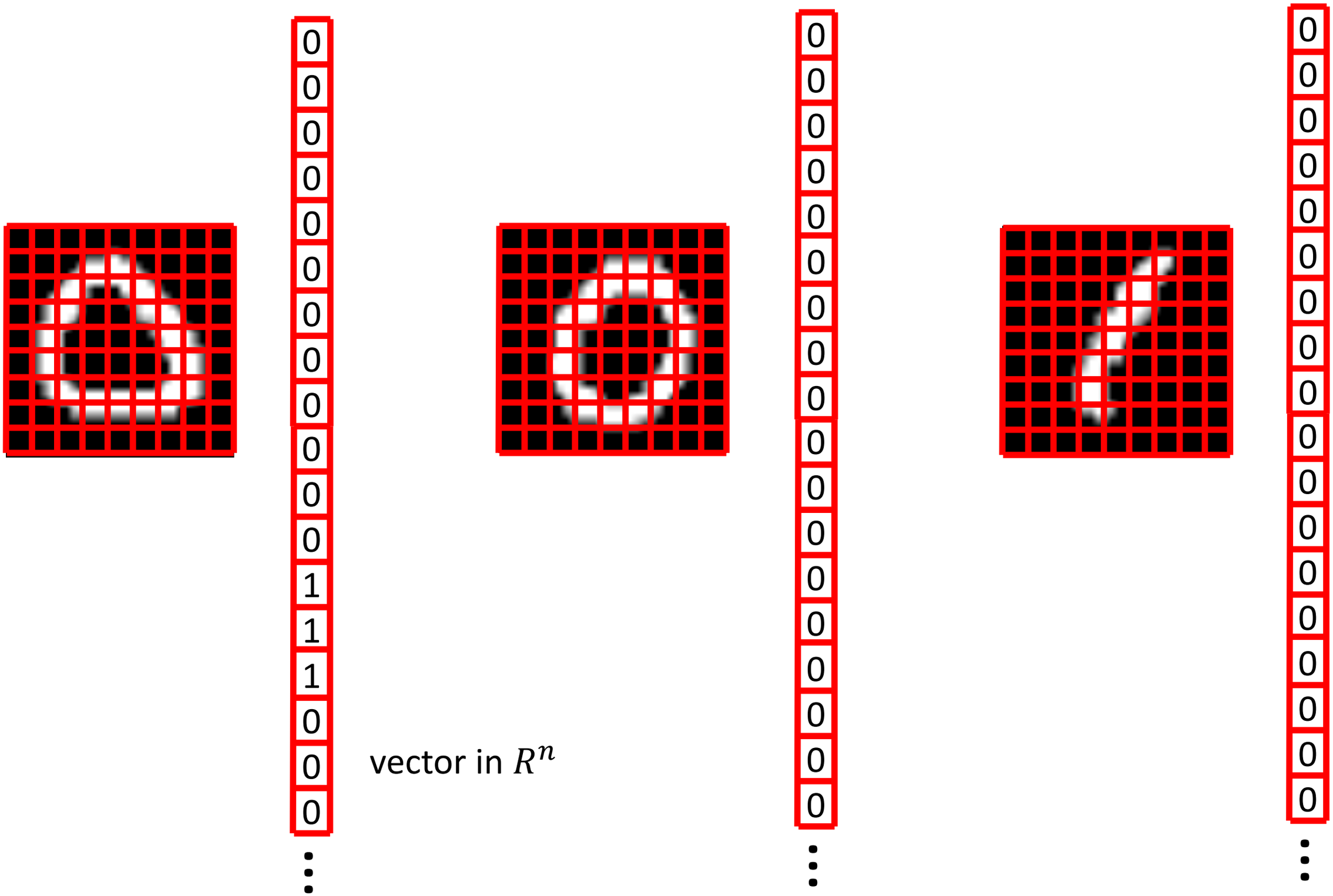
Cluster other things …

Piotr
Pyotr
Petros
Pietro
Pedro
Pierre
Piero
Peter
Peder
Peka
Peadar

Cluster handwritten digits



How to represent objects?



See demo kmeans_digit.m

Formal statement of clustering problem

- Given m data points, $\{x^1, x^2, \dots x^m\} \in R^n$
- Find k cluster centers, $\{c^1, c^2, \dots, c^k\} \in R^n$
- And assign each data point i to one cluster, $\pi(i) \in \{1, \dots, k\}$
- Such that the averaged square distances from each data point to its respective cluster center is small

$$\min_{c,\pi} \frac{1}{m} \sum_{i=1}^{m} \left\| x^i - c^{\pi(i)} \right\|^2$$

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)

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

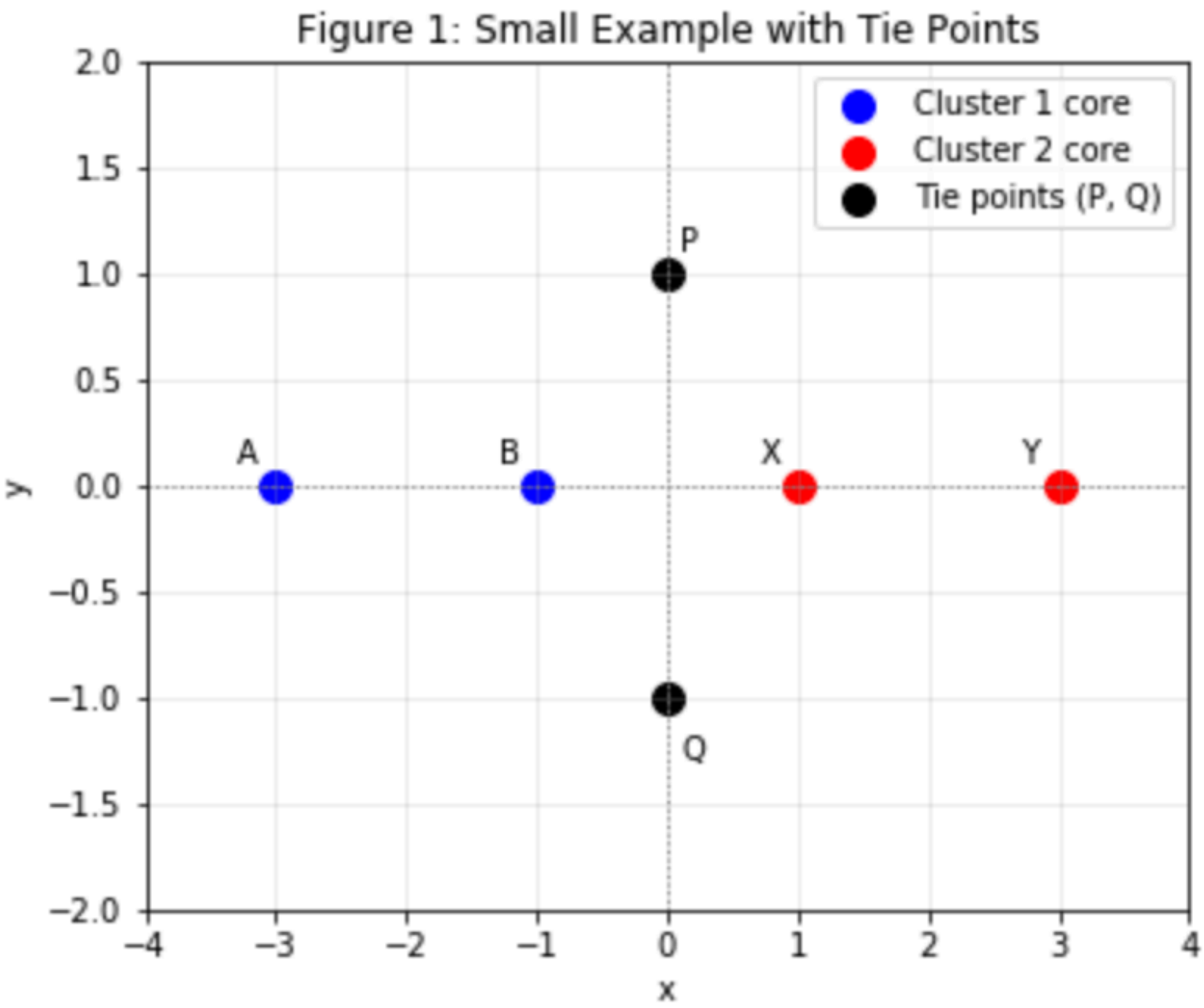
  - Adjust the cluster centers (center adjustment)

