Data Mining

INFS 4203/7203

Miao Xu

miao.xu@uq.edu.au

The University of Queensland, 2020 Semester 2

- Assignment 2 has been released, one-trial due Sep. 13th, 23:59

  (please pay attention to the academic integrity)

- Mid-term will be on Sep. 18th in class for one hour

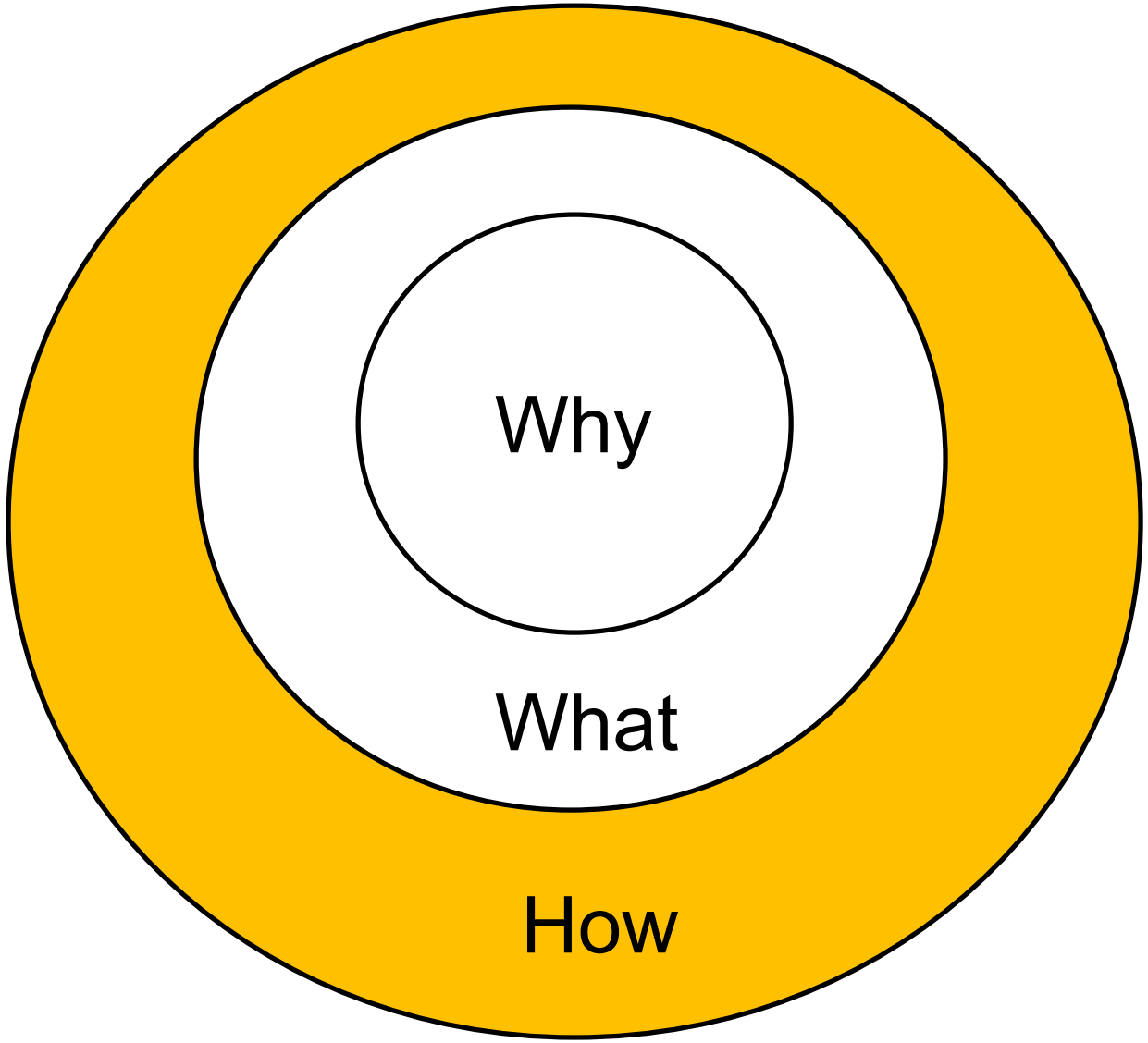
Last week

Given a set of *objects*, each object with descriptions of its properties, *partition* them into groups based on the descriptions, such that objects within one group are similar, and objects from different groups are dissimilar.

- Why?
  - Understanding, summarization
- What?
  - Numerical, nominal, ordinal; cluster, centroid; distance; validity index
- How?
  - *k*-means

Lecture 4: Clustering 2

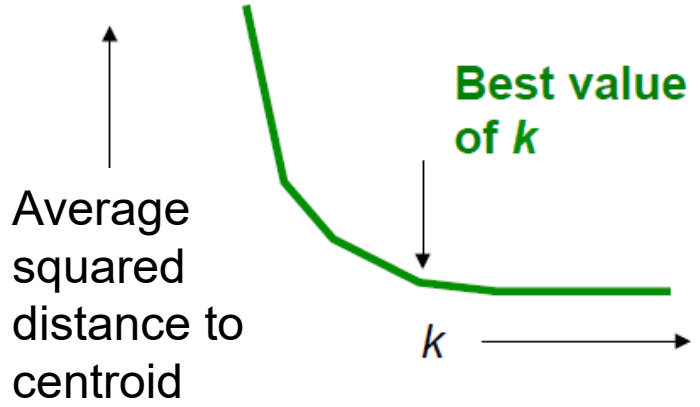
Continue



Pain of k-means: How to decide k

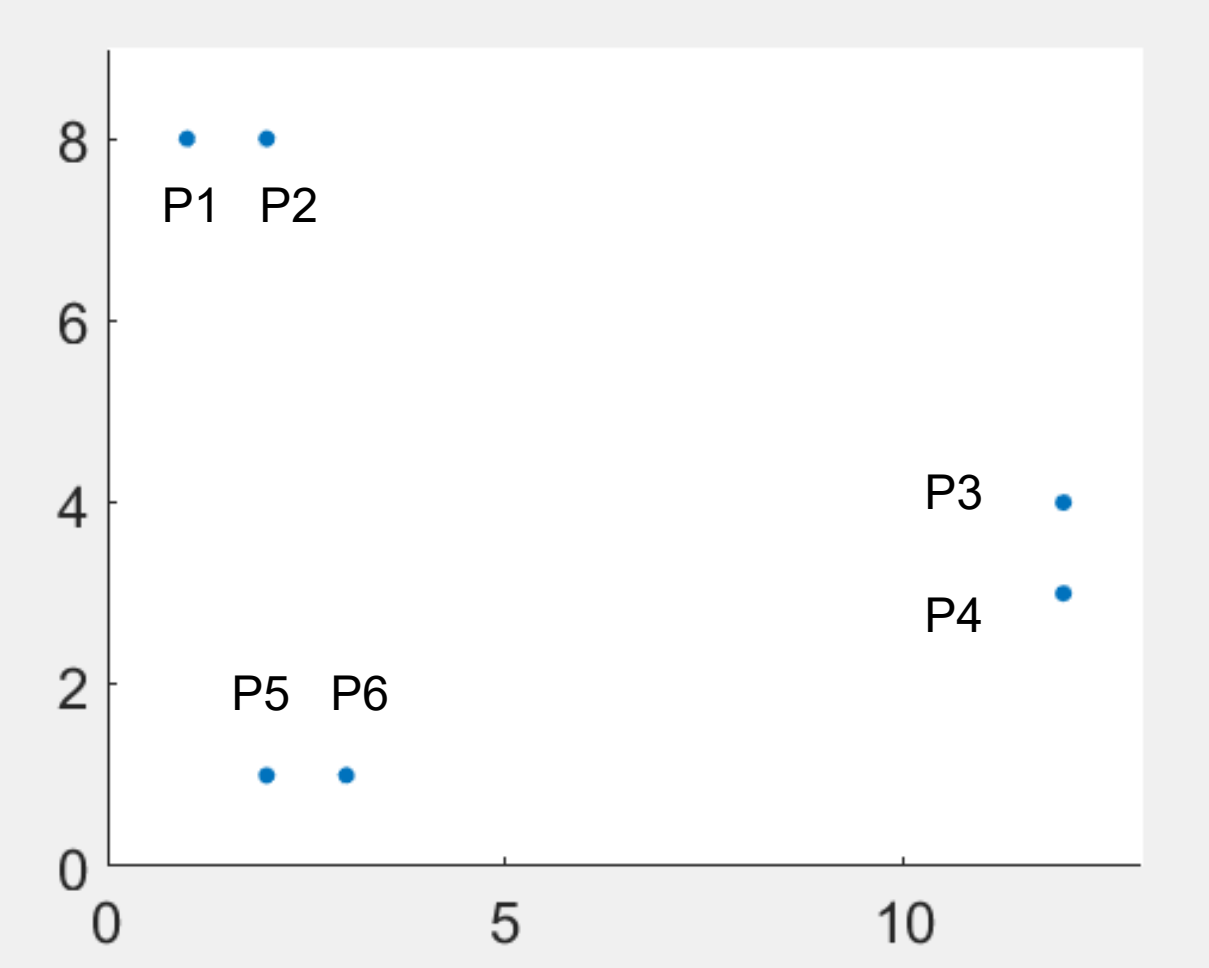
- **Elbow method**: try different k and see the average distance to centroid

$$\frac{1}{n}\sum_{i=1}^{k}\sum_{x\in C_i} dist(x, c_i)^2$$



- Try k from 2 to $\sqrt{n}$ (n is the number of all data points)

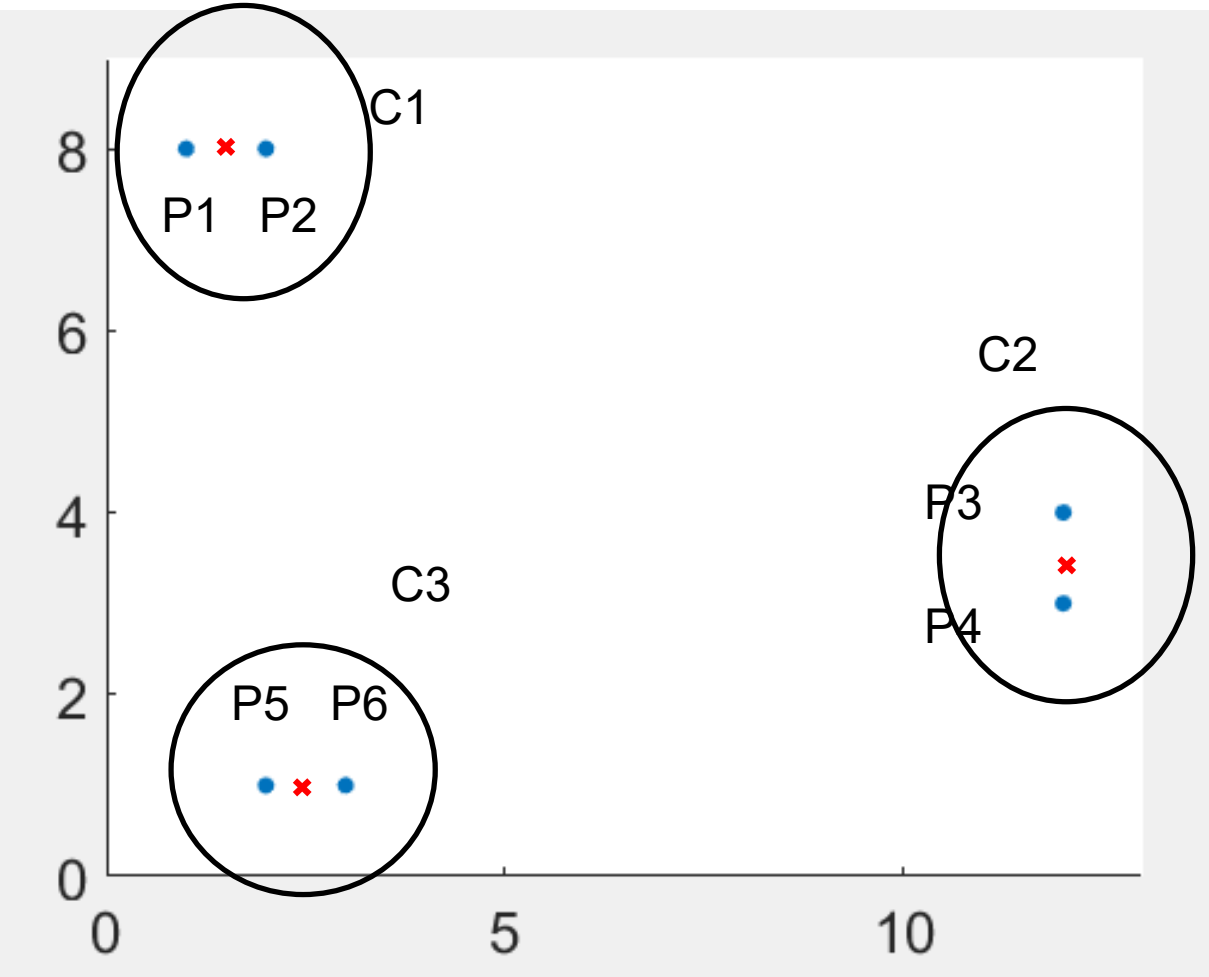
Example of the Elbow method



	x	y
P1	1	8
P2	2	8
P3	12	4
P4	12	3
P5	2	1
P6	3	1

$$\frac{1}{n}\sum_{i=1}^{k}\sum_{x\in C_i} dist(x, c_i)^2$$

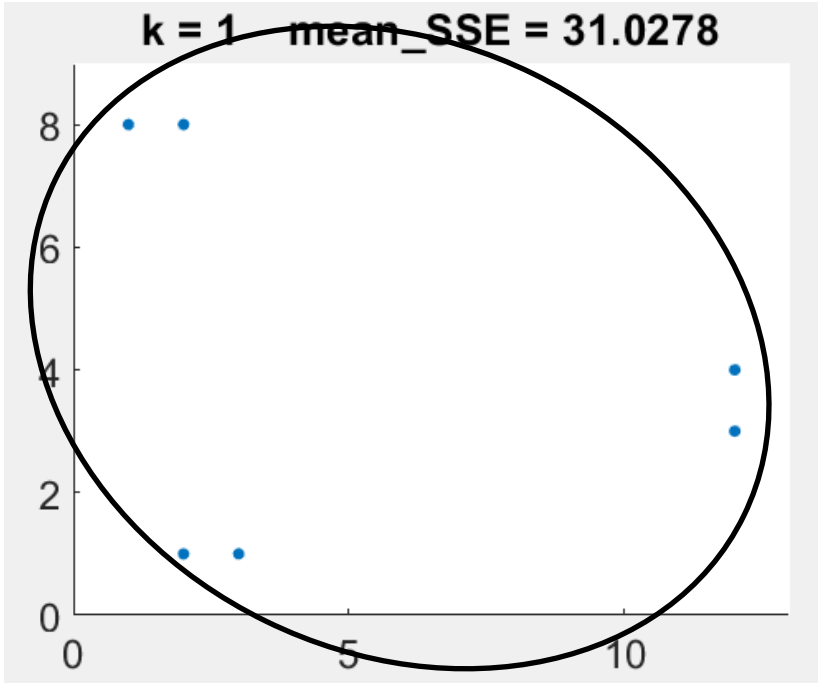
Example of the Elbow method: mean SSE for k=3

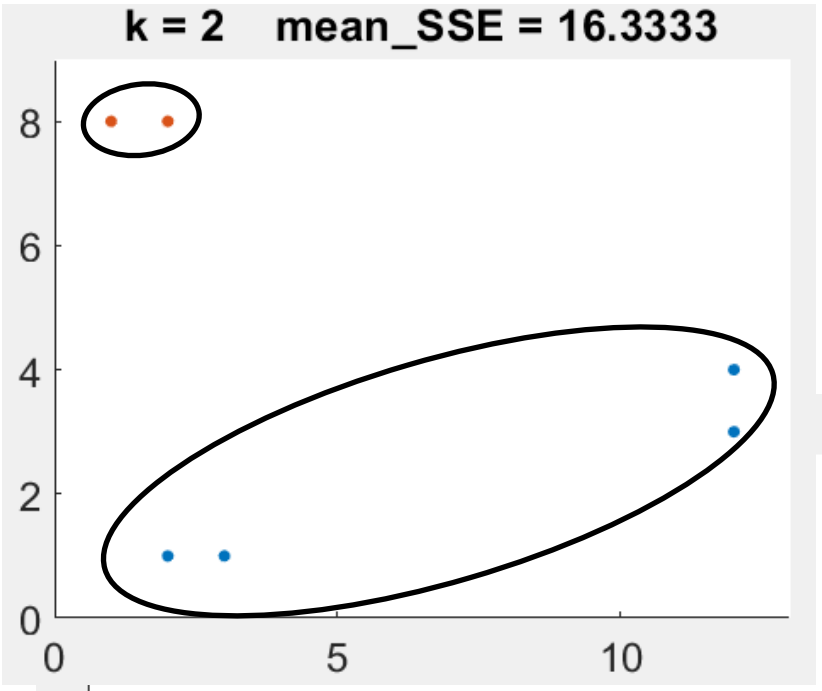


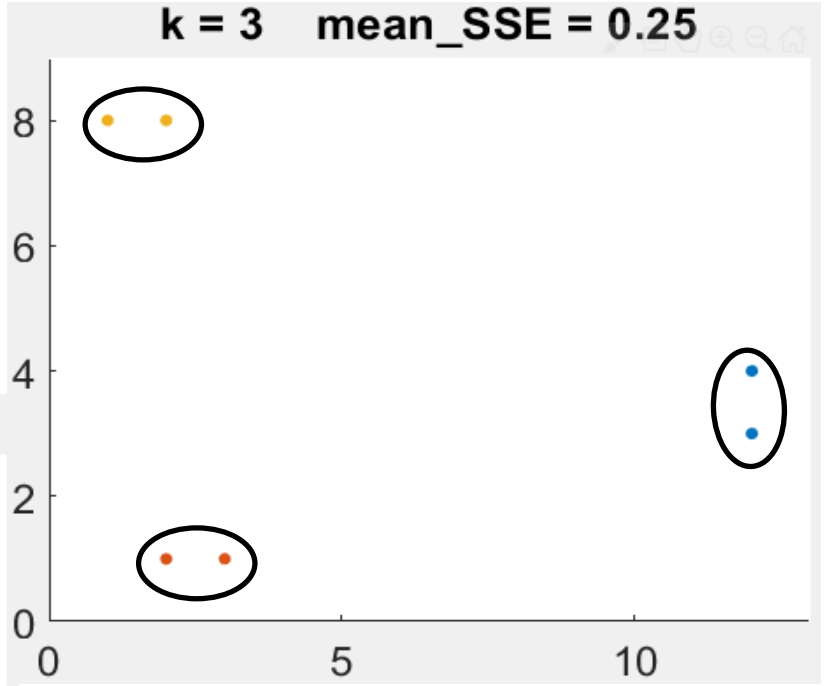
	x	y
P1	1	8
P2	2	8
P3	12	4
P4	12	3
P5	2	1
P6	3	1