$$c^j = \frac{1}{|\{i : \pi(i) = j\}|} \sum_{i:\pi(i)=j} x^i$$

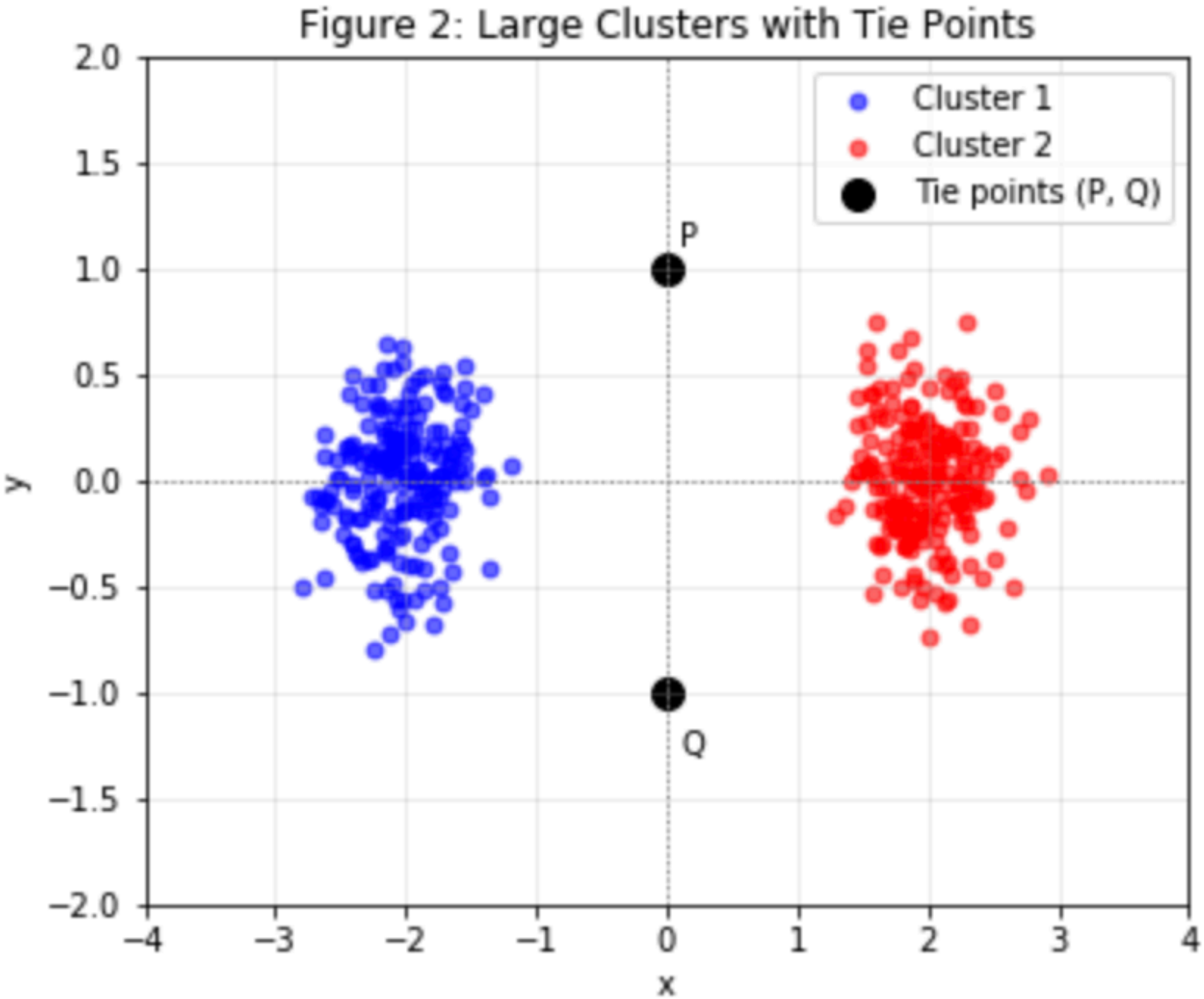
- While any cluster center has been changed

- Note: Tie points on cluster boundaries might flip between clusters without shifting centers.