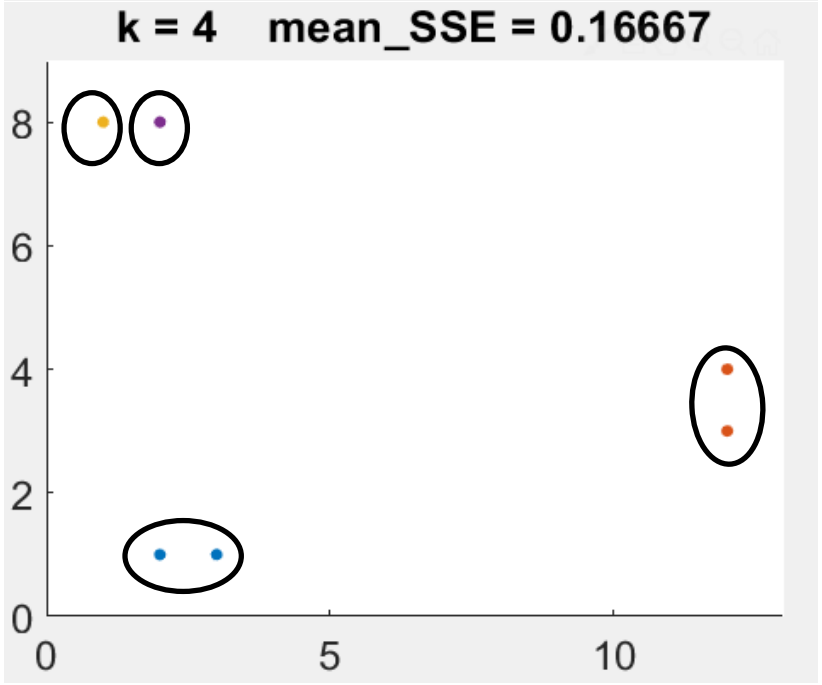
	x	y
c1	1.5	8
c2	12	3.5
c3	2.5	1

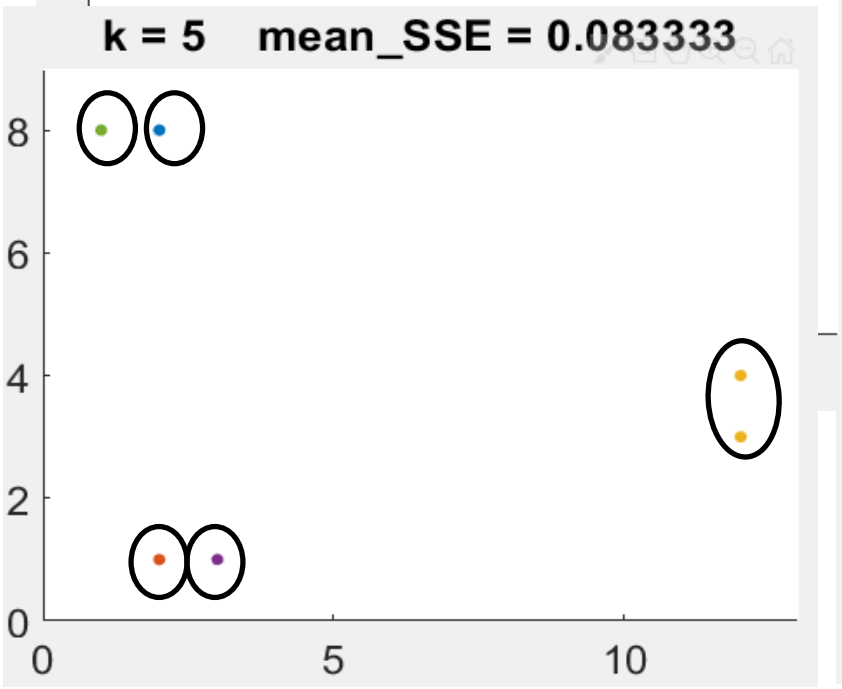
$$(dist(\mathrm{P1}, \mathrm{c1})^2 + dist(\mathrm{P2}, \mathrm{c1})^2 + dist(\mathrm{P3}, \mathrm{c2})^2 + dist(\mathrm{P4}, \mathrm{c2})^2 + dist(\mathrm{P5}, \mathrm{c3})^2 + dist(\mathrm{P6}, \mathrm{c3})^2)/6$$

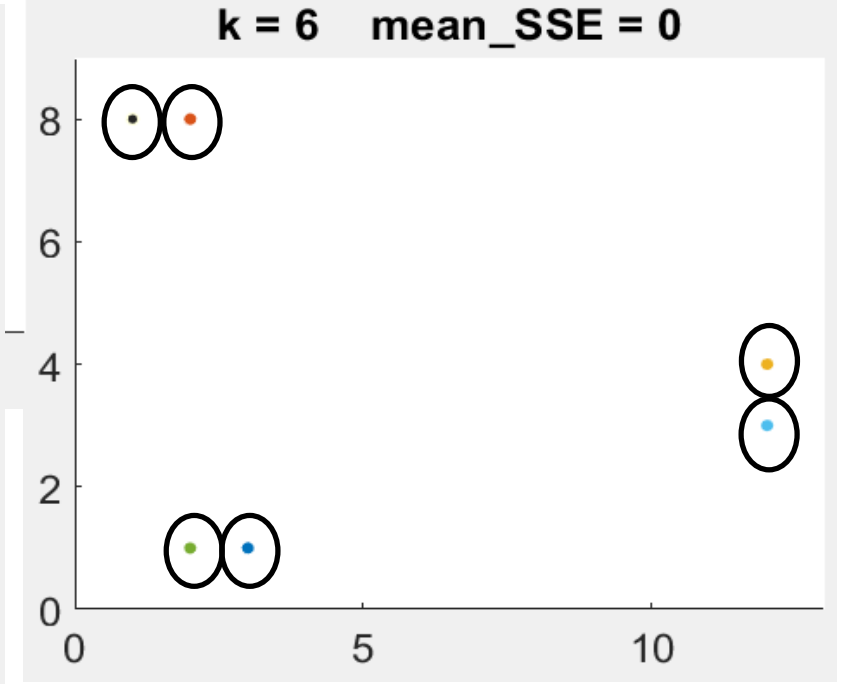

k = 1 mean_SSE = 31.0278


k = 2 mean_SSE = 16.3333

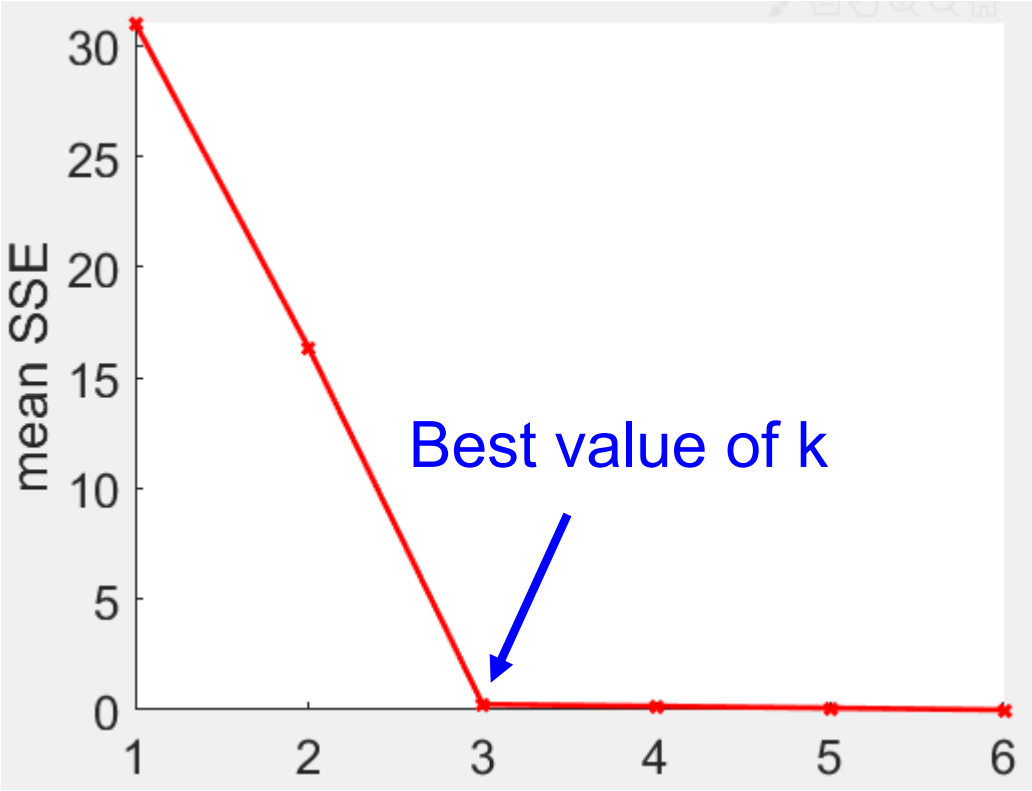

k = 3 mean_SSE = 0.25


k = 4 mean_SSE = 0.16667

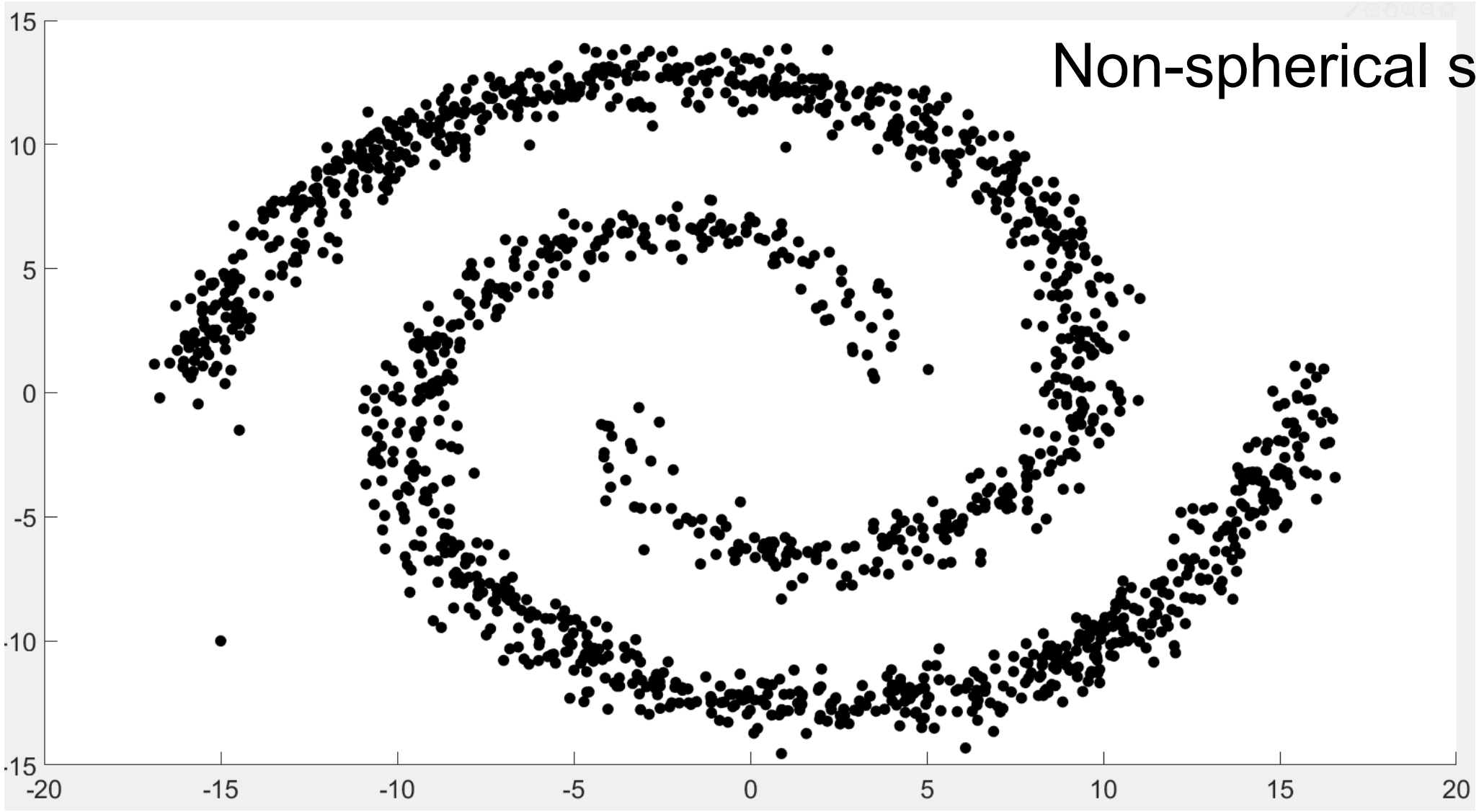

k = 5 mean_SSE = 0.083333


k = 6 mean_SSE = 0

Example of the Elbow method



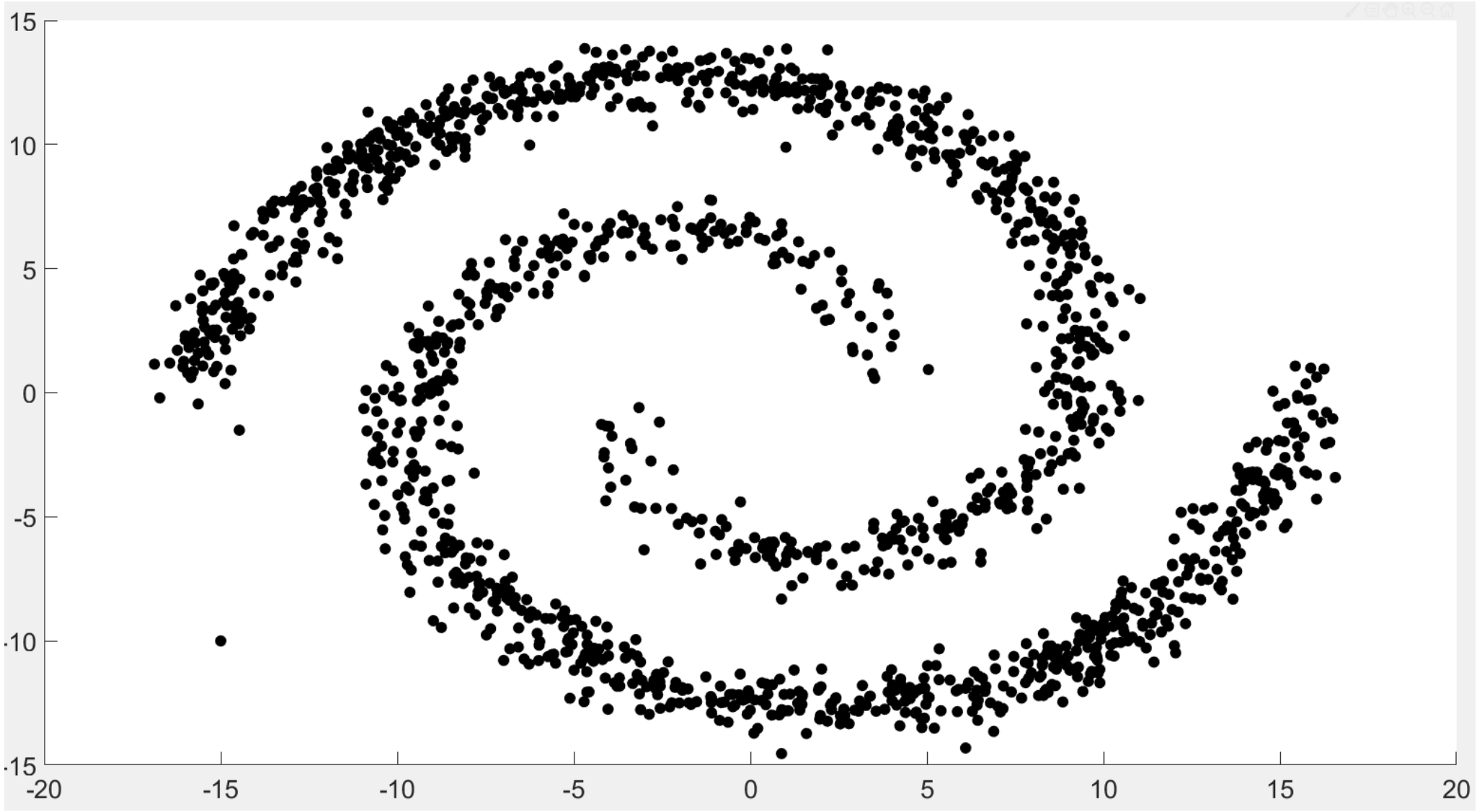
Pain of *k*-means: sometimes cannot work well



Outline

- Prototype-based clustering
  - k-means
- Density-based clustering
  - ??
- Hierarchical-based clustering
  - ??

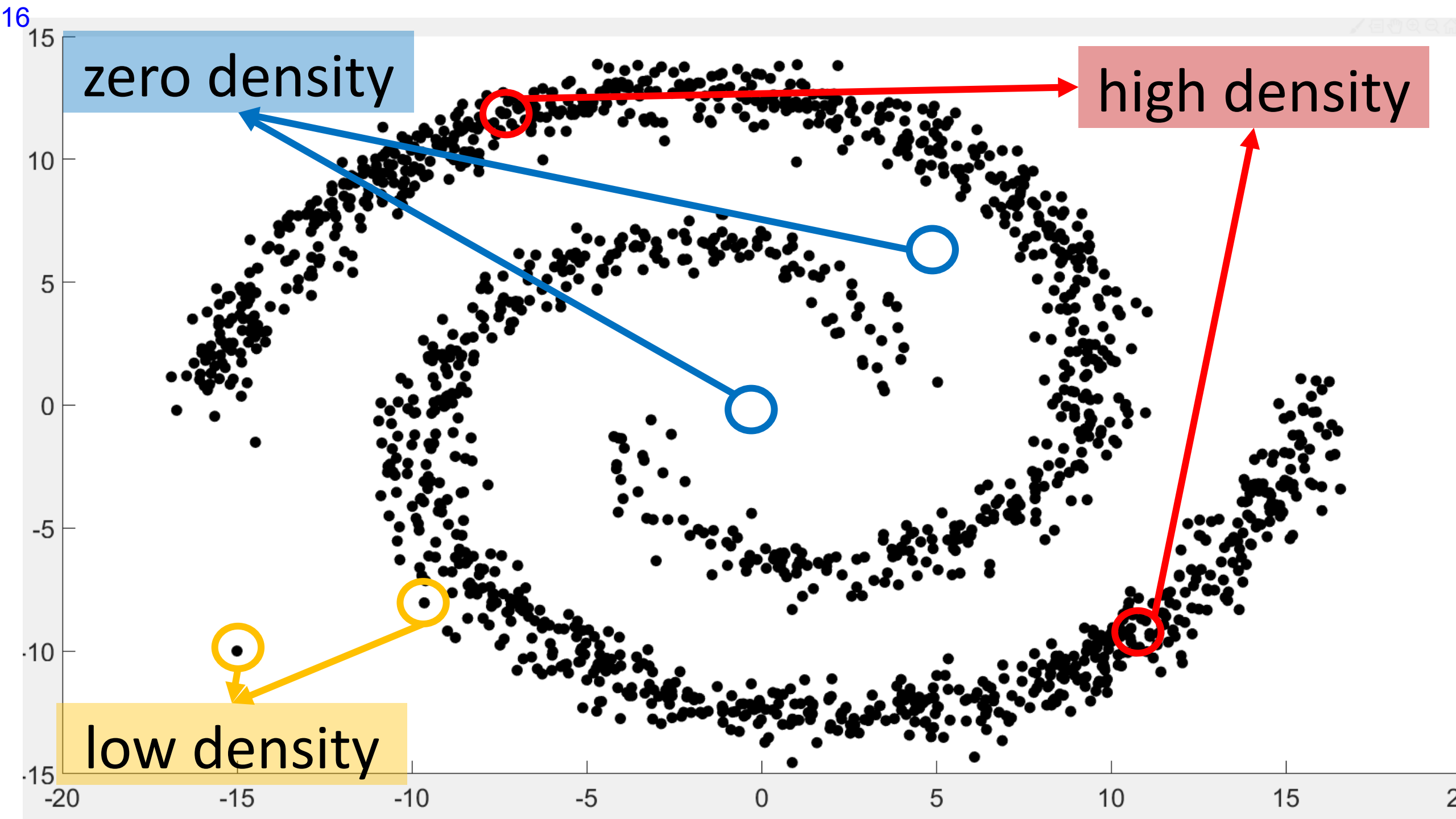
Consider the case that k-means cannot work well



Observe

If we draw a small circle in any area

- The circle contains a lot of points: high density
- The circle contains no point: zero density
- The circle contains a few, or only one point: low density

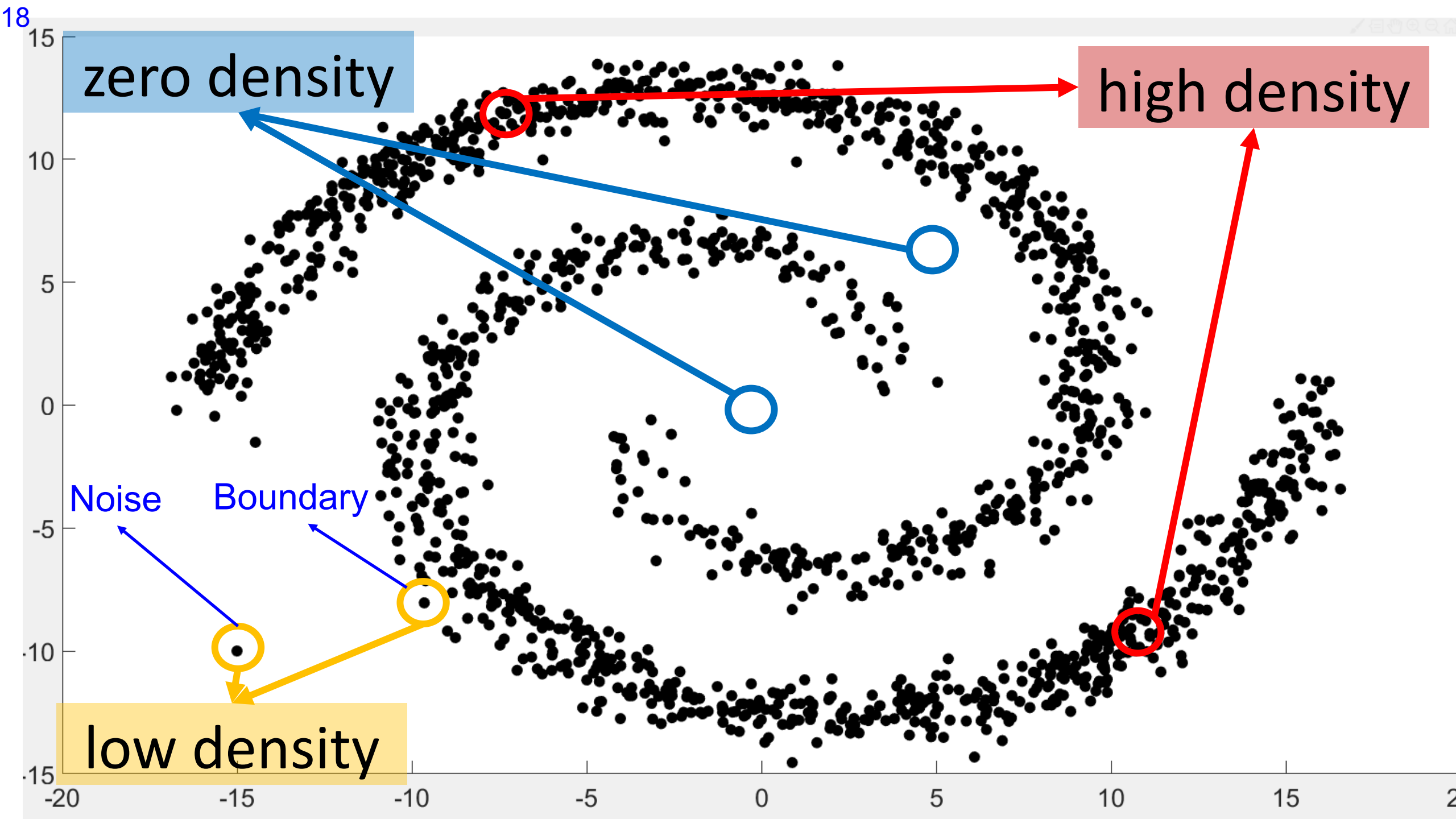

zero density
high density
low density
15
10
5
0
-5
-10
-15
-20
-15
-10
-5
0
5
10
15

High density region vs. zero-density region

If we draw a small circle in any area

- The circle contains a lot of points: high density
  - In cluster
- The circle contains no point: zero density
  - Between clusters
- The circle contains a few, or only one point: low density

How about low density?


zero density
high density
Noise
Boundary
low density
15
10
5
0
-5
-10
-15
-20
-15
-10
-5
0
5
10
15

High density region vs. zero-density region

If we draw a small circle in any area

- The circle contains a lot of points: high density
  - In cluster
- The circle contains no point: zero density
  - Between clusters
- The circle contains a few, or only one point: low density
  1. Boundary of cluster
  2. Noise

Core, Border, and Noise Points

Specify two parameters by human beings:

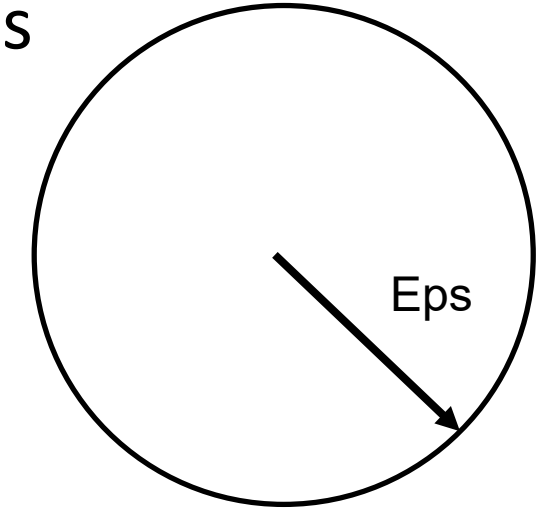
- –Eps: positive number
- –MinPts: positive integer

- A core point:
  - –draw a circle centred it with radius Eps
  - –the number of points with the circle is at least MinPts
- A border point:
  - –not a core point, but within a circle of radius Eps of a core point
- A noise point:
  - –all other points

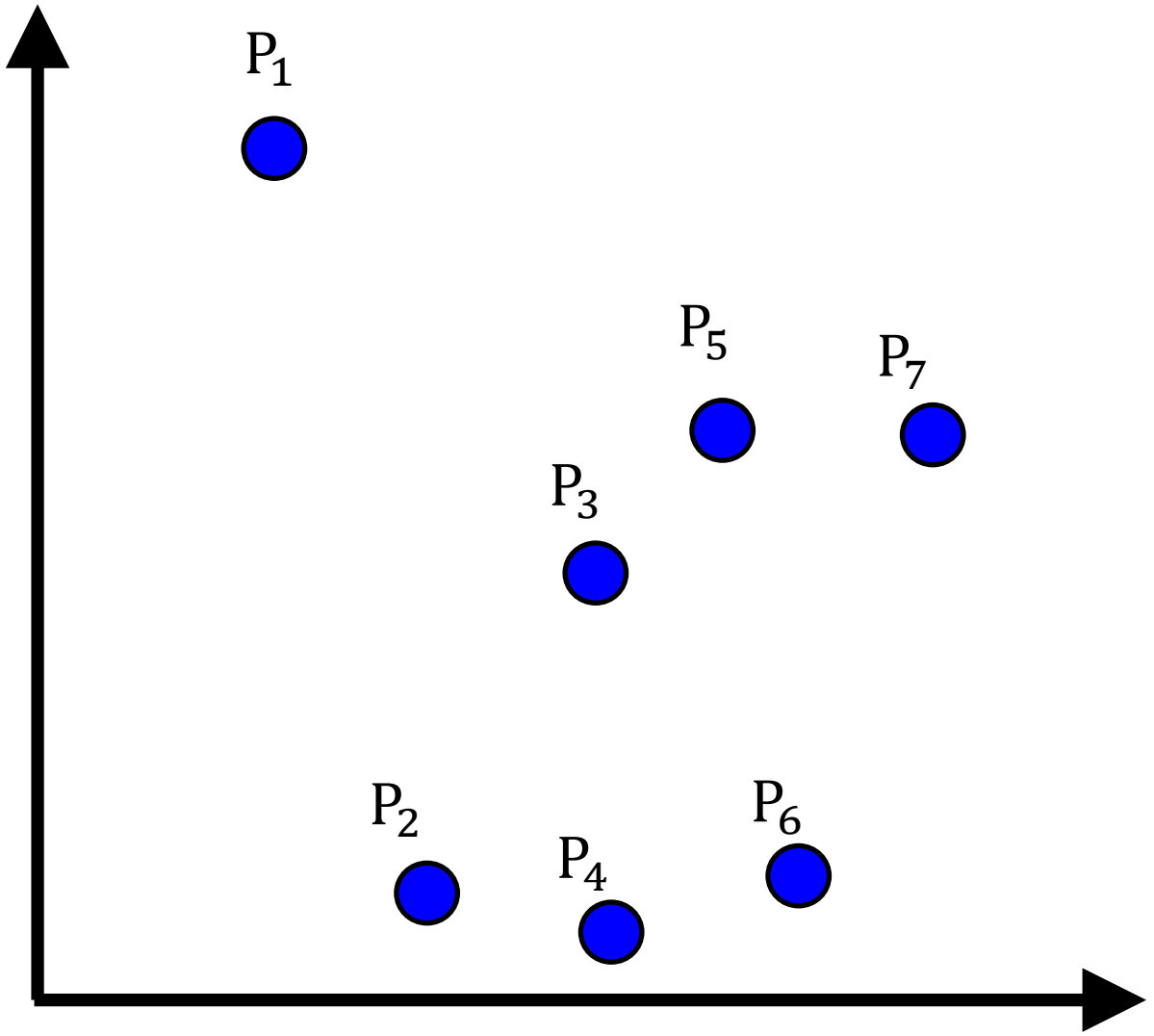
Example of Core, Border, and Noise Points

- Given the seven points

- Setting
  - Eps

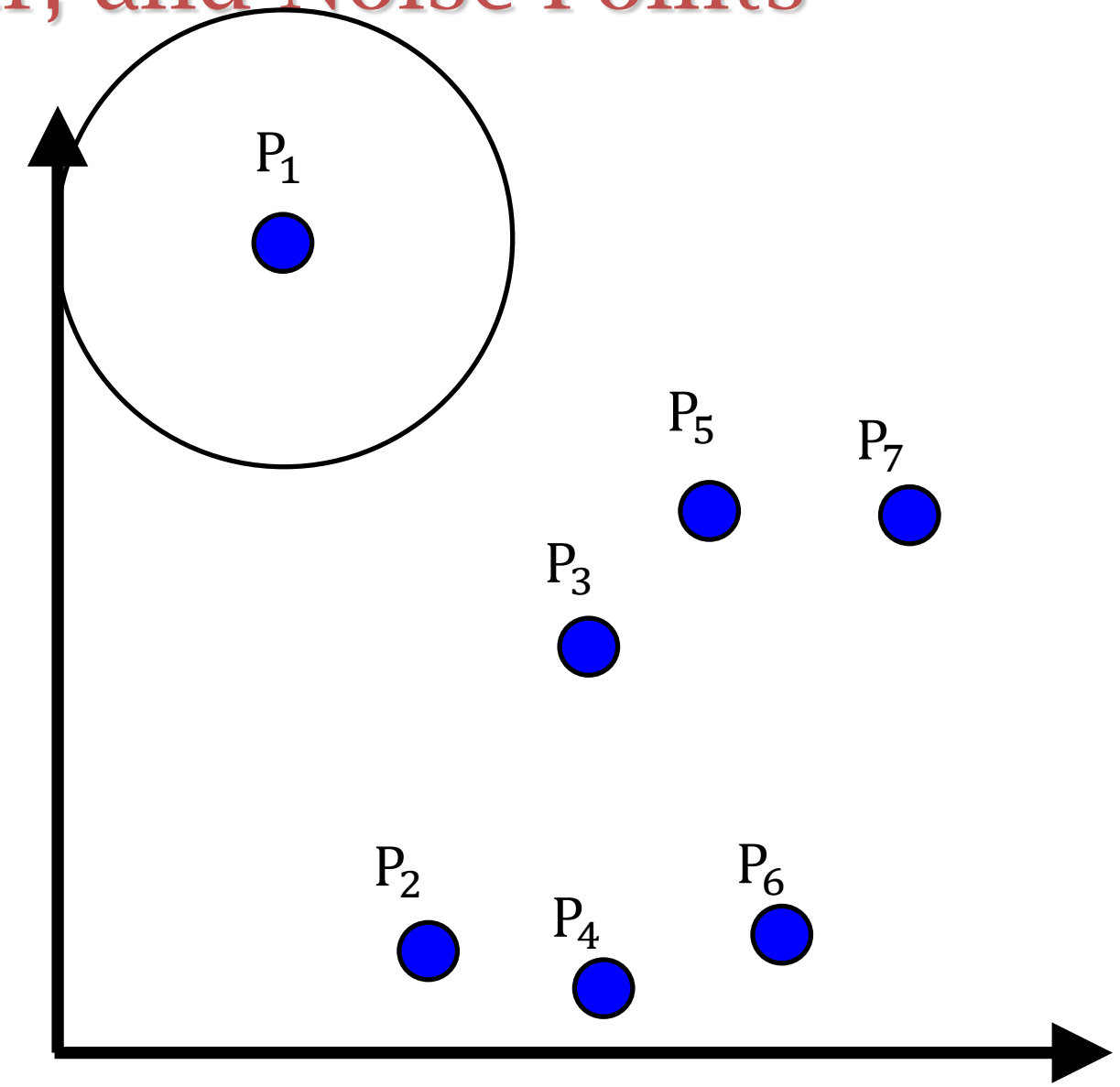


  - MinPts = 3



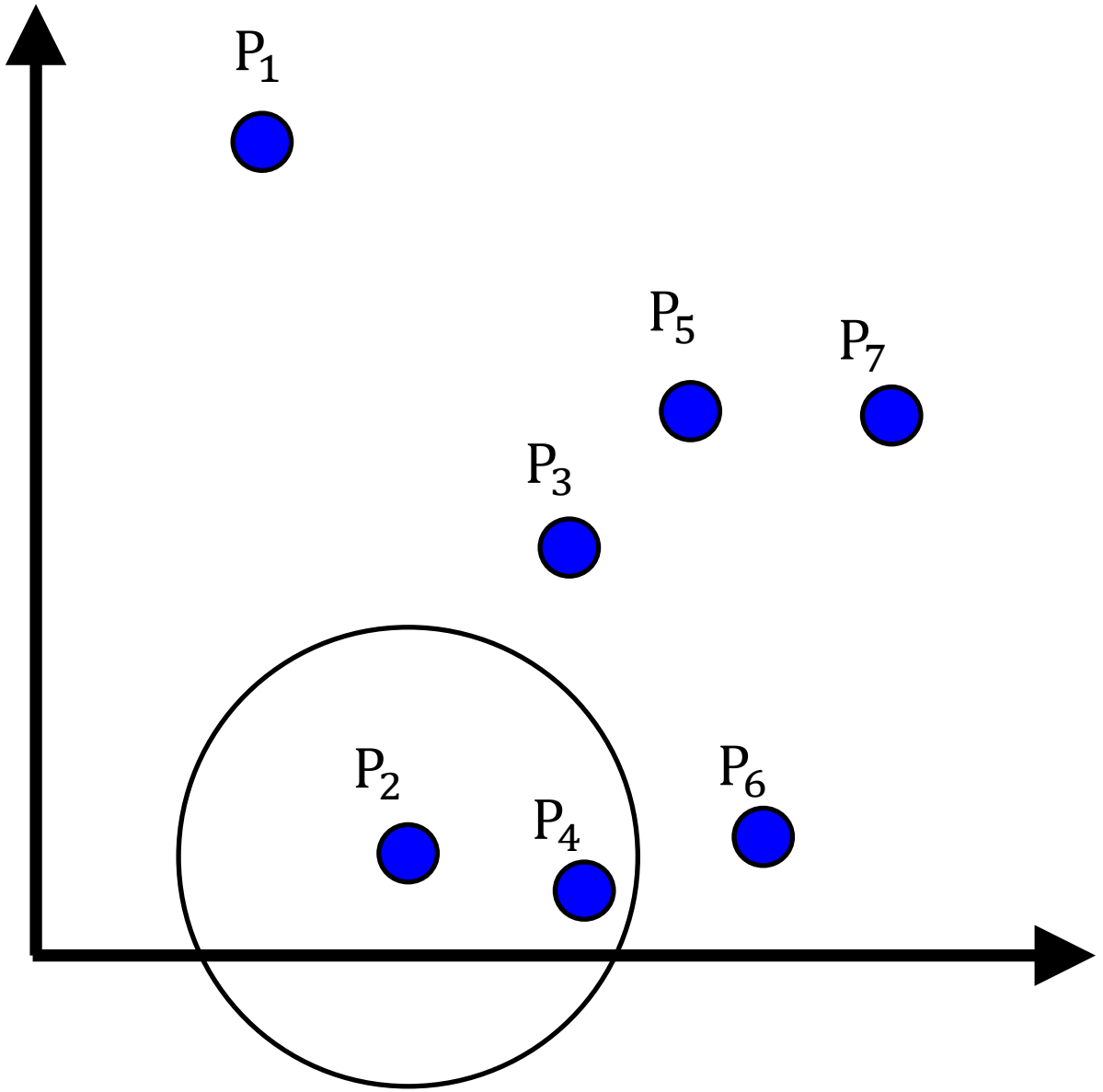
Example of Core, Border, and Noise Points