K-means algorithm

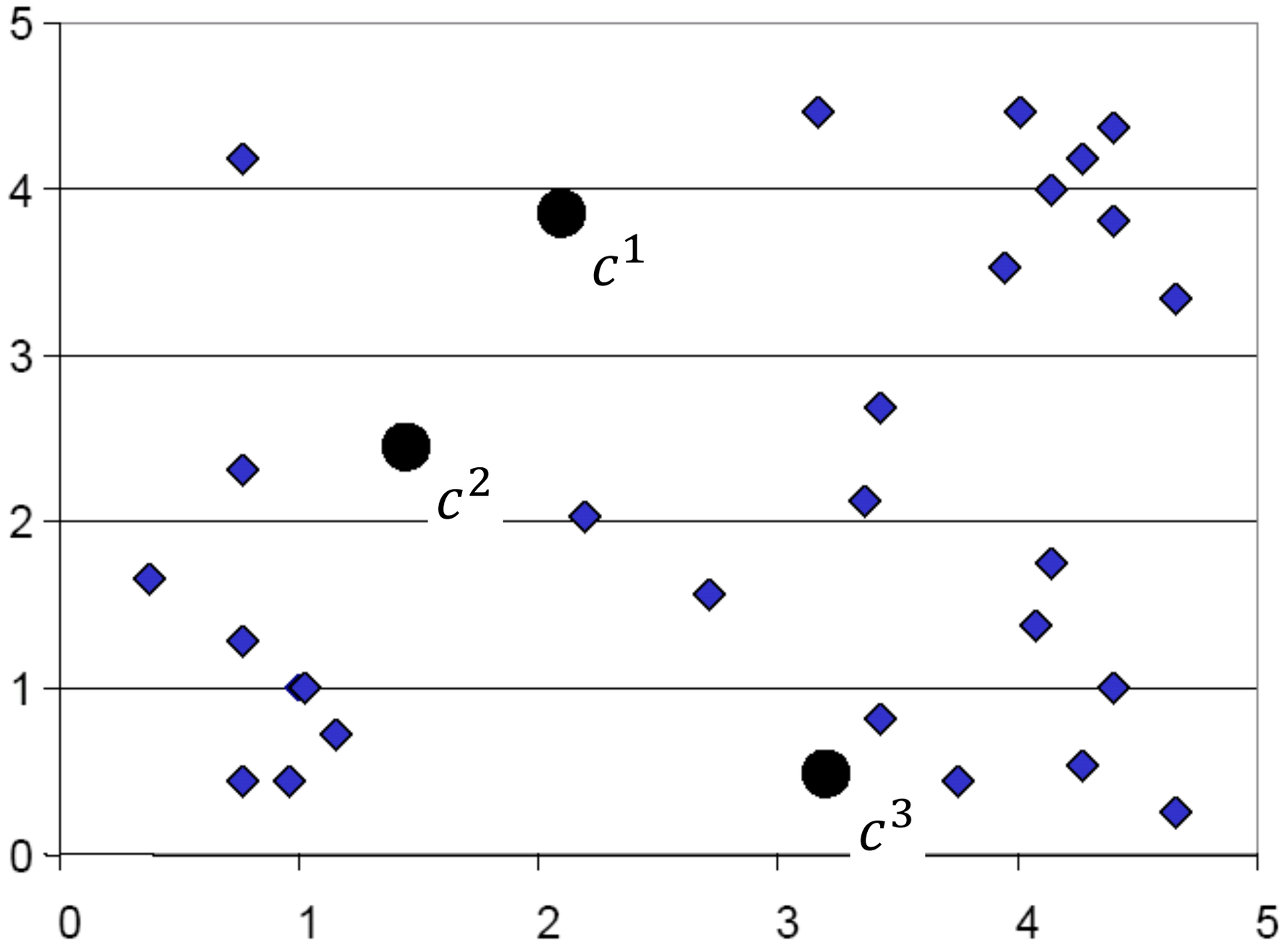


K-means algorithm

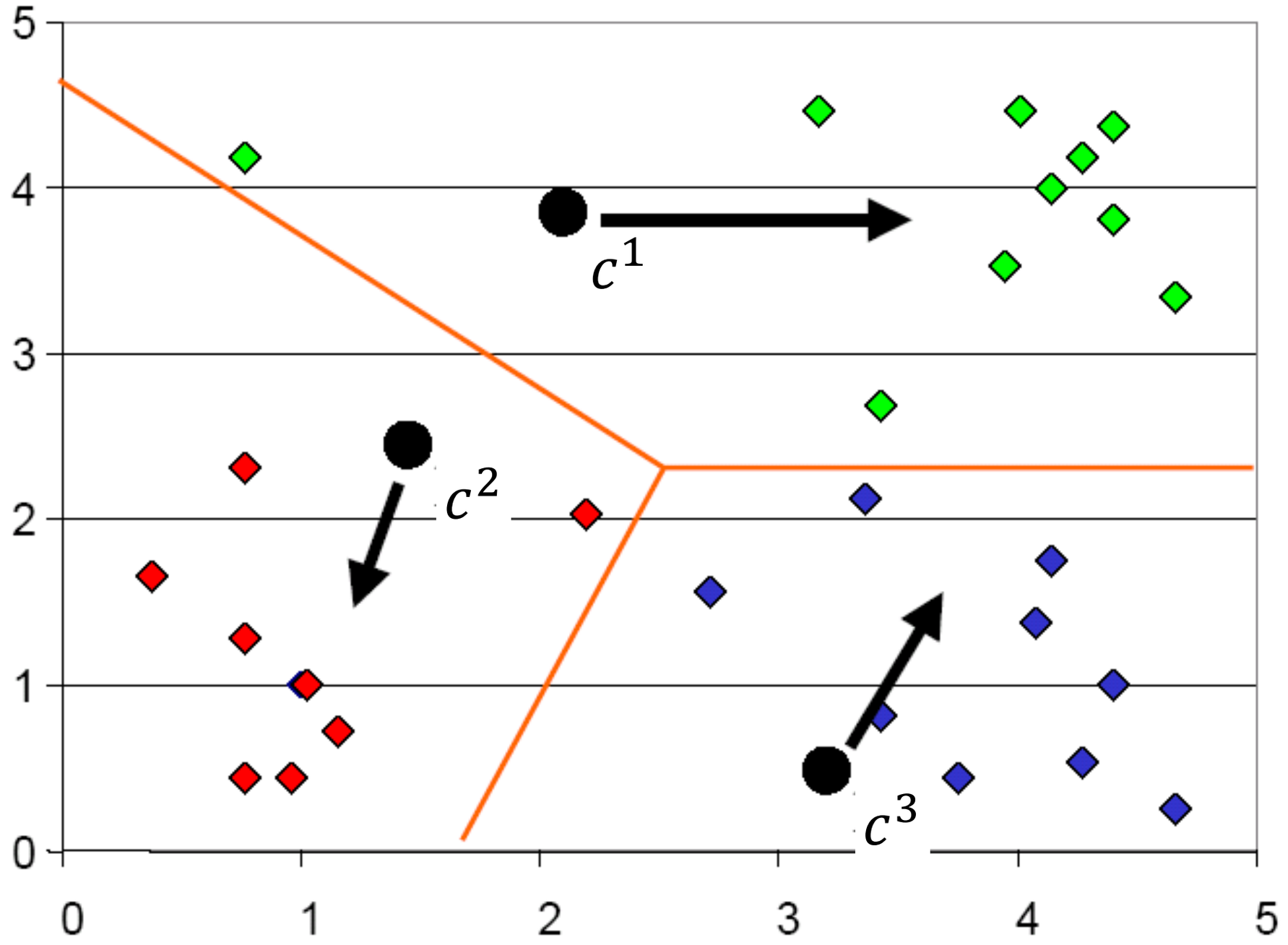


See demo kmeans_animation.m

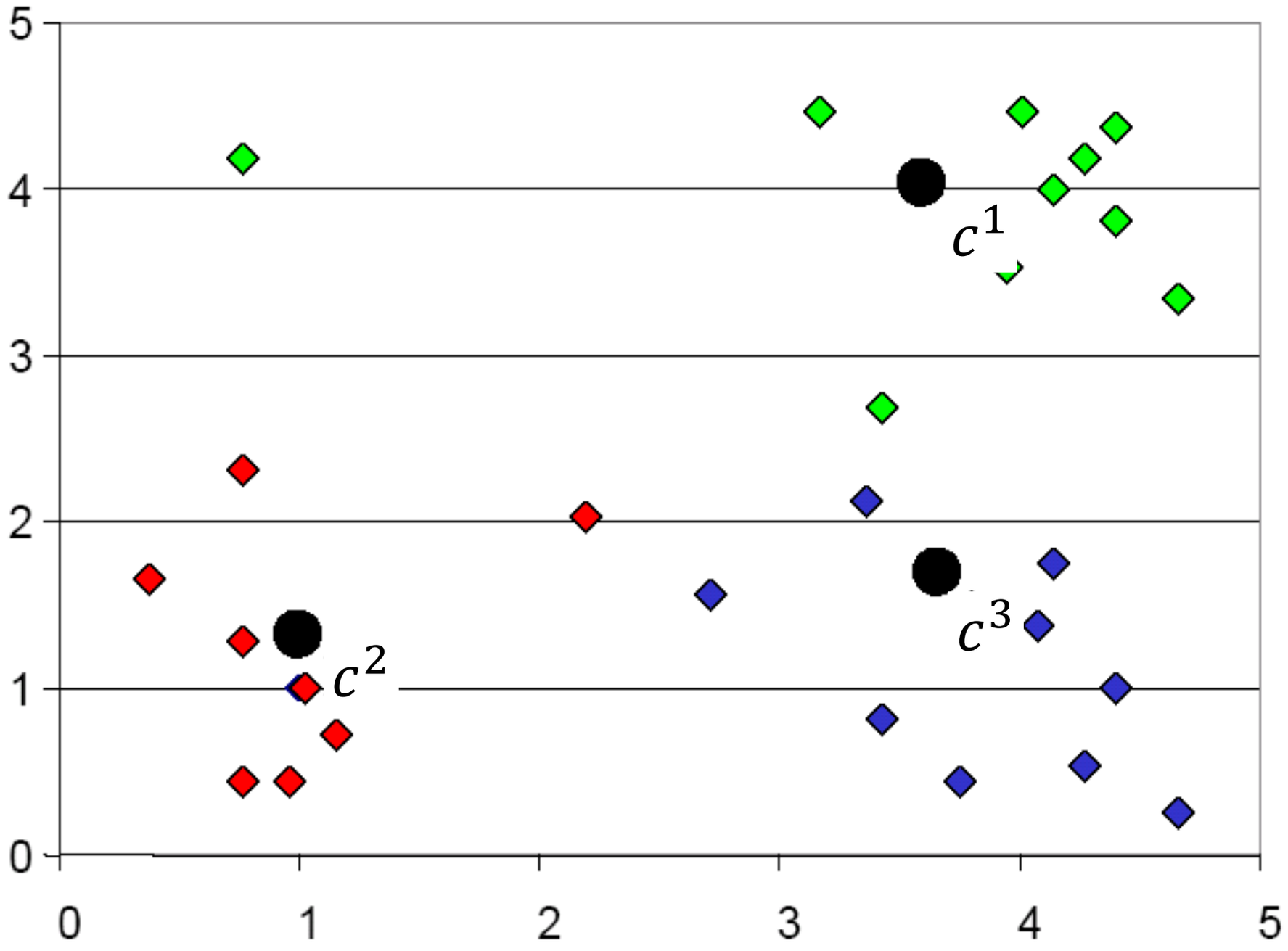
K-means: step 1



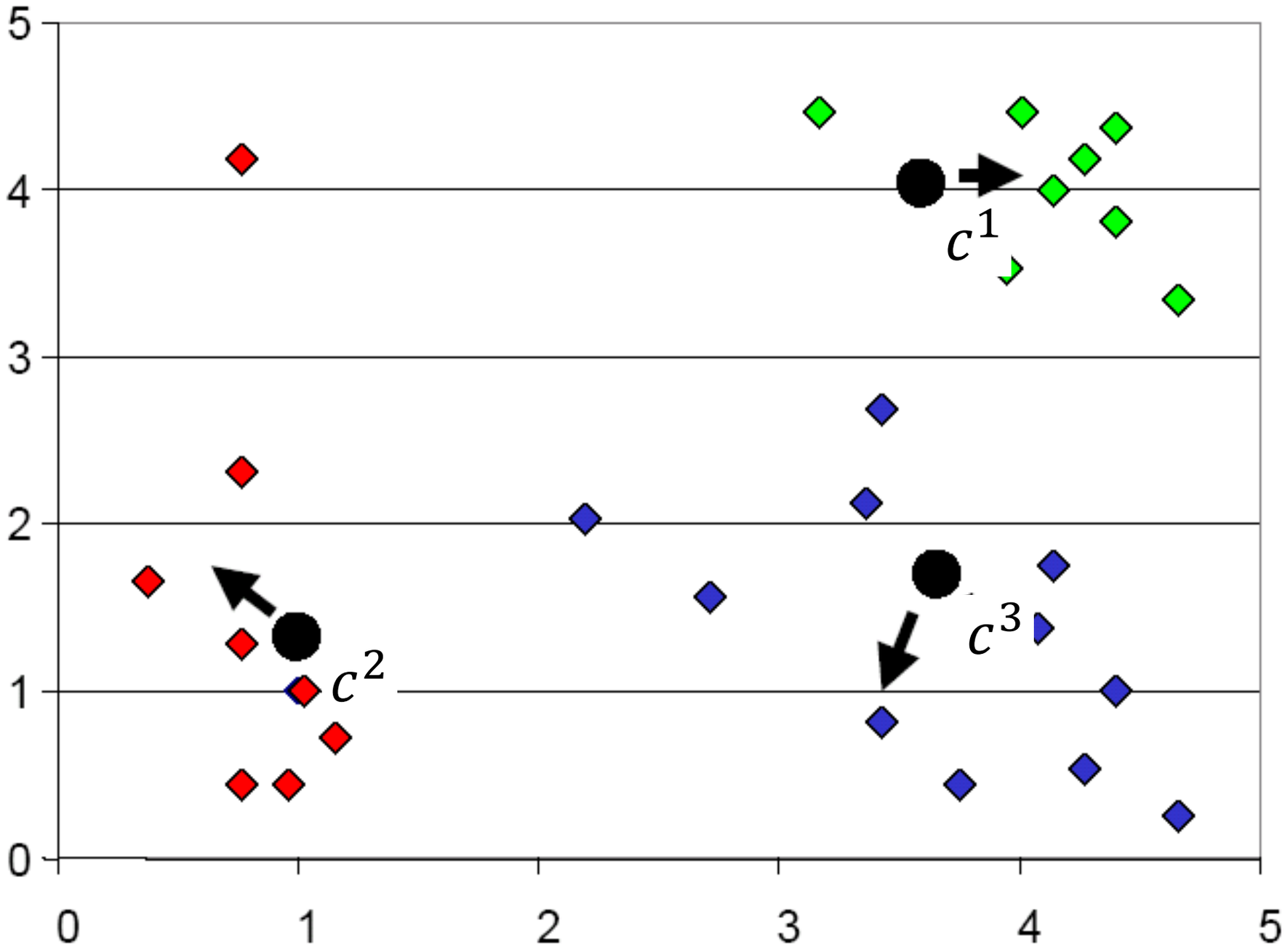
K-means: step 2