- Points within the circle: -1
- P_1 is not a core point



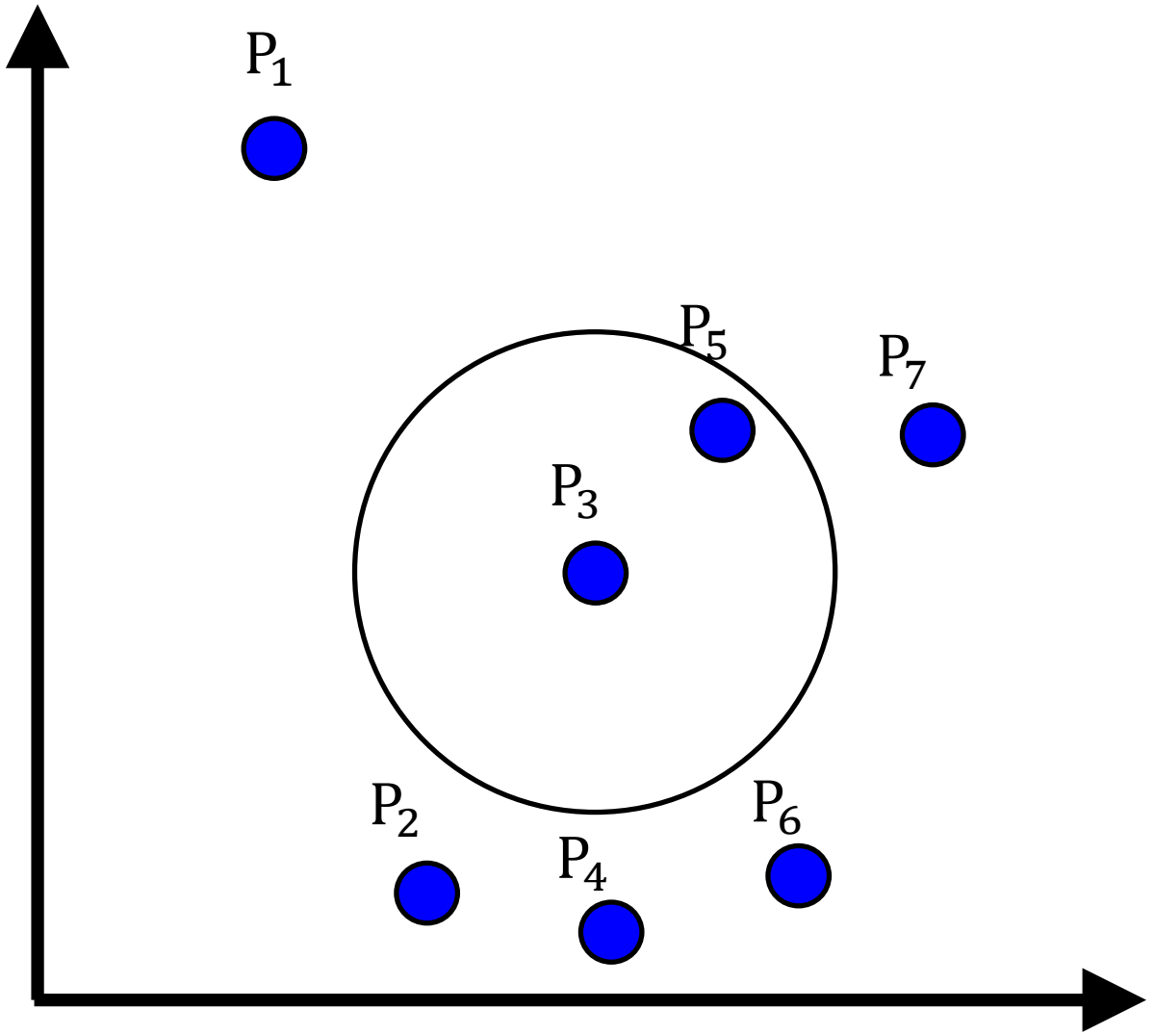
Example of Core, Border, and Noise Points

- Points within the circle: -2
- P_2 is **not** a core point



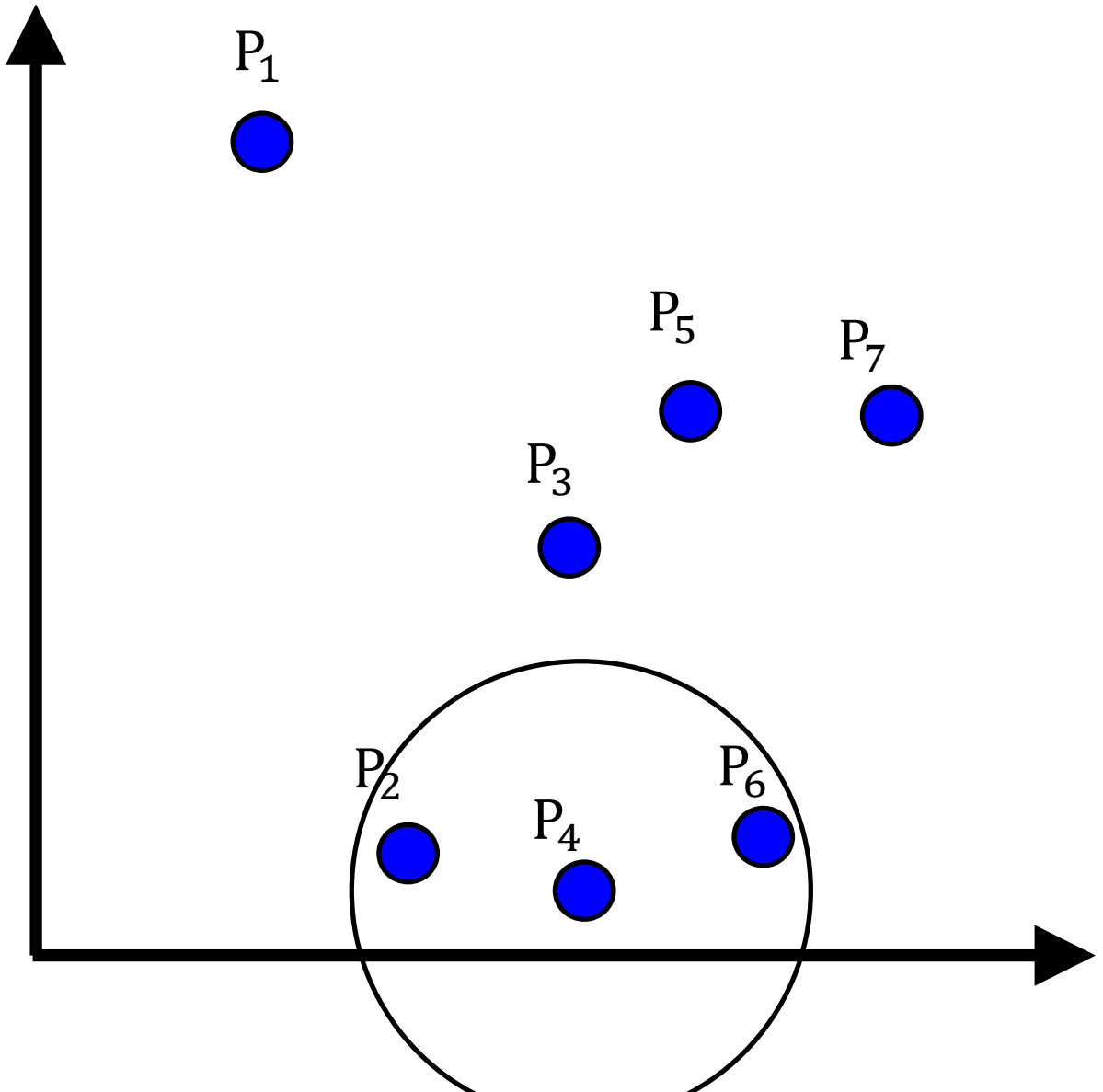
Example of Core, Border, and Noise Points

- Points within the circle: -2
- P_3 is **not** a core point



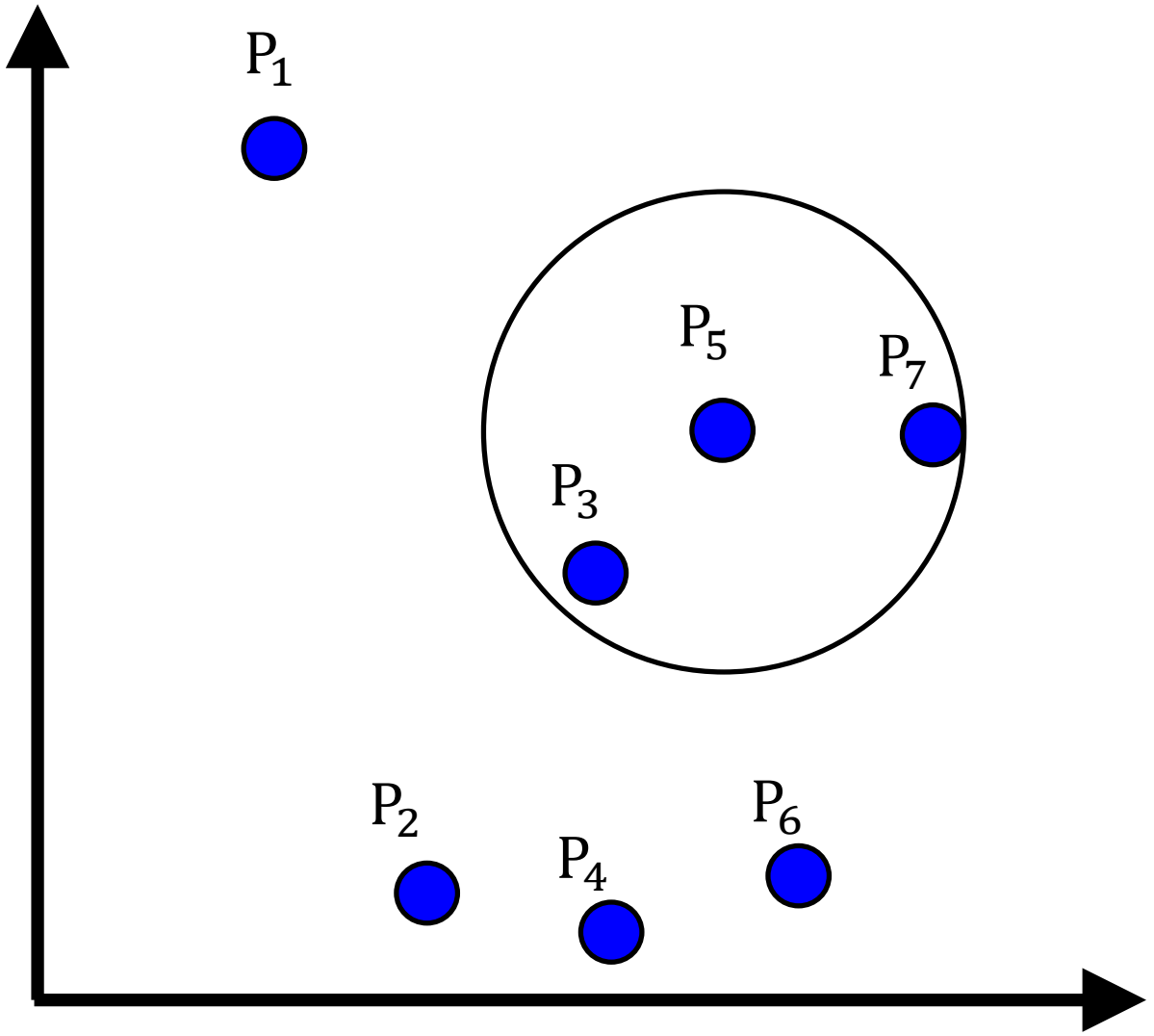
Example of Core, Border, and Noise Points

- Points within the circle: -3
- P_4 is a core point
- P_2 is a border point



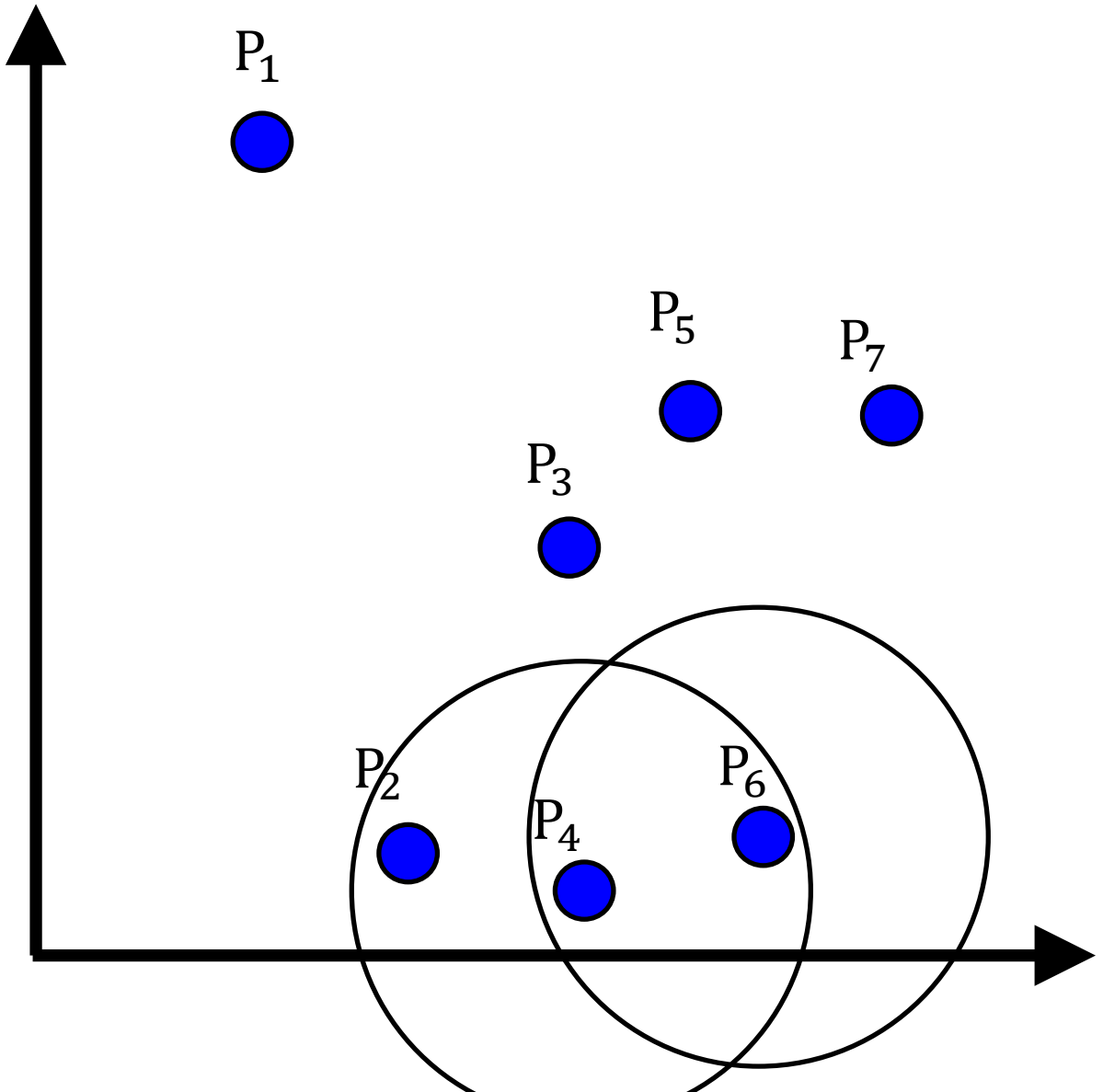
Example of Core, Border, and Noise Points

- Points within the circle: -3
- P_5 is a core point
- P_3 is a border point



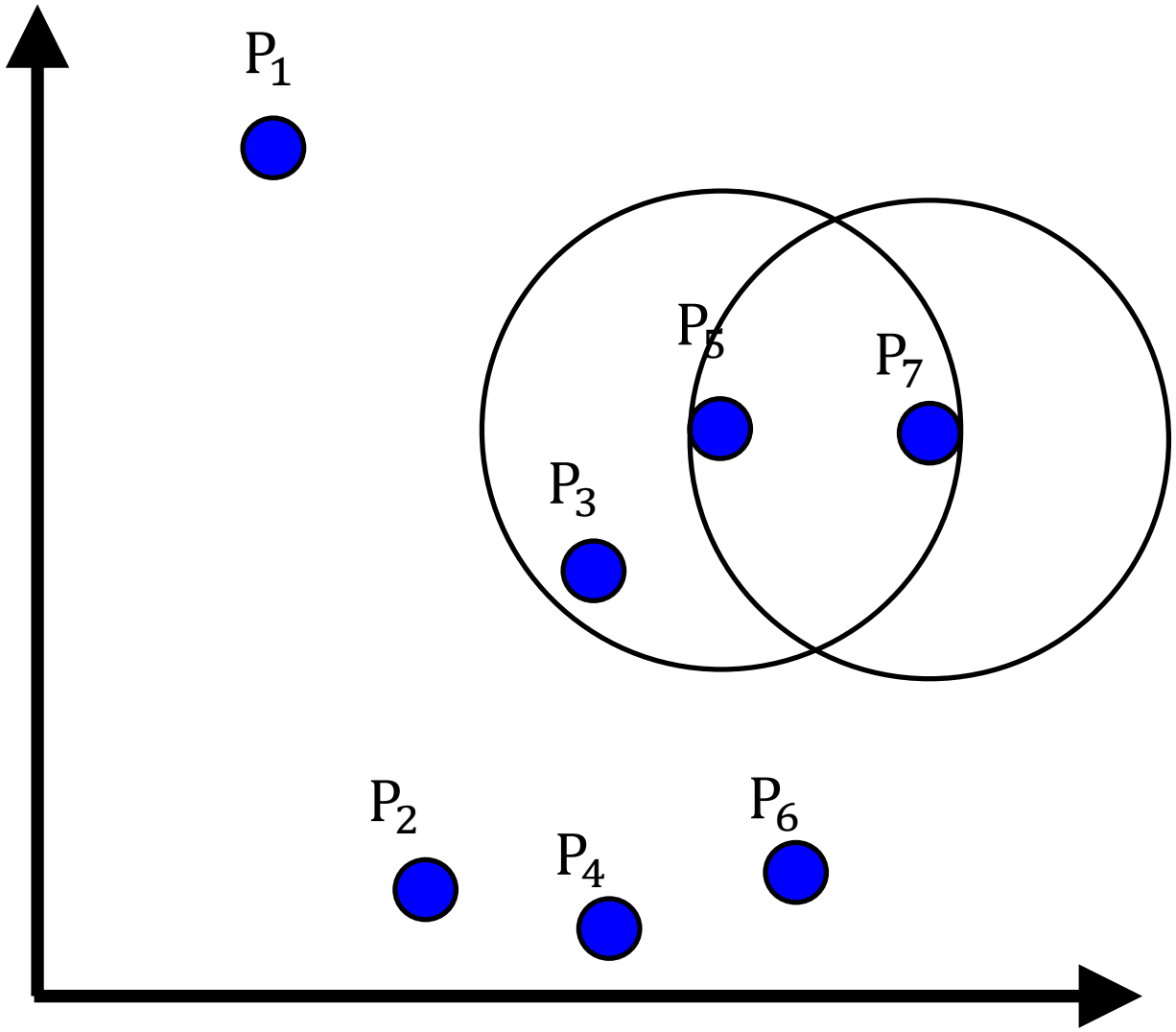
Example of Core, Border, and Noise Points