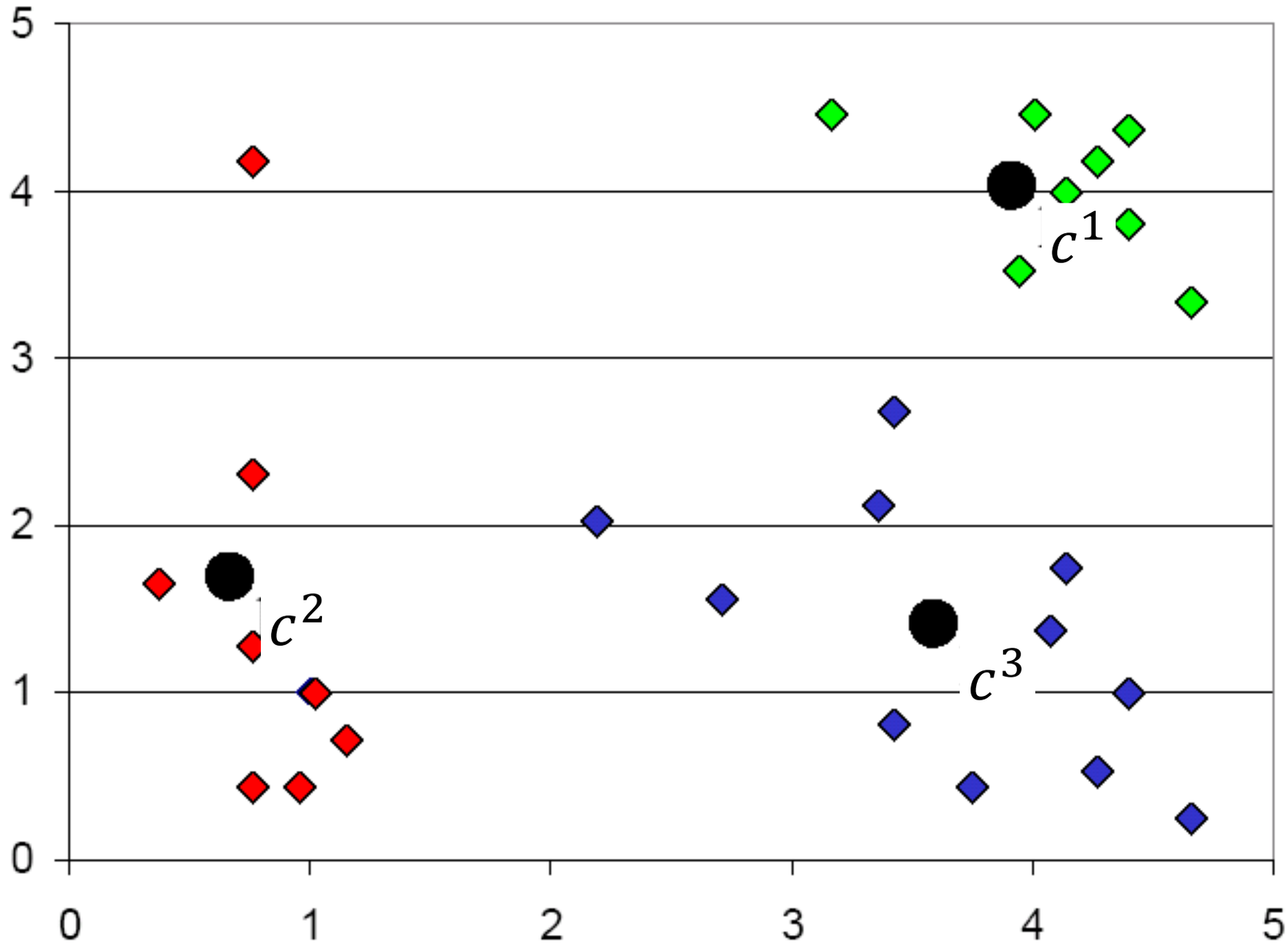
K-means: step 3



K-means: step 4



K-means: step 5

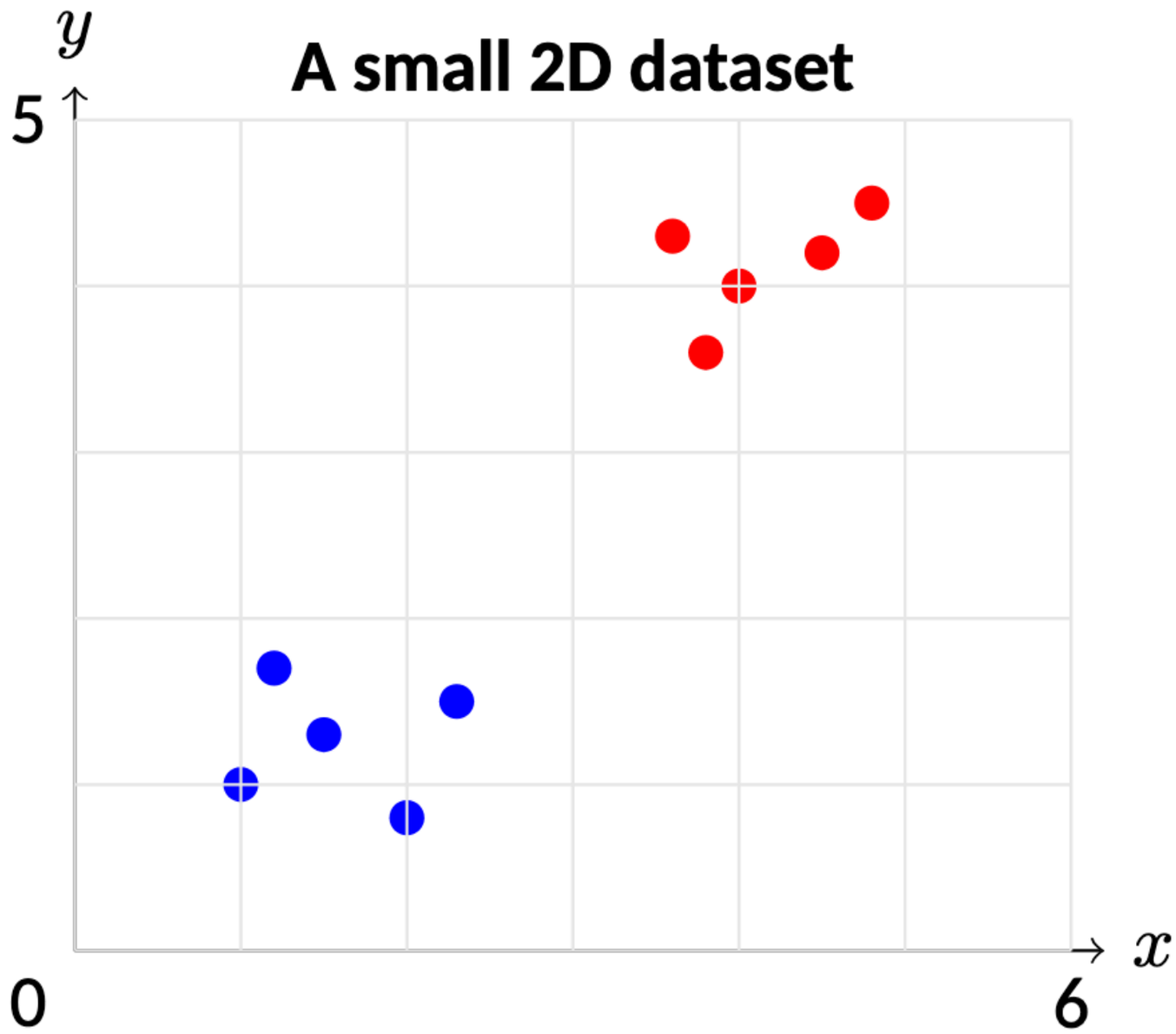


Questions

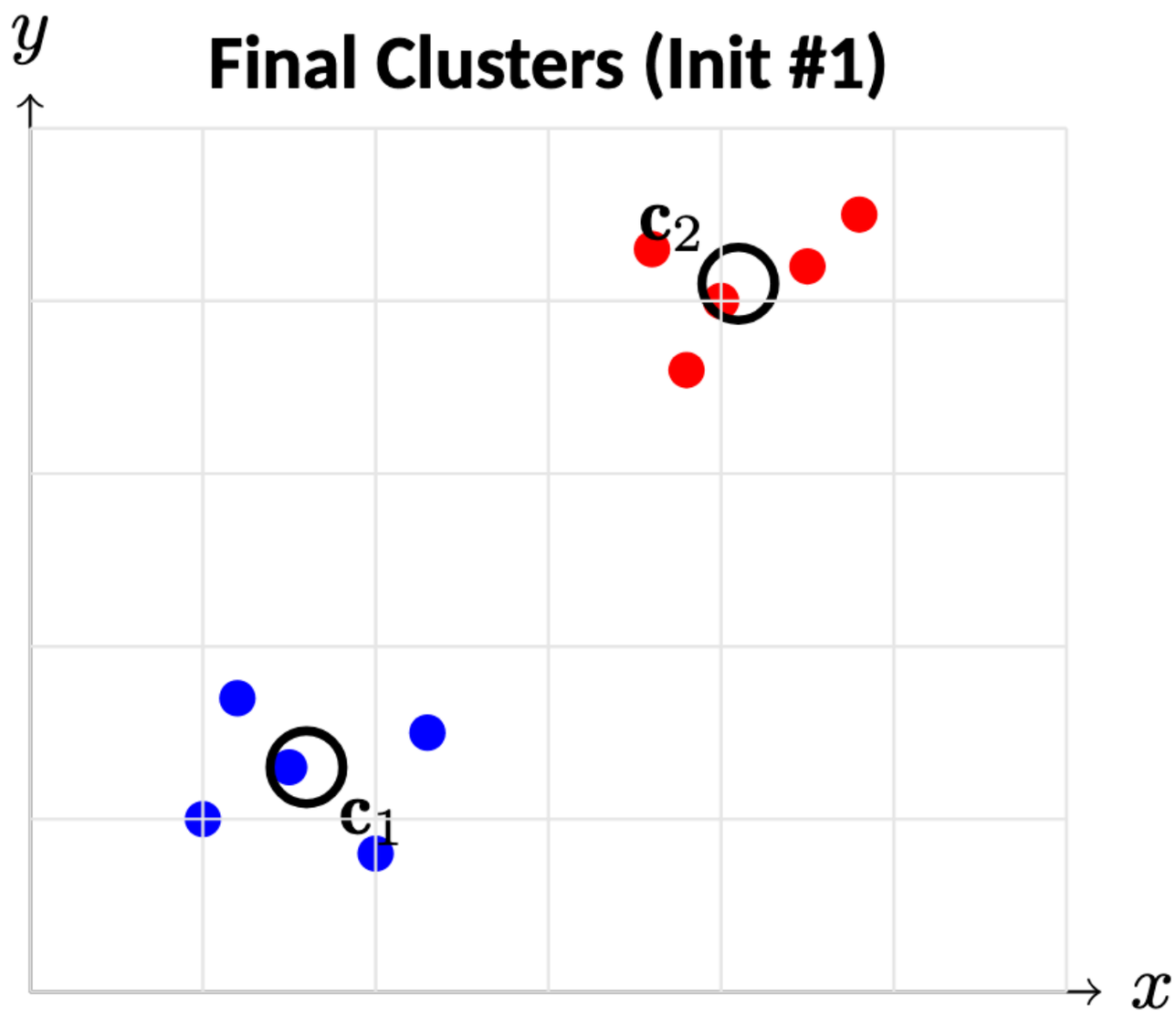
- Will different initialization lead to different results?
  - Yes
  - No
  - Sometimes