- Point within the circle: -2
- P_6 is not a core point
- P_6 is a border point



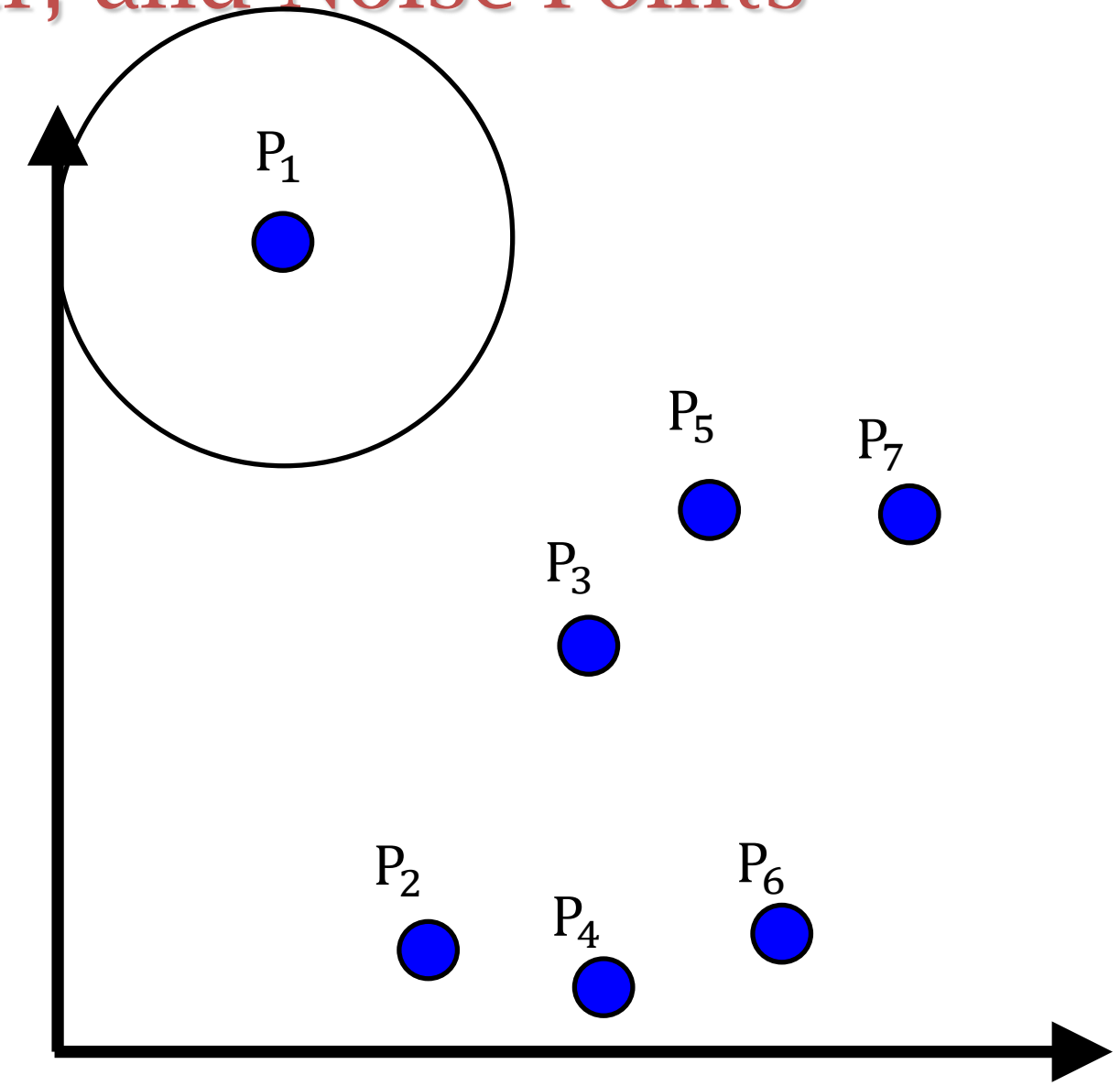
Example of Core, Border, and Noise Points

- Point within the circle: -2
- P_7 is not a core point
- P_7 is a border point



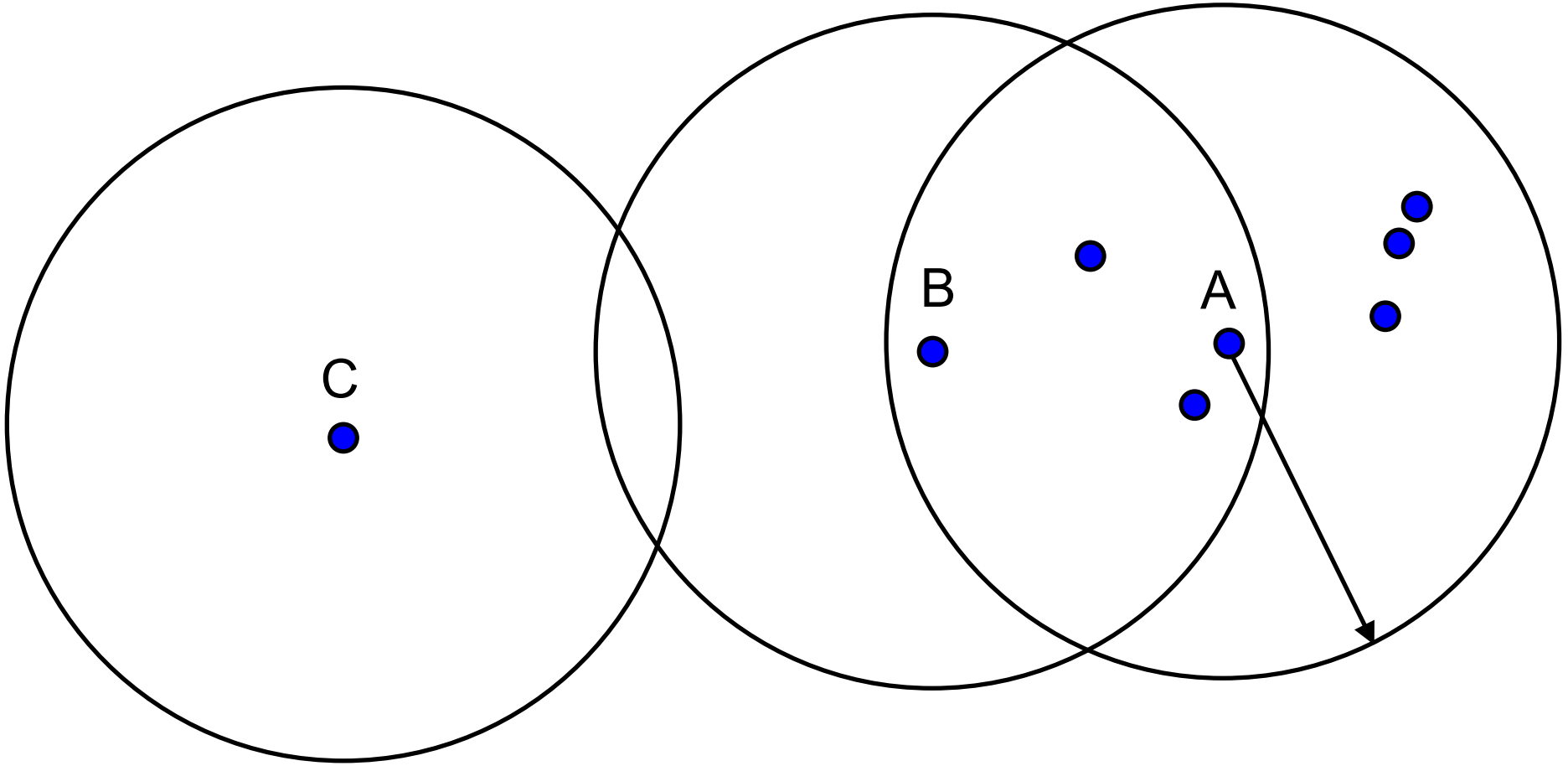
Example of Core, Border, and Noise Points

- Point within the circle: -1
- P_1 is not a core point
- P_1 is not a border point
- P_1 is a noisy point!



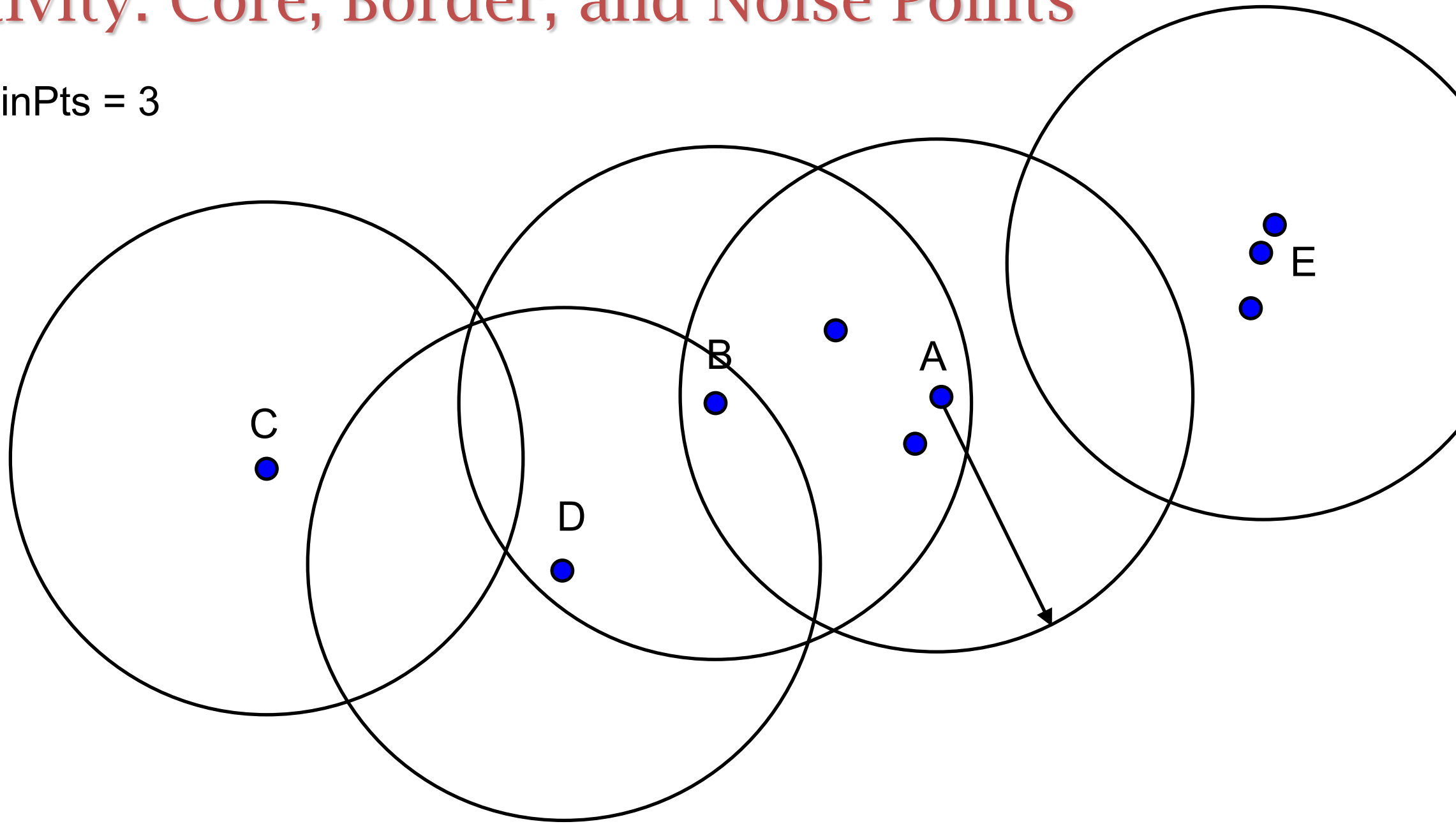
Activity: Core, Border, and Noise Points

MinPts = 7



Activity: Core, Border, and Noise Points

MinPts = 3

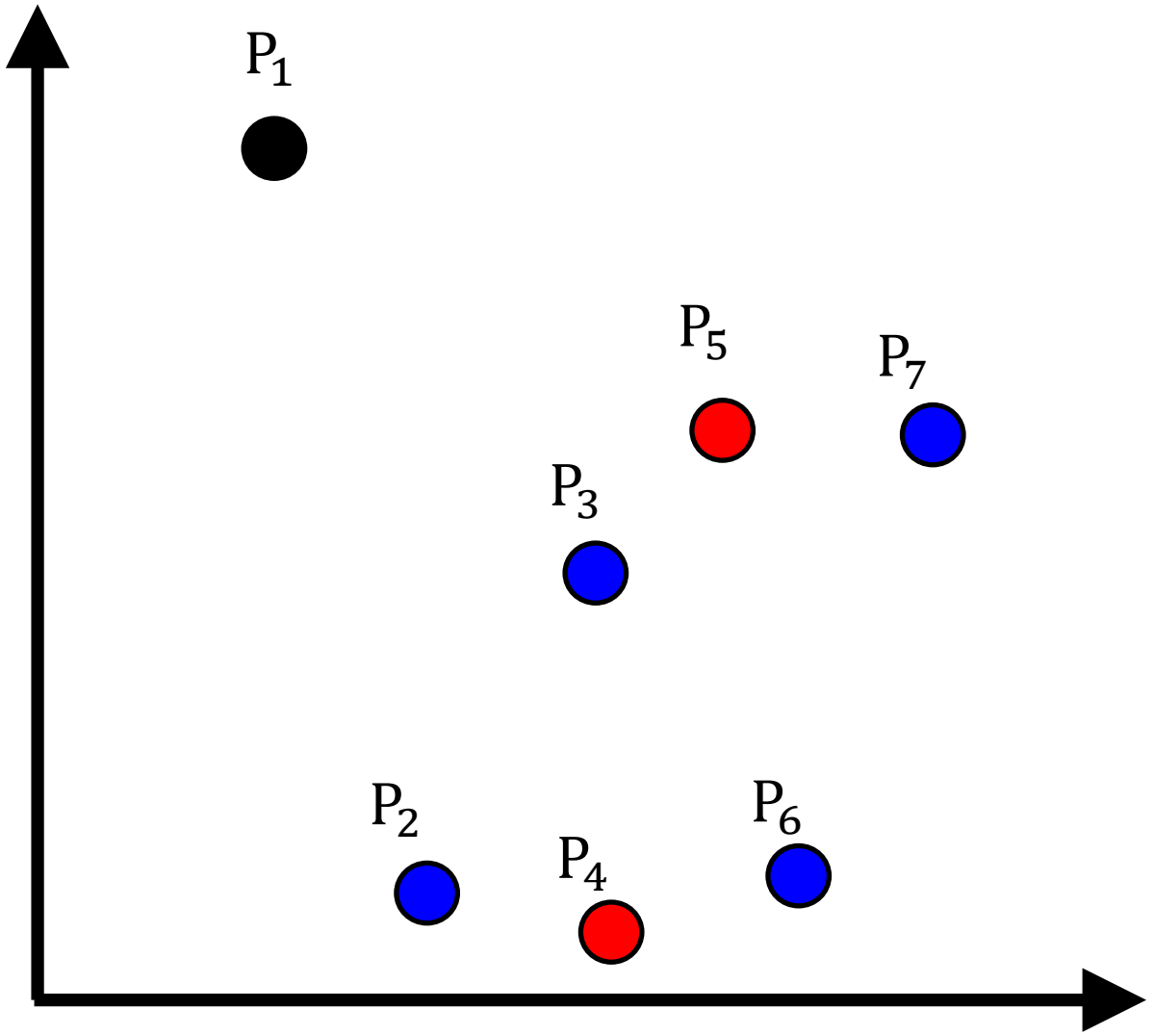


The density-based clustering algorithm: DBSCAN

1. Label all points as noise, core or border points
2. Eliminate all noise points
3. ***Initialization:*** cluster_index = 0
4. ***for*** all core points
5. ***if*** it is not assigned a cluster_index
6. cluster_index = cluster_index +1
7. assign the core point the cluster_index
8. ***for*** all points within the *Eps* circle of the core point
9. ***if*** it is not assigned a cluster_index
10. assign the point the cluster_index

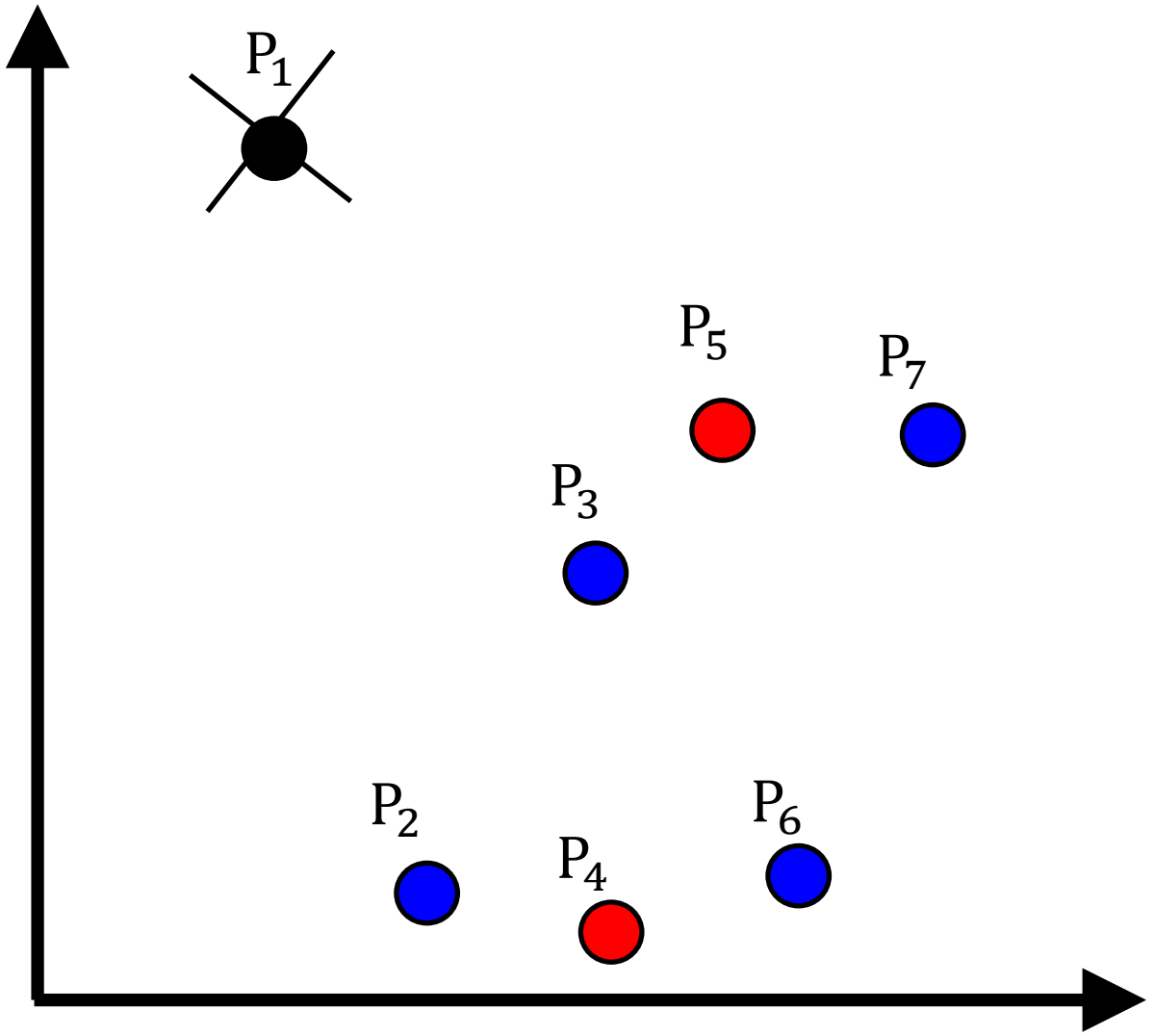
How to do clustering: an example

1. Label all points as noise, core or border points



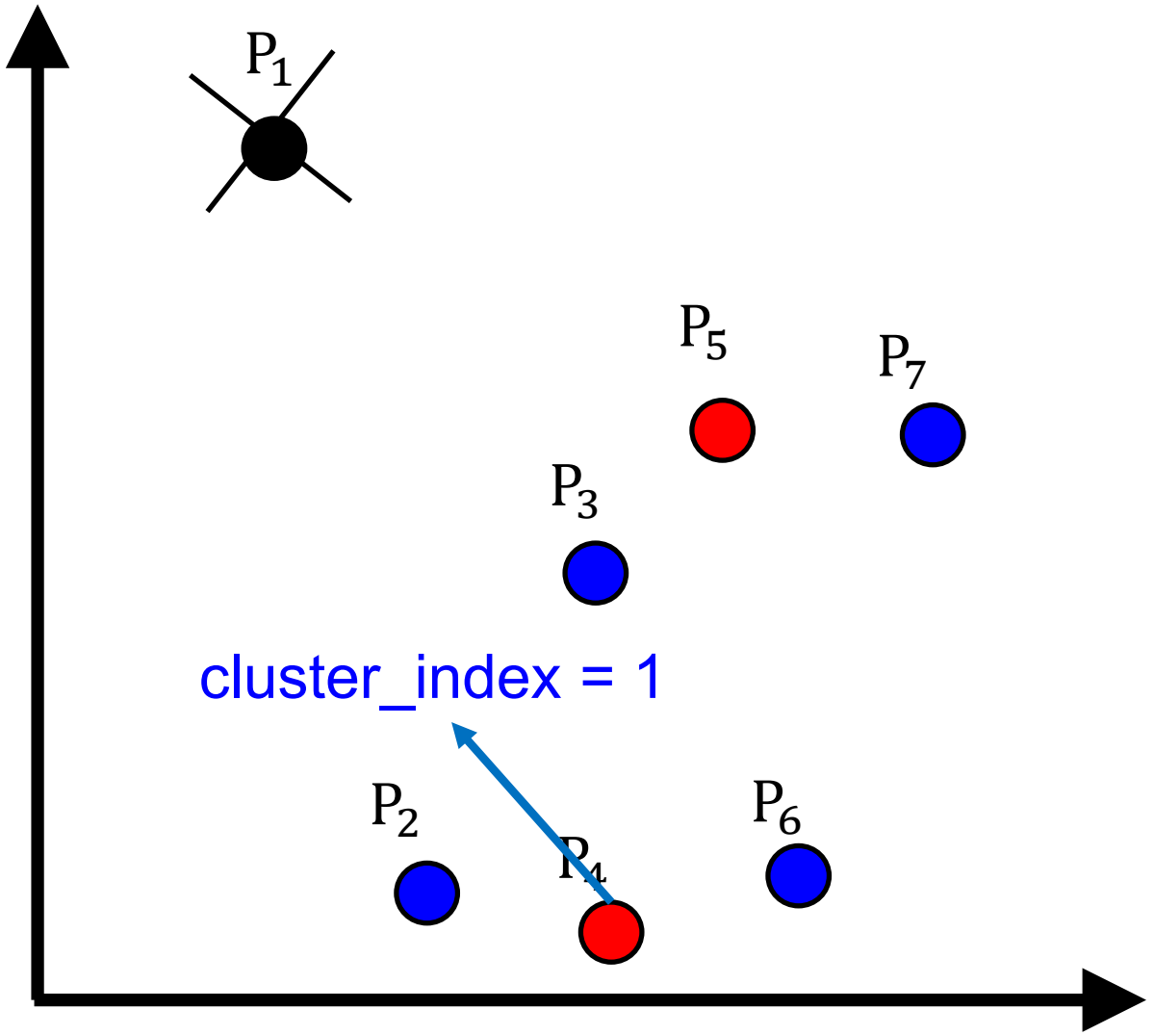
How to do clustering: an example

2. Eliminate all noise points
3. ***Initialization:*** cluster_index = 0



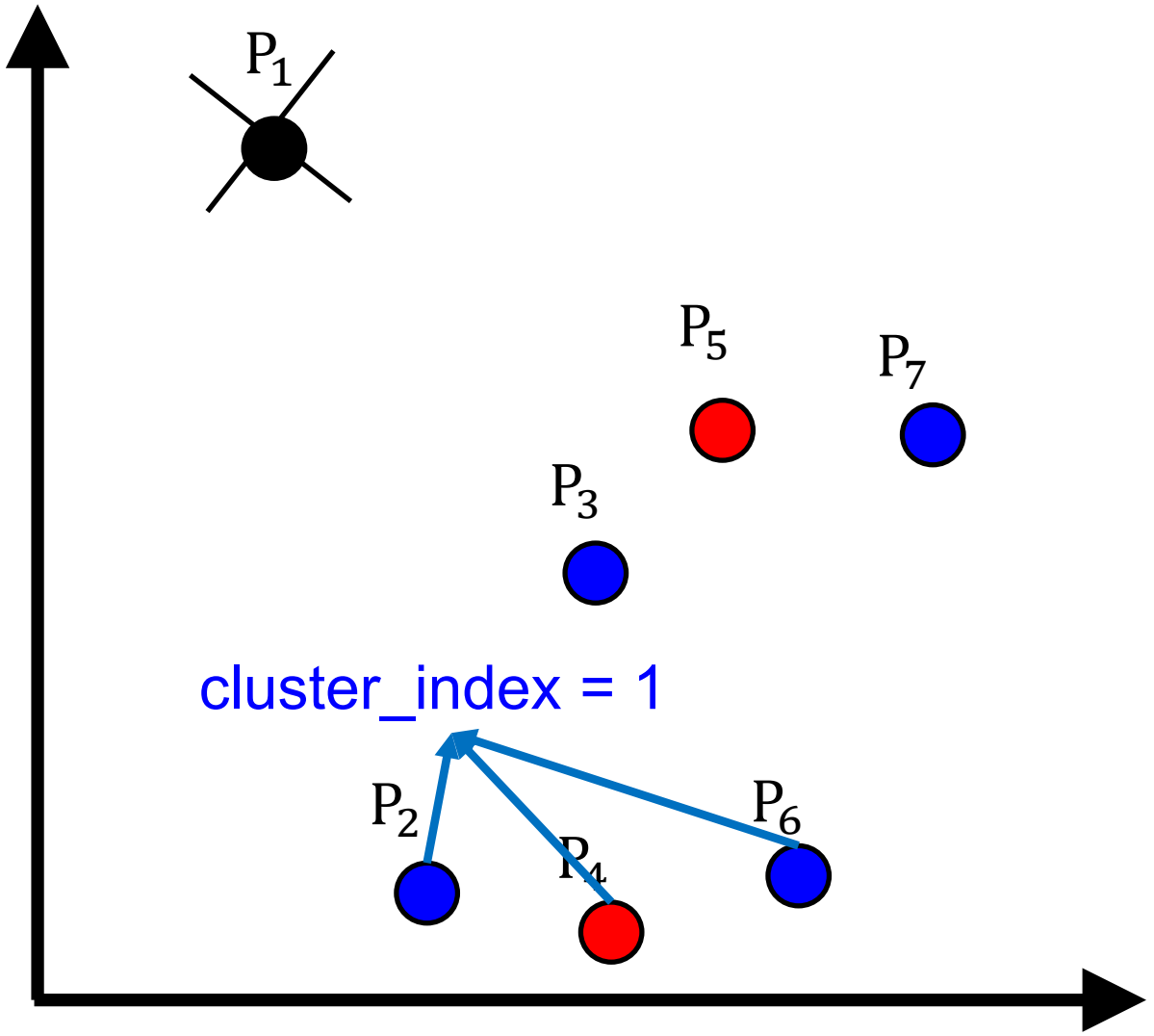
How to do clustering: an example

4. ***for*** all core points
5. ***if*** it is not assigned a cluster_index
6. cluster_index = cluster_index +1
7. assign the core point the cluster_index



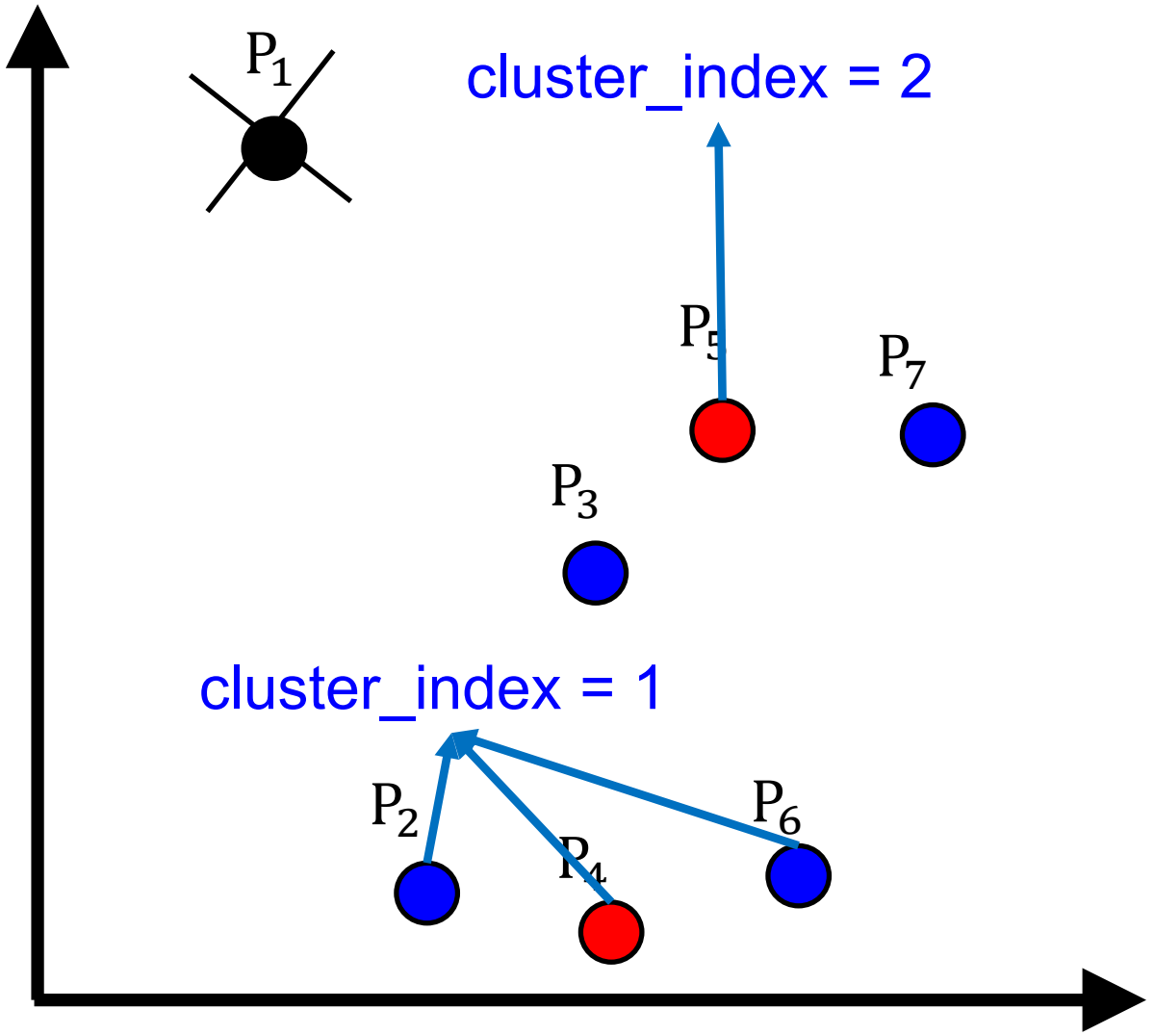
How to do clustering: an example

8. ***for*** all points within the *Eps* circle of the core point
9. ***if*** it is not assigned a cluster_index
10. assign the point the cluster_index



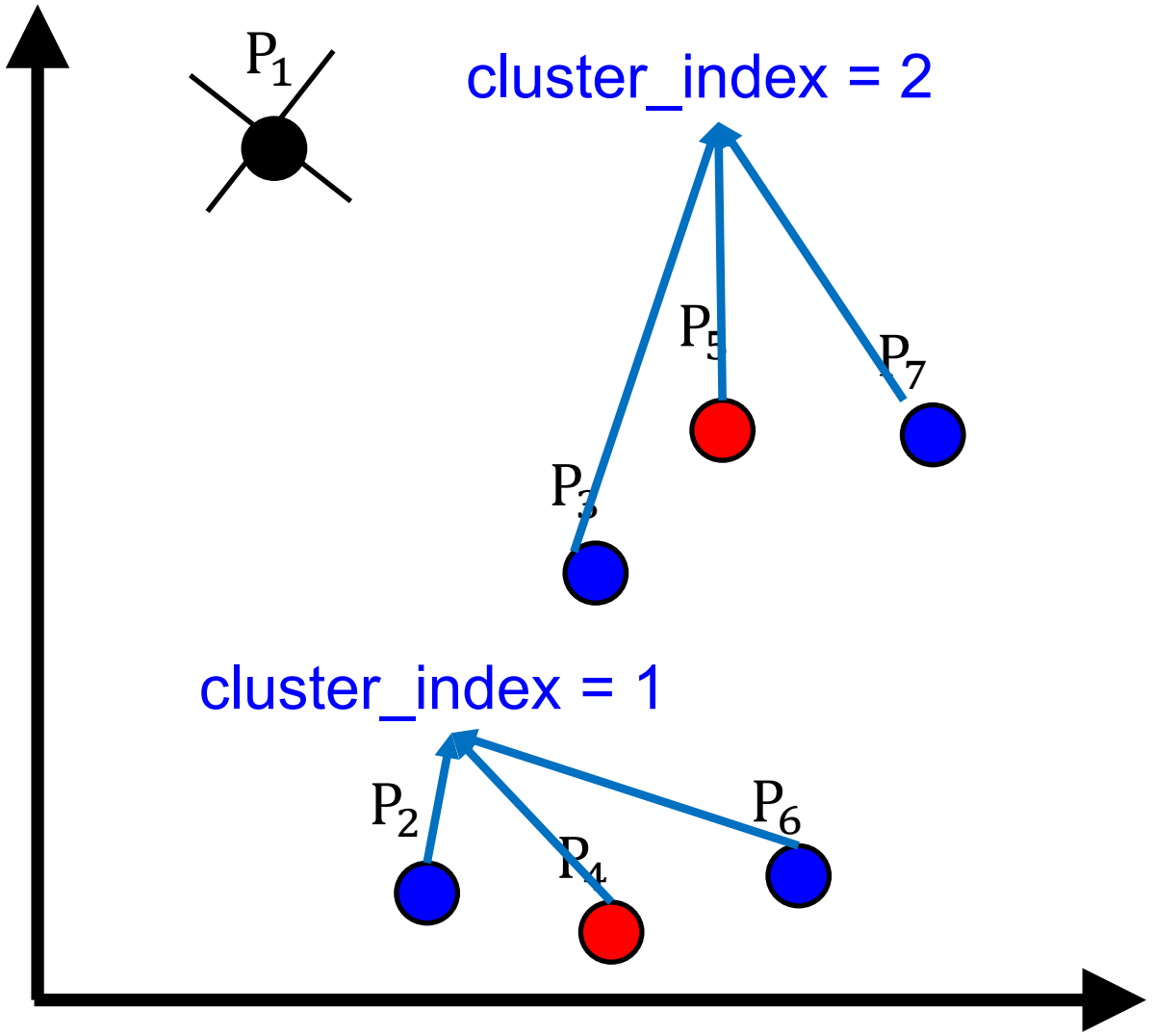
How to do clustering: an example

4. ***for*** all core points
5. ***if*** it is not assigned a cluster_index
6. cluster_index = cluster_index +1
7. assign the core point the cluster_index



How to do clustering: an example

8. ***for*** all points within the *Eps* circle of the core point
9. ***if*** it is not assigned a cluster_index
10. assign the point the cluster_index



How to do clustering: an example

- *cluster_index = 1*
  - $\{P_2, P_4, P_6\}$
- *cluster_index = 2*
  - $\{P_3, P_5, P_7\}$
- P_1 is a noise point