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

Questions

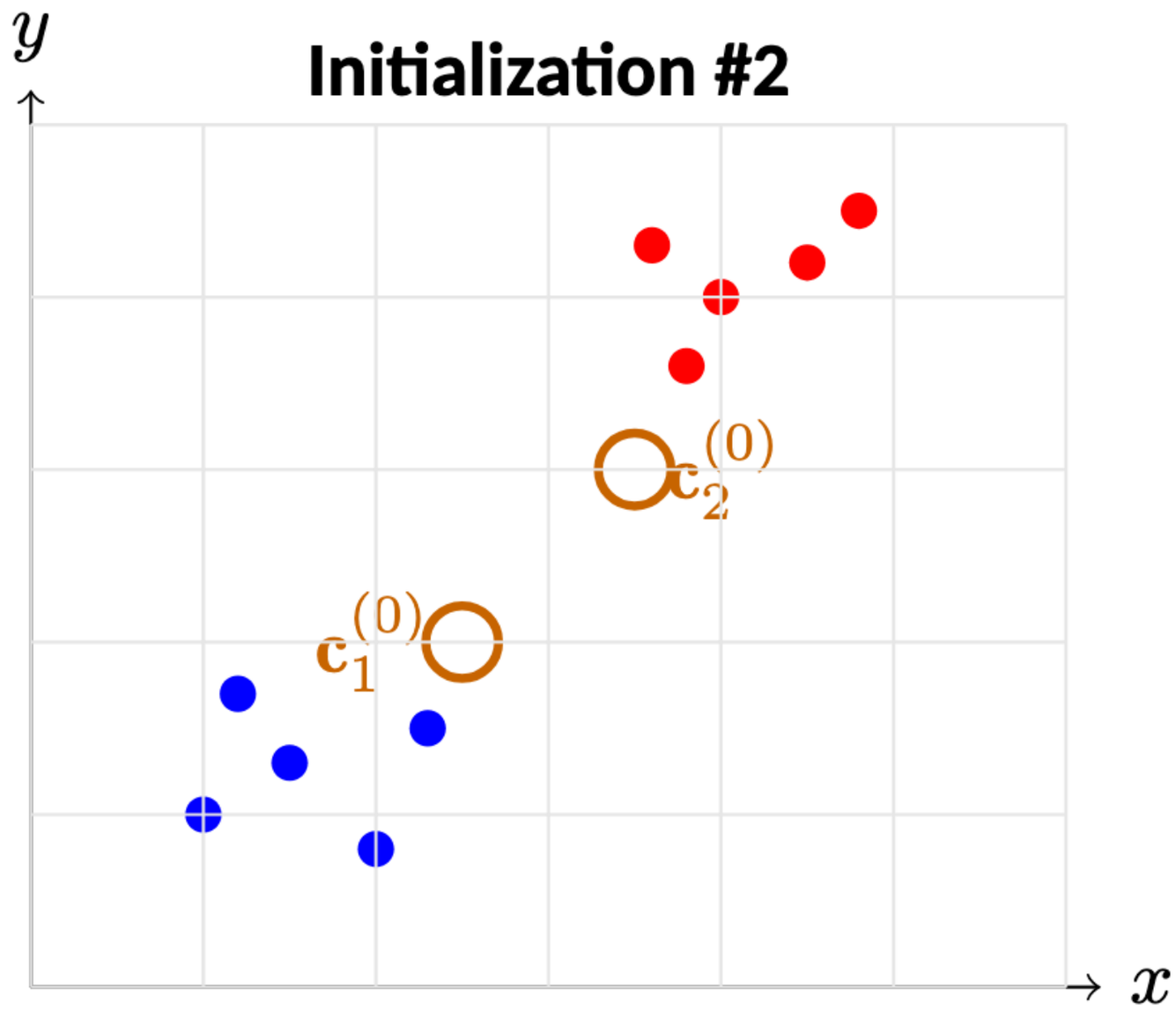


Questions



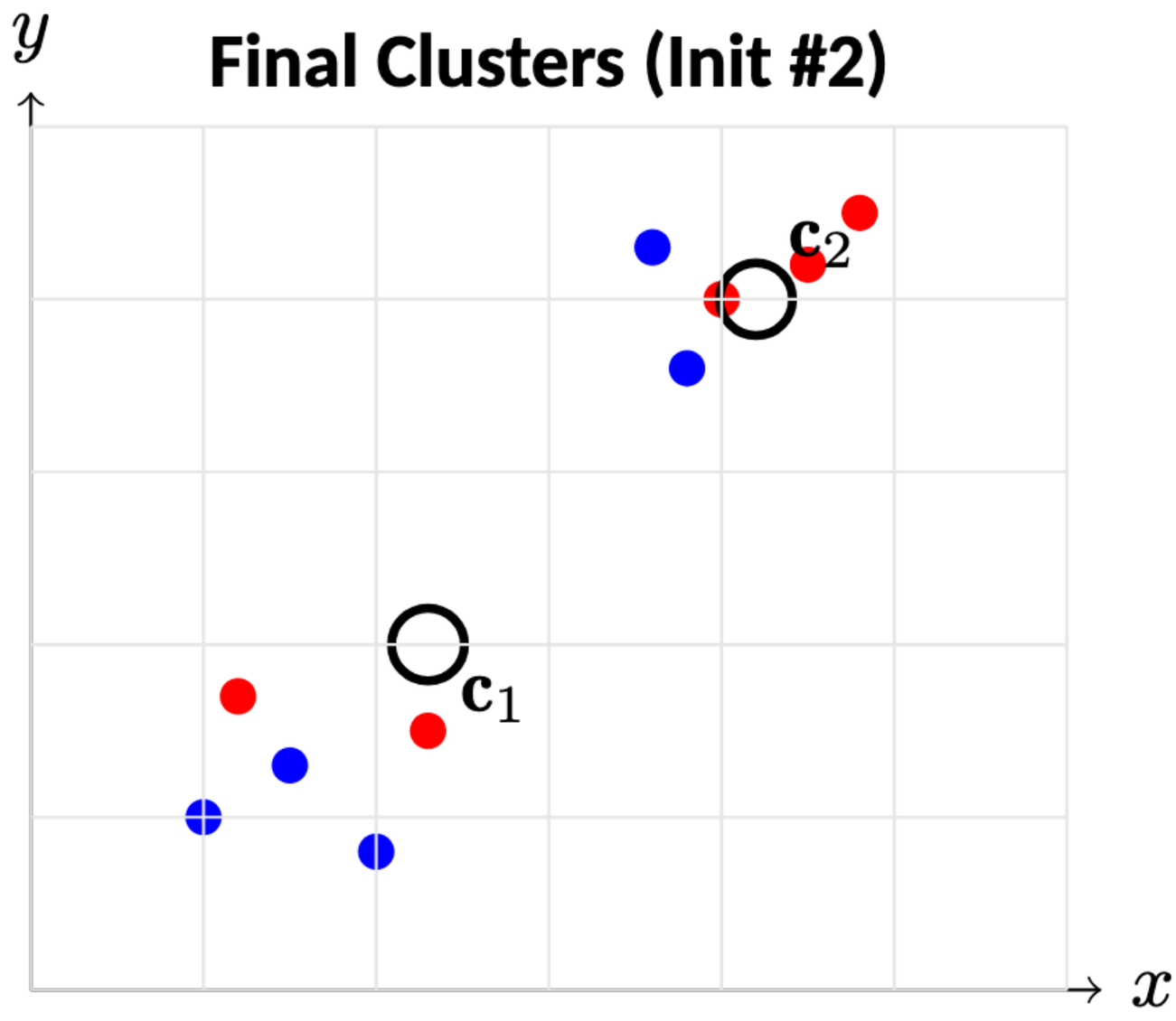
K-Means converges to one partition of the data.

Questions



A different, more central initial placement for centroids.

Questions



A different final grouping emerges, reflecting a different local optimum.

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

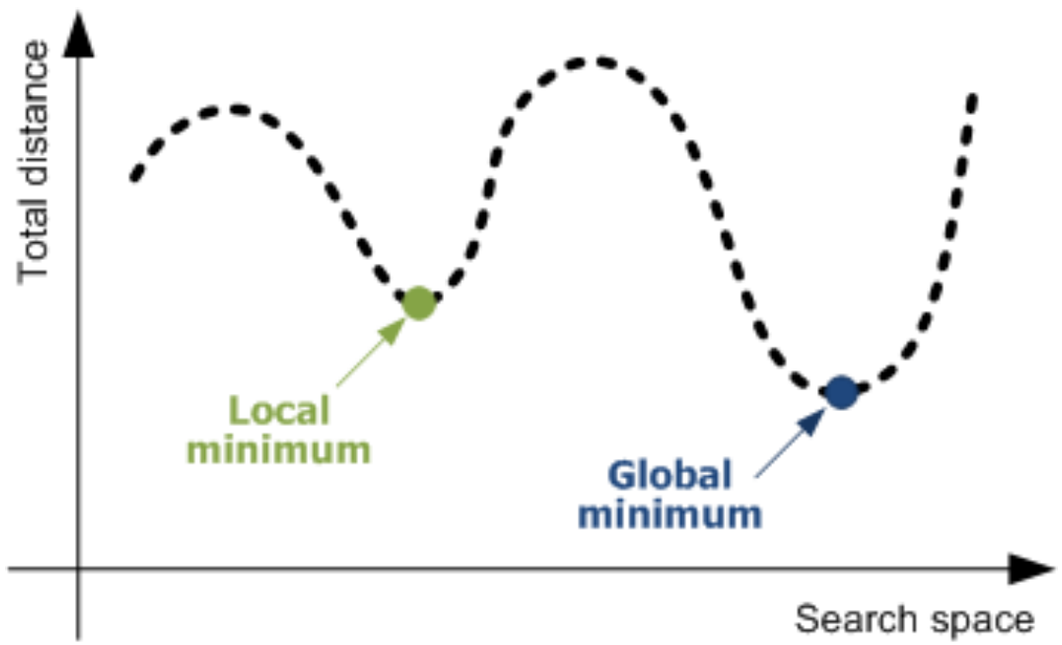
Clustering is NP-hard in general

- Find k cluster centers, $\{c^1, c^2, \dots, c^k\} \in R^n$, and assign each data point i to one cluster, $\pi(i) \in \{1, \dots, k\}$, to minimize

$$\min_{c,\pi} \frac{1}{m} \sum_{i=1}^{m} \left\| x^i - c^{\pi(i)} \right\|^2$$

NP-hard!

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



Convergence of kmeans algorithm

- Will kmeans objective oscillate?

$$\frac{1}{m}\sum_{i=1}^{m}\left\|x^i - c^{\pi(i)}\right\|^2$$

- The minimum value of the objective is finite

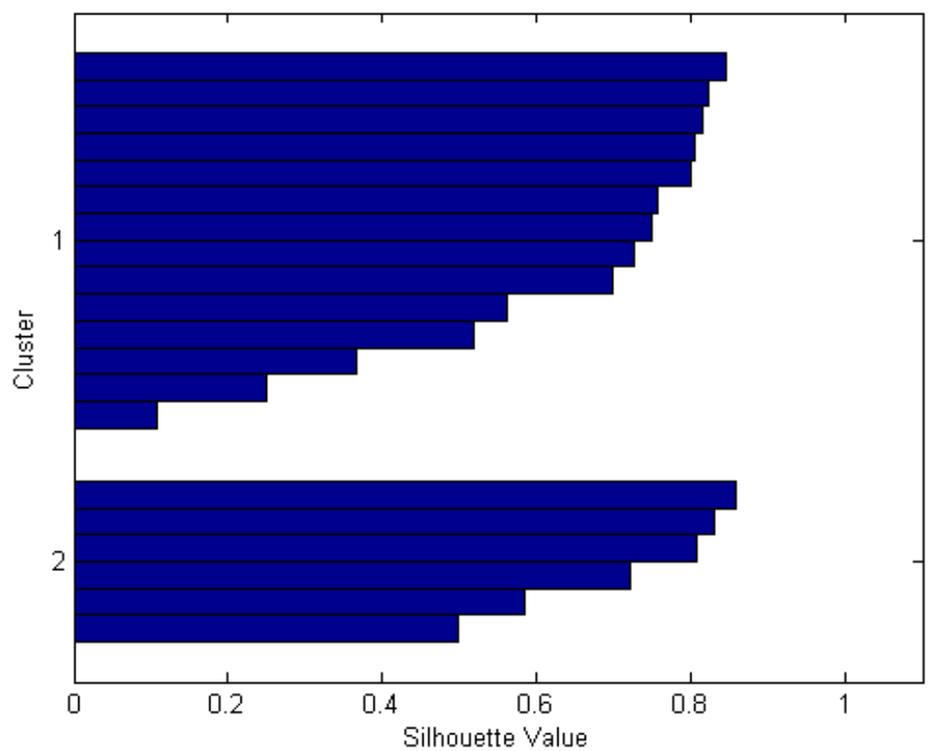
- Each iteration of kmeans algorithm decrease the objective
  - Cluster assignment step decreases objective
    - $\pi(i) = argmin_{j=1,\ldots,k}\left\|x^i - c^j\right\|^2$ for each data point i
  - Center adjustment step decreases objective
    - $c^j = \frac{1}{|\{i:\pi(i)=j\}|}\sum_{i:\pi(i)=j} x^i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

How many clusters?

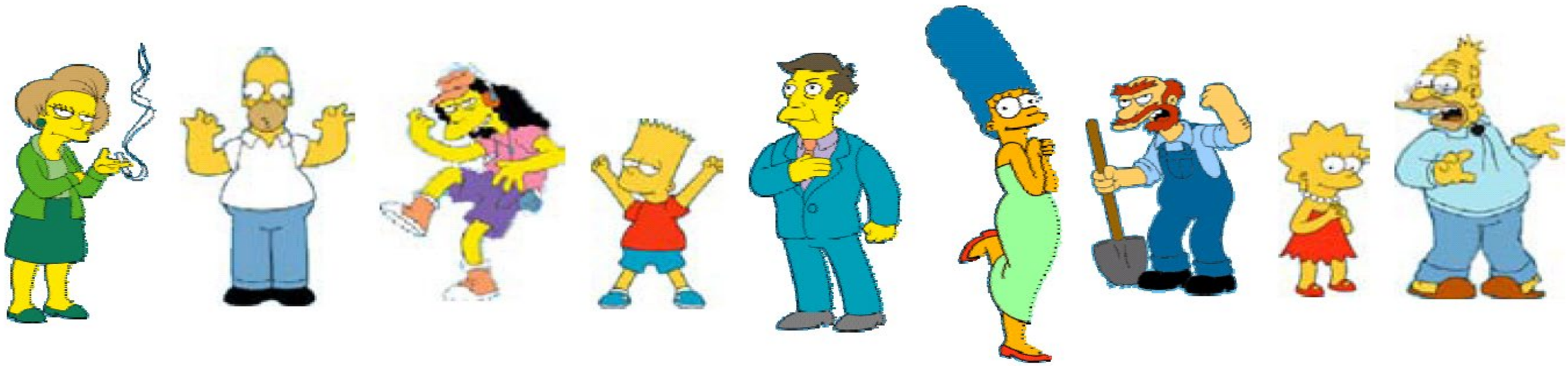
- Fixed a-priori? Data-driven approach?

- Silhouette value: $S_i = \frac{b_i - a_i}{\max(a_i, b_i)}$ (one heuristic)
  - a_i: the average distance from the ith point to the other points in the same cluster as i,
  - b_i: the minimum average distance from the ith point to points in a different cluster, minimized over clusters.

- No gold standard method
  - Often determine by trial-and-error

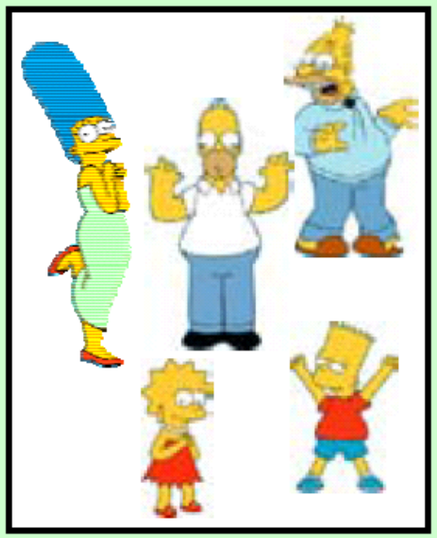


Are these everything about clustering?



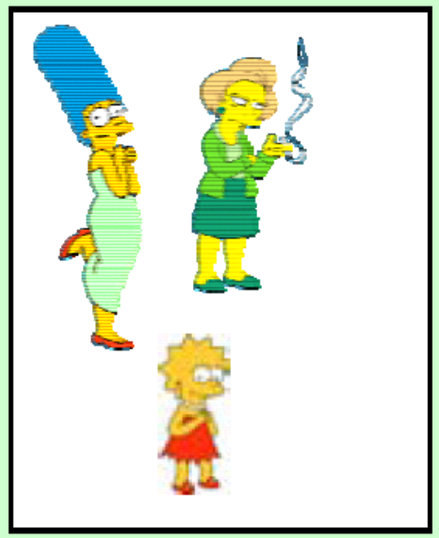
What is consider similar/dissimilar?