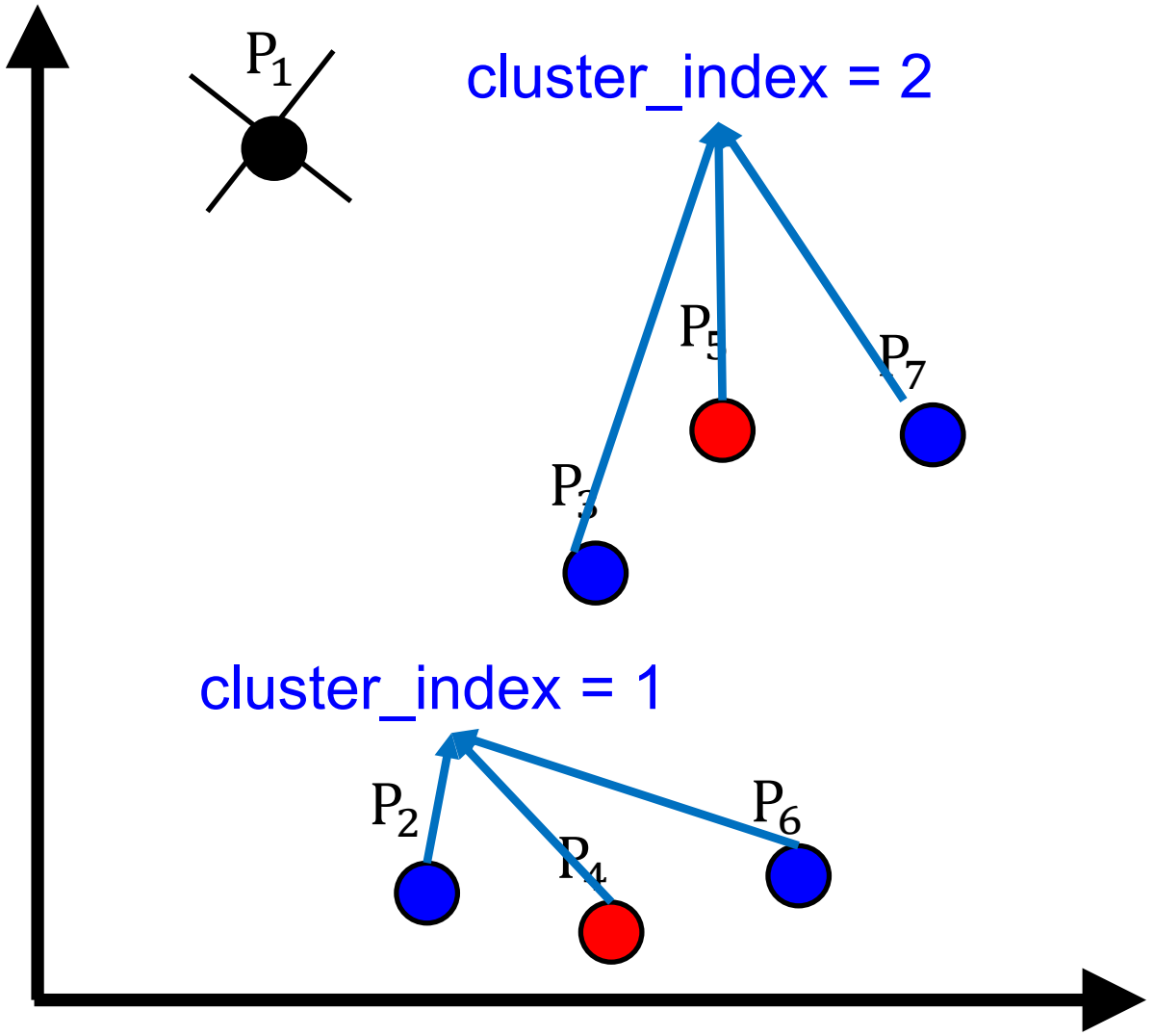


Illustration of a more complex example

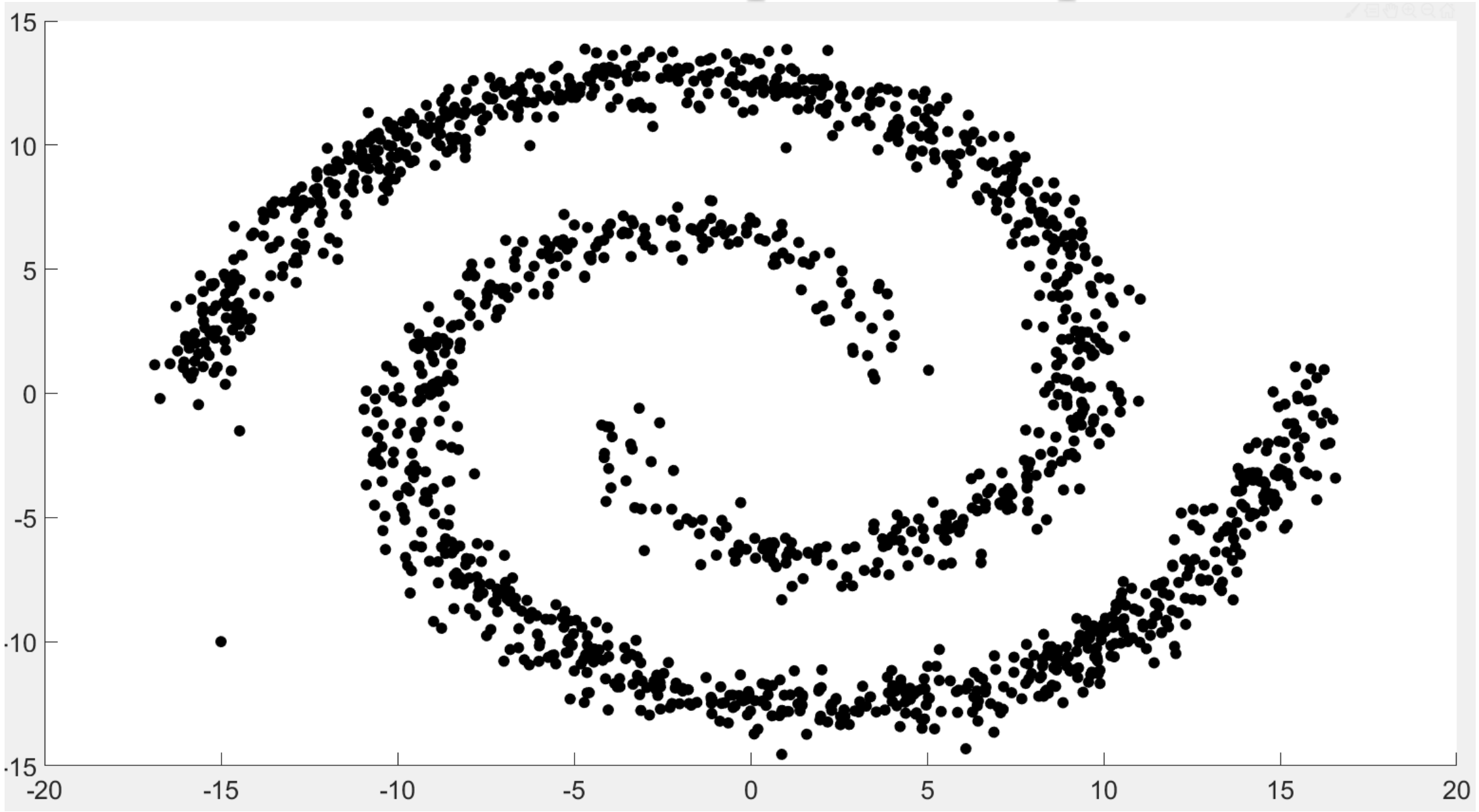
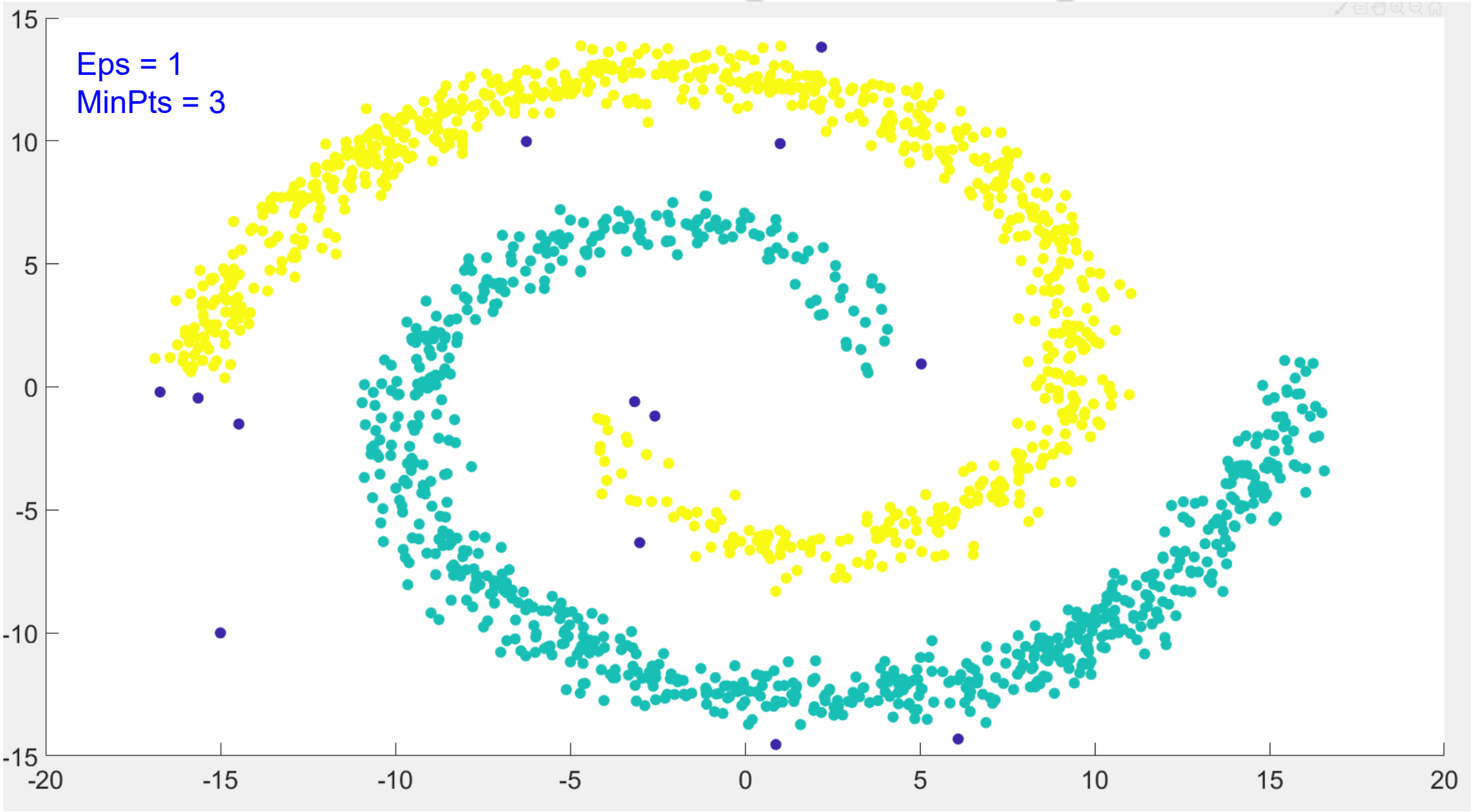
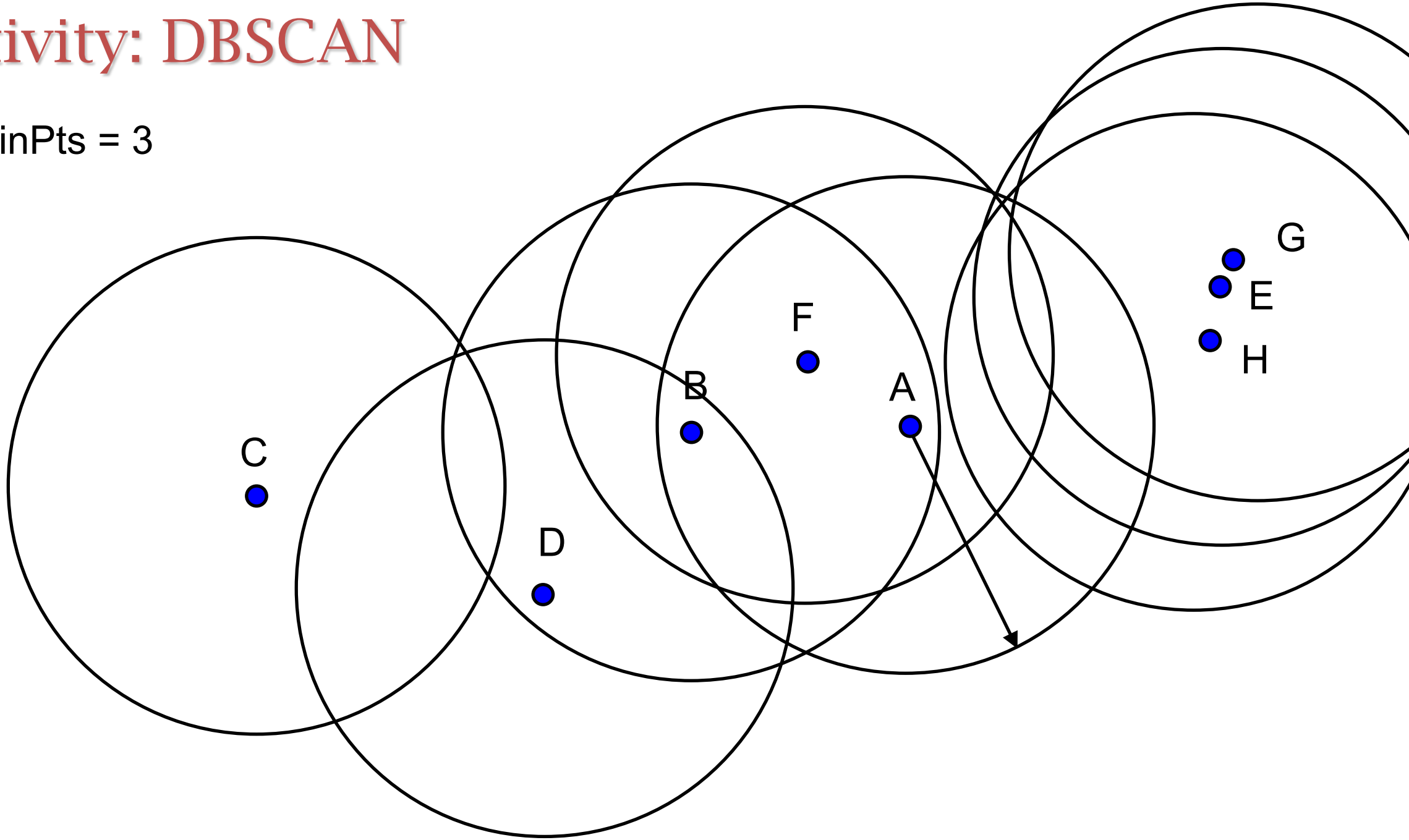