Clustering is subjective



Simpson's Family

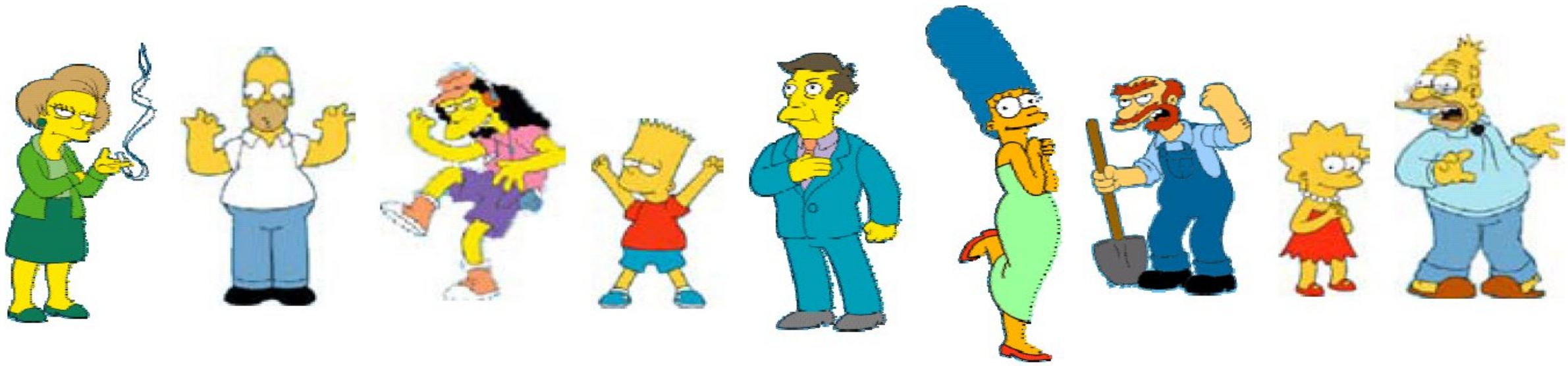
School Employees



Females

Males

Objects in real life



Family: 0/1/2 …

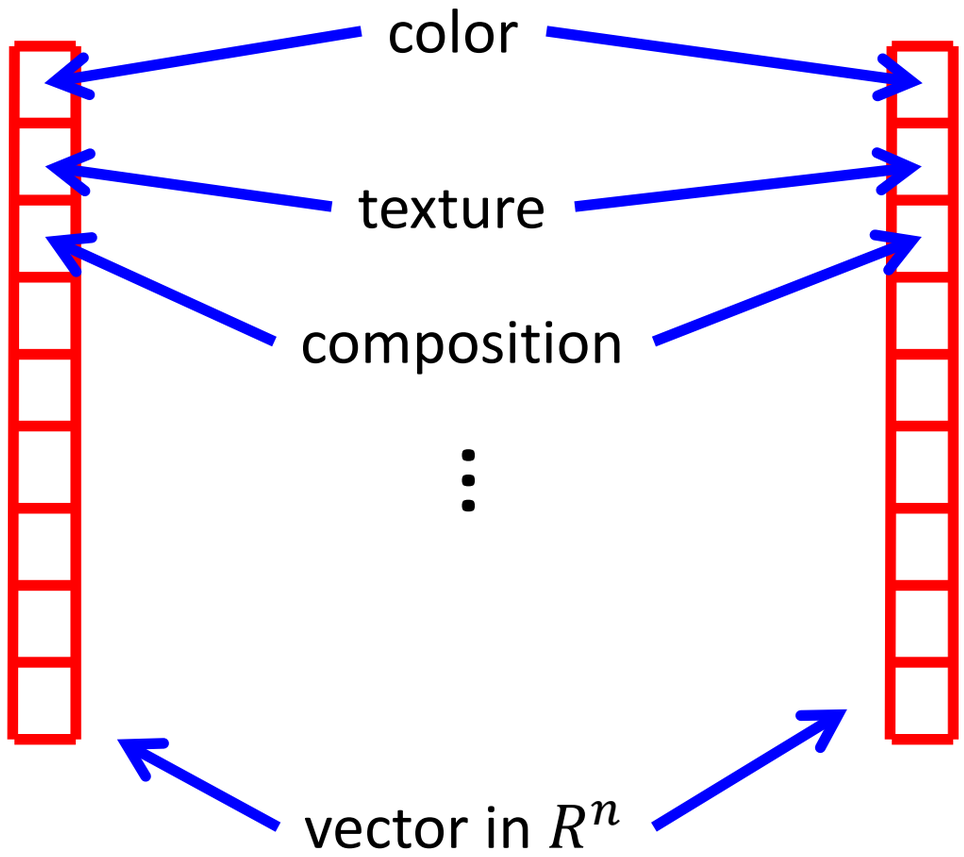
Sex: 0/1 …

Work place: 0/1/2/3 …

…

vector in R^n

Images of different sizes



You pick your similarity/dissimilarity

What similarity/dissimilarity 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"*

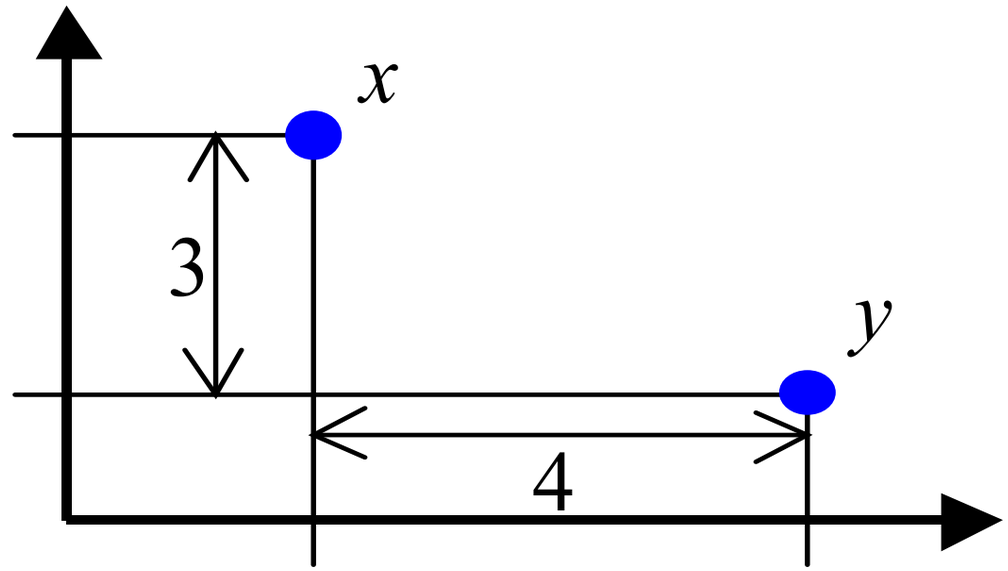
Distance functions for vectors

- Suppose two data points, both in R^n
  - $x = (x_1, x_2, \ldots, x_n)^\top$
  - $y = (y_1, y_2, \ldots, y_n)^\top$

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

- Minkowski distance: $d(x, y) = \sqrt[p]{\sum_{i=1}^{n}(x_i - y_i)^p}$
  - Euclidian distance: $p = 2$
  - Manhattan distance: $p = 1, d(x, y) = \sum_{i=1}^{n}|x_i - y_i|$
  - "inf"-distance: $p = \infty, d(x, y) = \max_{i=1}^{n}|x_i - y_i|$

Distance example



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

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)

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

Generalized K-means algorithm

- Initialize k cluster centers, $\{c^1, c^2, \ldots, c^k\}$, randomly
- Do
  - Decide the cluster memberships of each data point, x^i, by assigning it to the nearest cluster center

$$\pi(i) = argmin_{j=1,\ldots,k}\, d(x^i, c^j)$$

  - Adjust the cluster centers

$$c^j = argmin_{v \in R^n} \sum_{i:\pi(i)=j} d(x^i, v)$$

squared Eclidian distance:

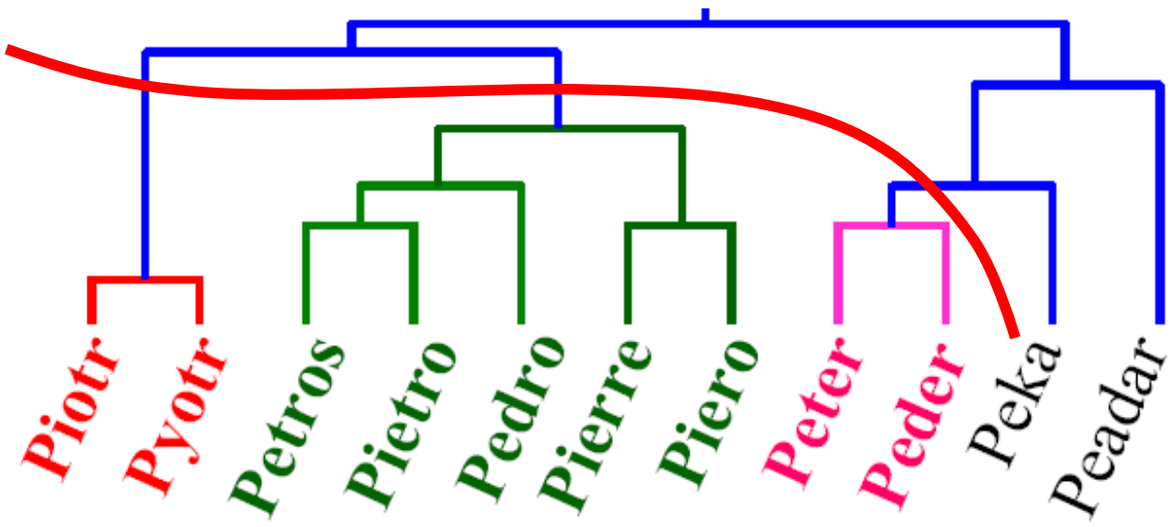
$$c^j = \frac{1}{\#\{\pi(i) = j\}} \sum_{i:\pi(i)=j} x^i$$

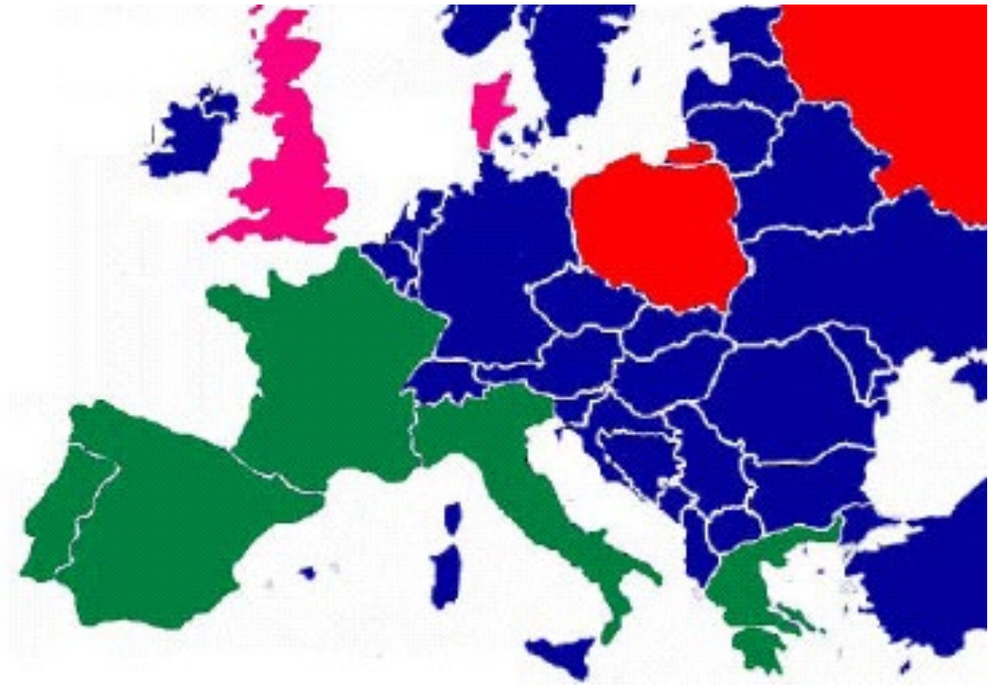
- While any cluster center has been changed

Hierarchical clustering

- Organize data in a hierarchical fashion (dendrogram)

- Clustering obtained by cutting the dendrogram at a desired level: each connected component forms a cluster.





- How to do it?

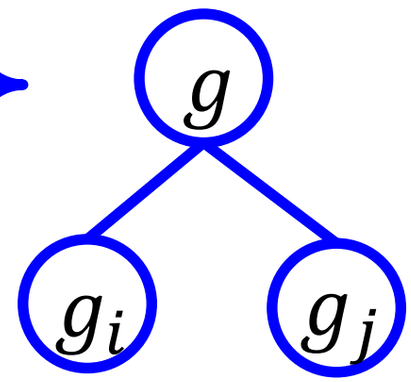
Bottom up hierarchical clustering

- Assign each data point to its own cluster, $g_1 = \{x_1\}, g_2 = \{x_2\}, \dots, g_m = \{x_m\}$, and let $G = \{g_1, g_2, \dots, g_m\}$

$$D(g_i, g_j) = \min_{x \in g_i, y \in g_j} d(x, y)$$

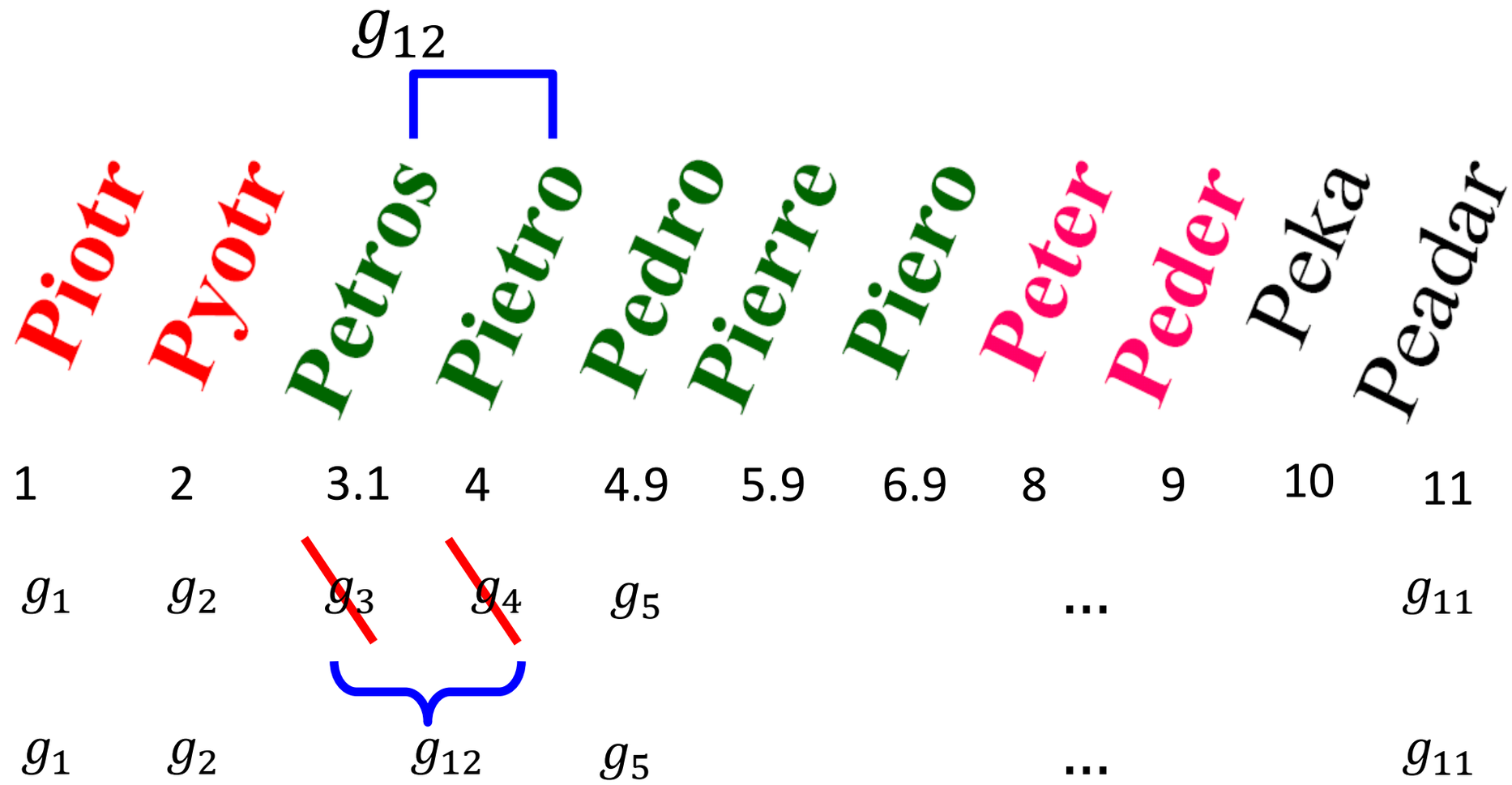
- Do
  - Find two clusters to merge: $i, j = argmin_{1 \le i,j \le |G|} \; D(g_i, g_j)$
  - Merge the two clusters to a new cluster: $g \leftarrow g_i \cup g_j$
  - Remove the merged clusters: $G \leftarrow G \setminus g_i, G \leftarrow G \setminus g_j$
  - Add the new cluster: $G \leftarrow G \cup \{g\}$

keep track of relations

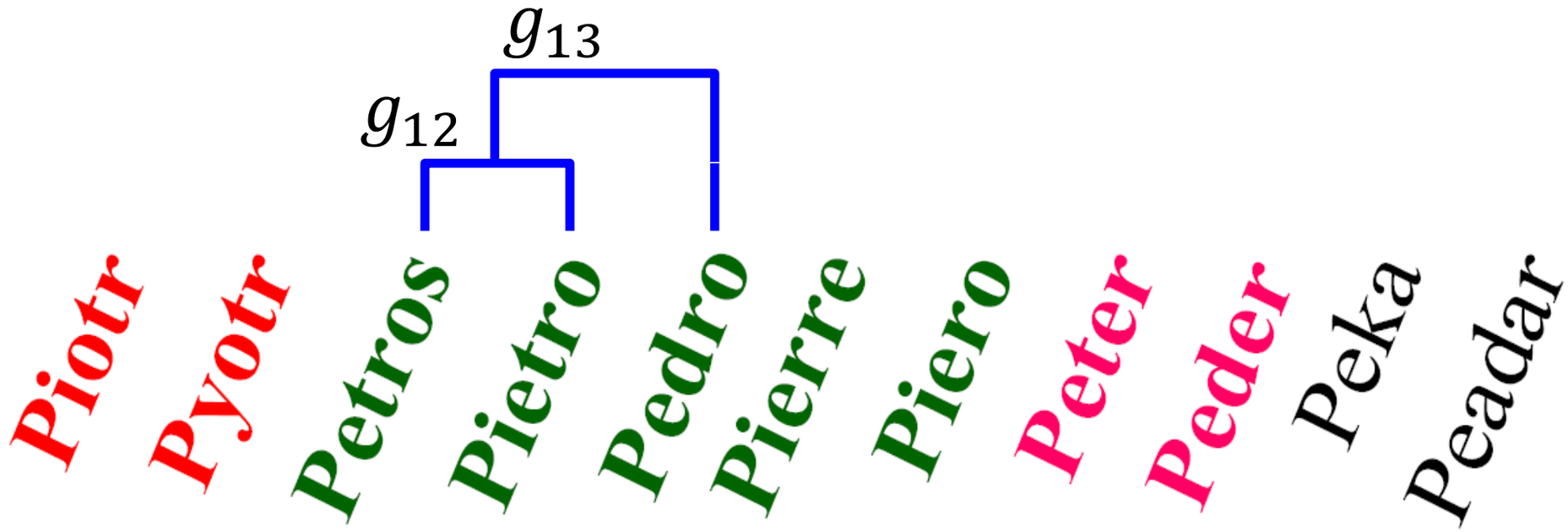


- While $|G| > 1$

Hierarchical clustering: step-2



Hierarchical clustering: step-3



1 2 3.1 4 4.9 5.9 6.9 8 9 10 11

$G = \{ \; g_1 \quad g_2 \quad g_3 \quad g_4 \quad g_5 \quad \ldots \quad g_{11} \; \}$

$G = \{ \; g_1 \quad g_2 \quad g_{12} \quad g_5 \quad \ldots \quad g_{11} \; \}$

$G = \{ \; g_1 \quad g_2 \quad g_{13} \quad \ldots \quad g_{11} \; \}$

Kmeans vs Hierarchal clustering

	Kmeans	HC
Number of clusters	If there is a specific number of clusters in the dataset, but the group they belong to is unknown	It is easier to determine the number of clusters by hierarchical clustering's dendrogram
Cluster result	unstructured	more interpretable and informative
Time complexity	$O(n \times k \times t)$	$O(n^3)$
Space complexity	$O(n(d + k))$	$O(n^2)$
With a large number of data points	compute faster	compute slower