Illustration of a more complex example

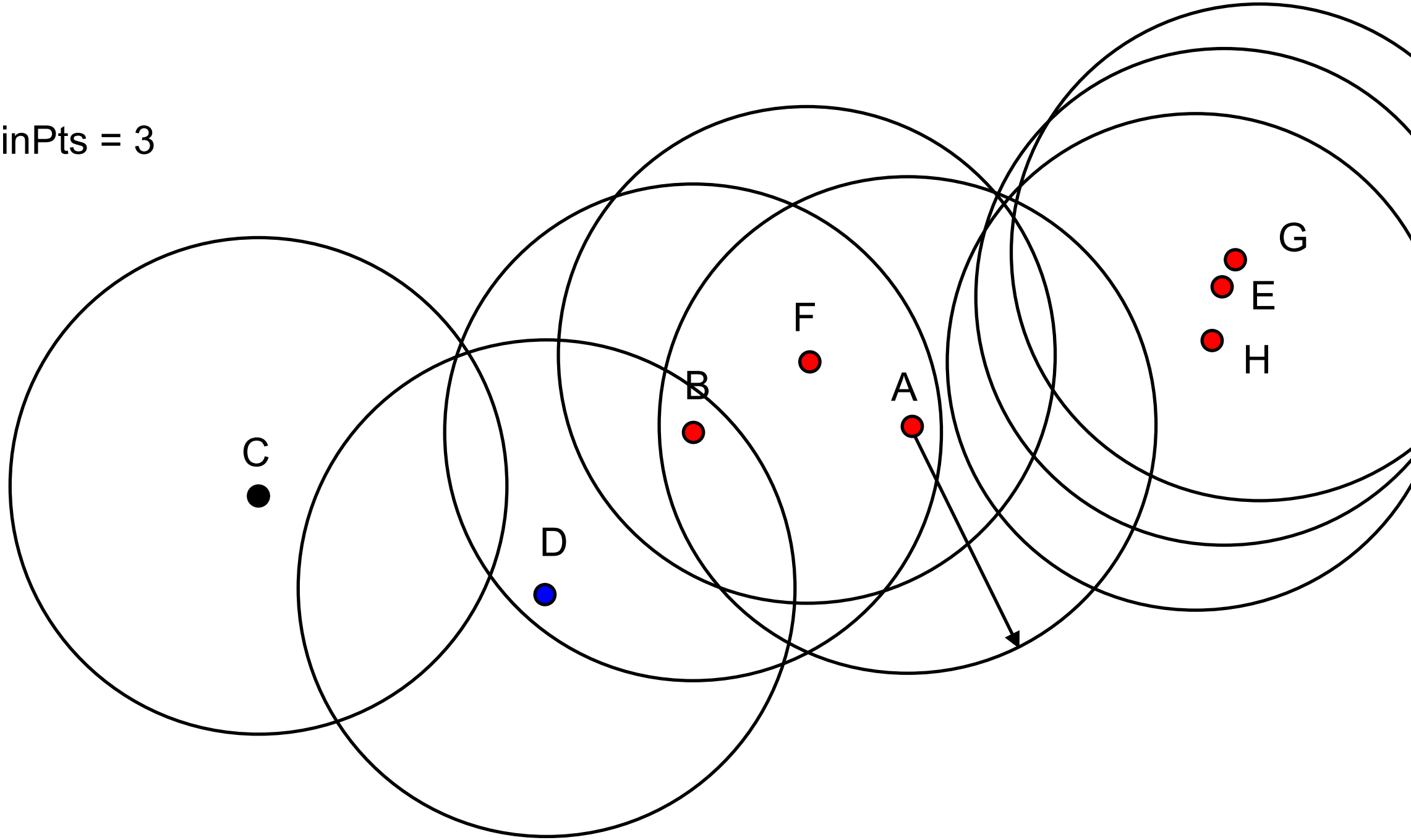


Activity: DBSCAN

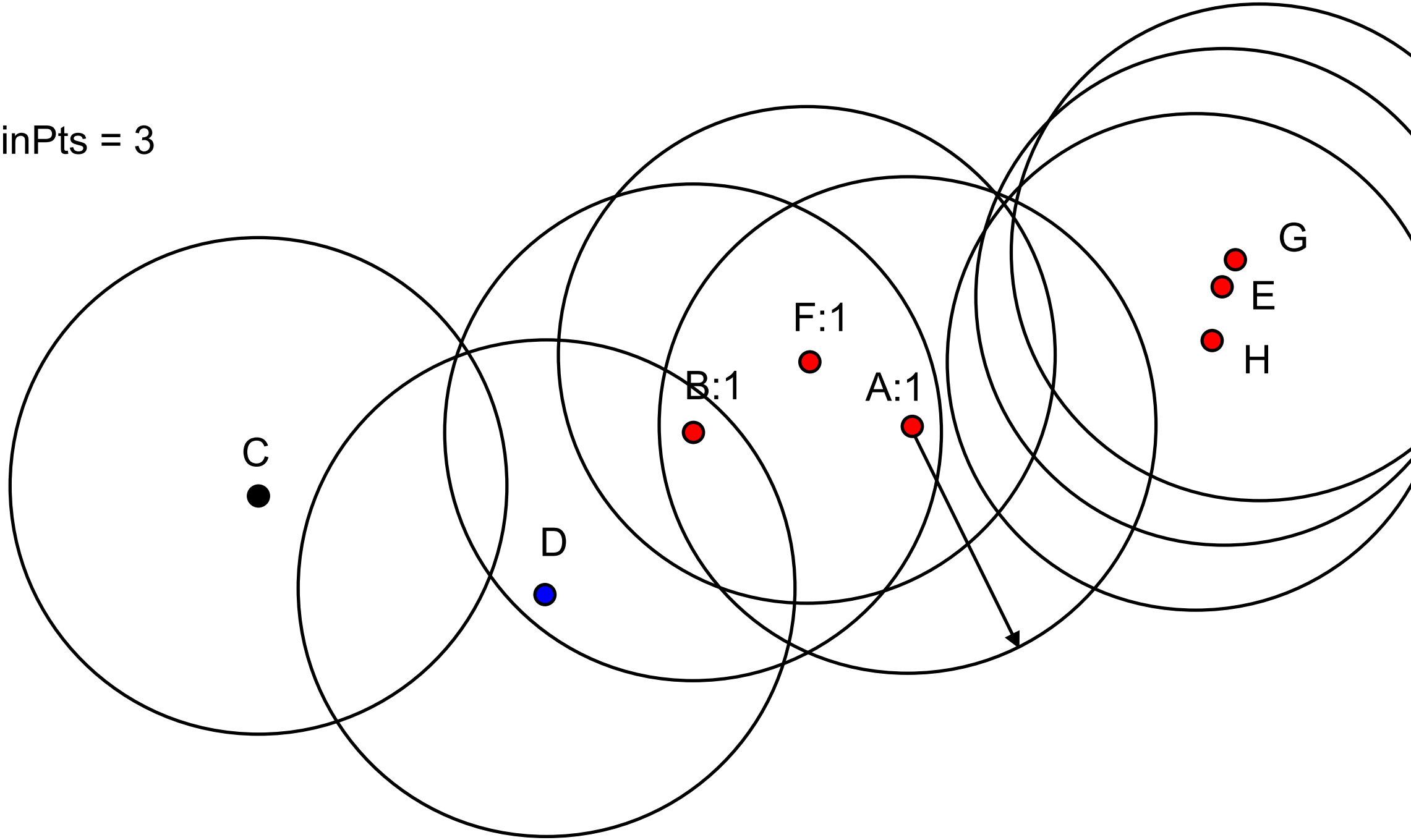
MinPts = 3



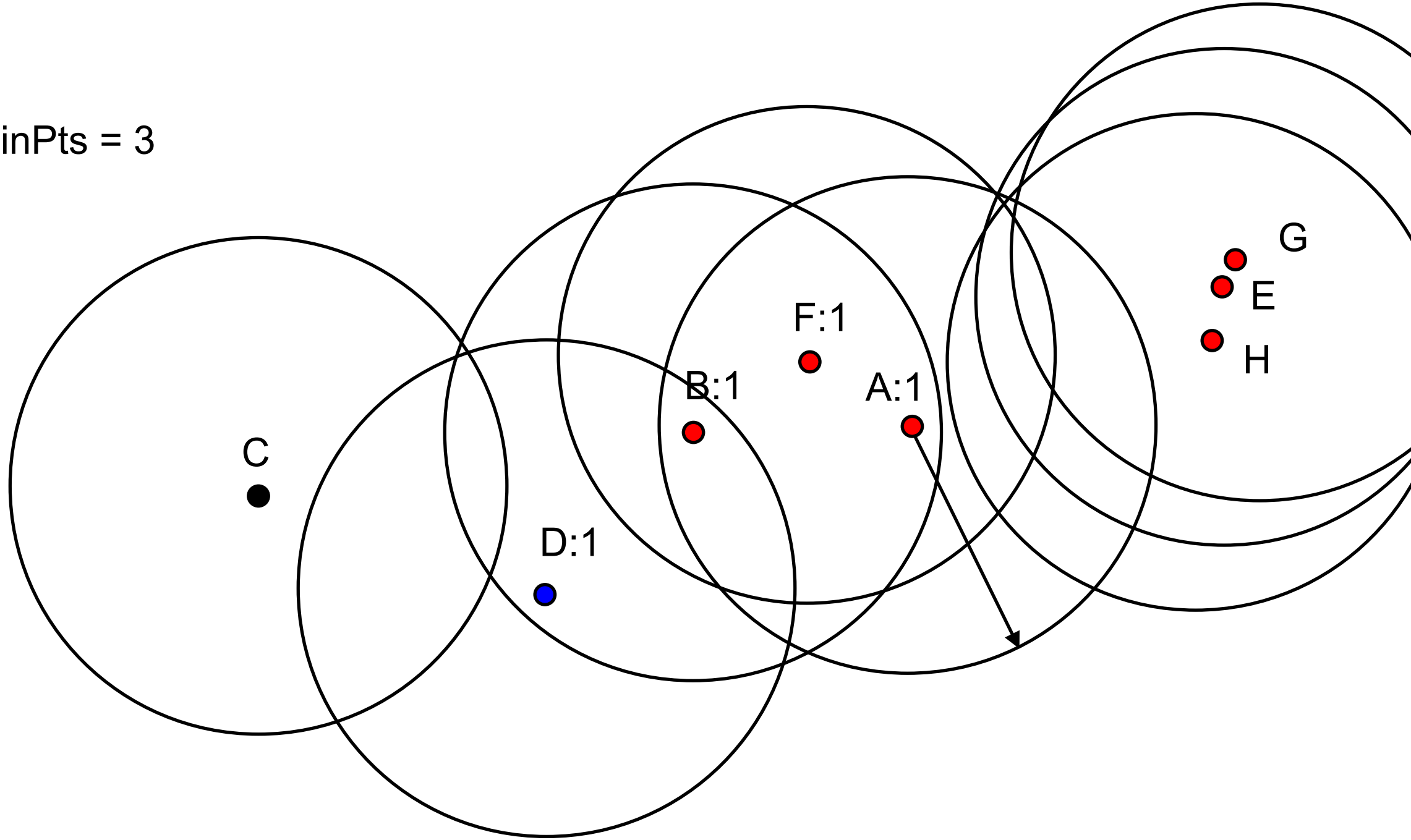
MinPts = 3



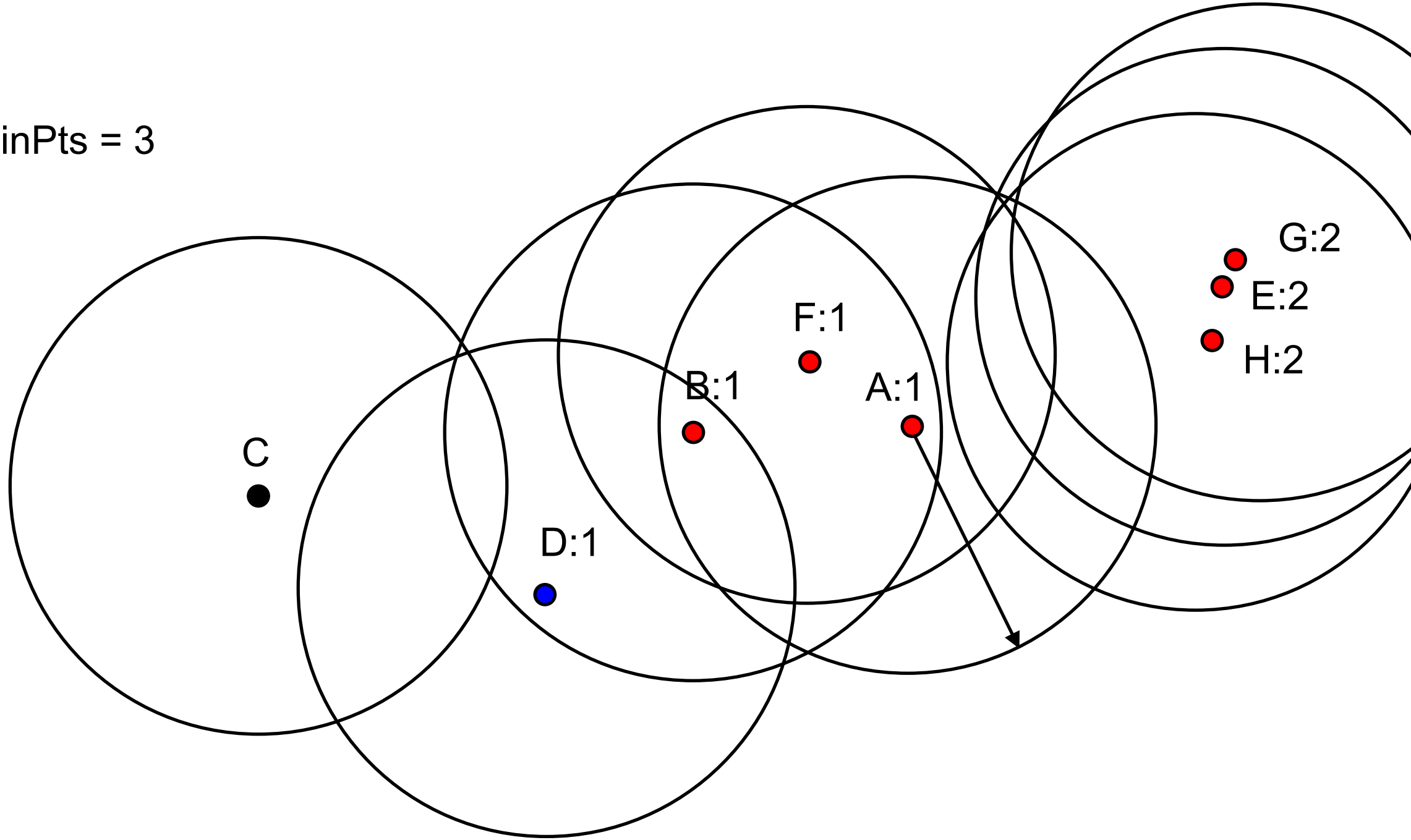
MinPts = 3



MinPts = 3



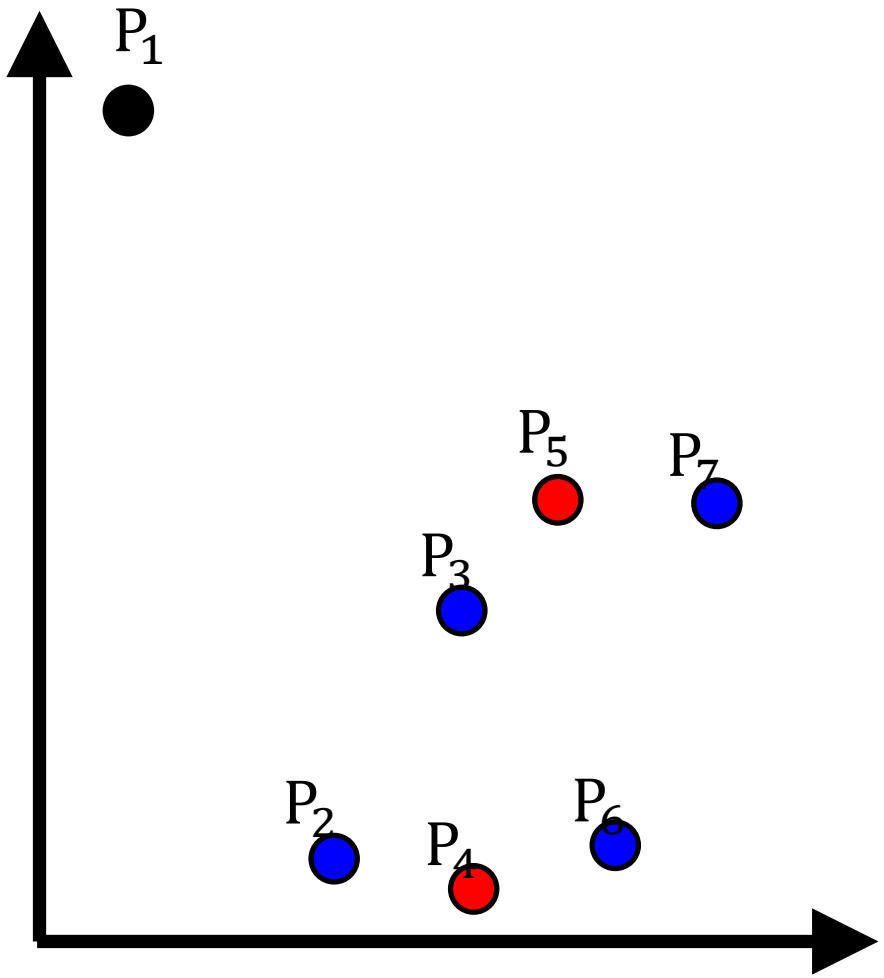
MinPts = 3



How to determine Eps and MinPts

One recommended Elbow method:

- Fix MinPts to be *k*, (e.g., k=4)
- Calculate all points' distances to their $(k\text{-}1)^{th}$ nearest point
- Sort the distance in ascending order and plot them
- Find the "elbow" point, whose corresponding distance is Eps



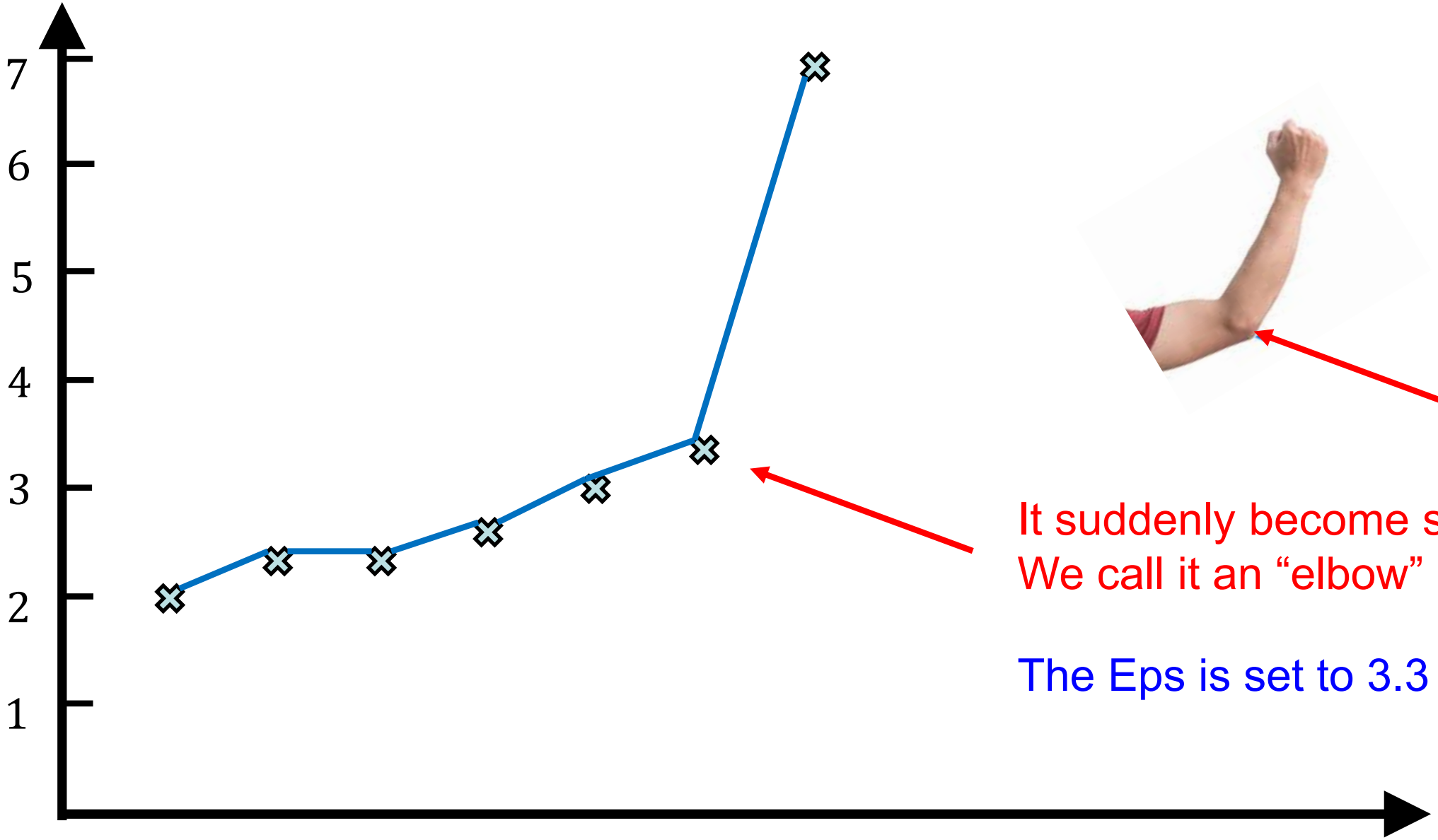
Set k = 4

Point	Distance to the 3rd nearest point
P_1	7.0
P_2	2.3
P_3	2.3
P_4	2.0
P_5	3.0
P_6	2.6
P_7	3.3

After sorting: 2.0, 2.3, 2.3, 2.6, 3.0, 3.3, 7.0

Plot (1, 2.0), (2, 2.3), (3, 2.3), (4, 2.6), (5, 3.0), (6, 3.3), (7, 7.0) into x-y axis

Plot (1, 2.0), (2, 2.3), (3, 2.3), (4, 2.6), (5, 3.0), (6, 3.3), (7, 7.0) into x-y axis

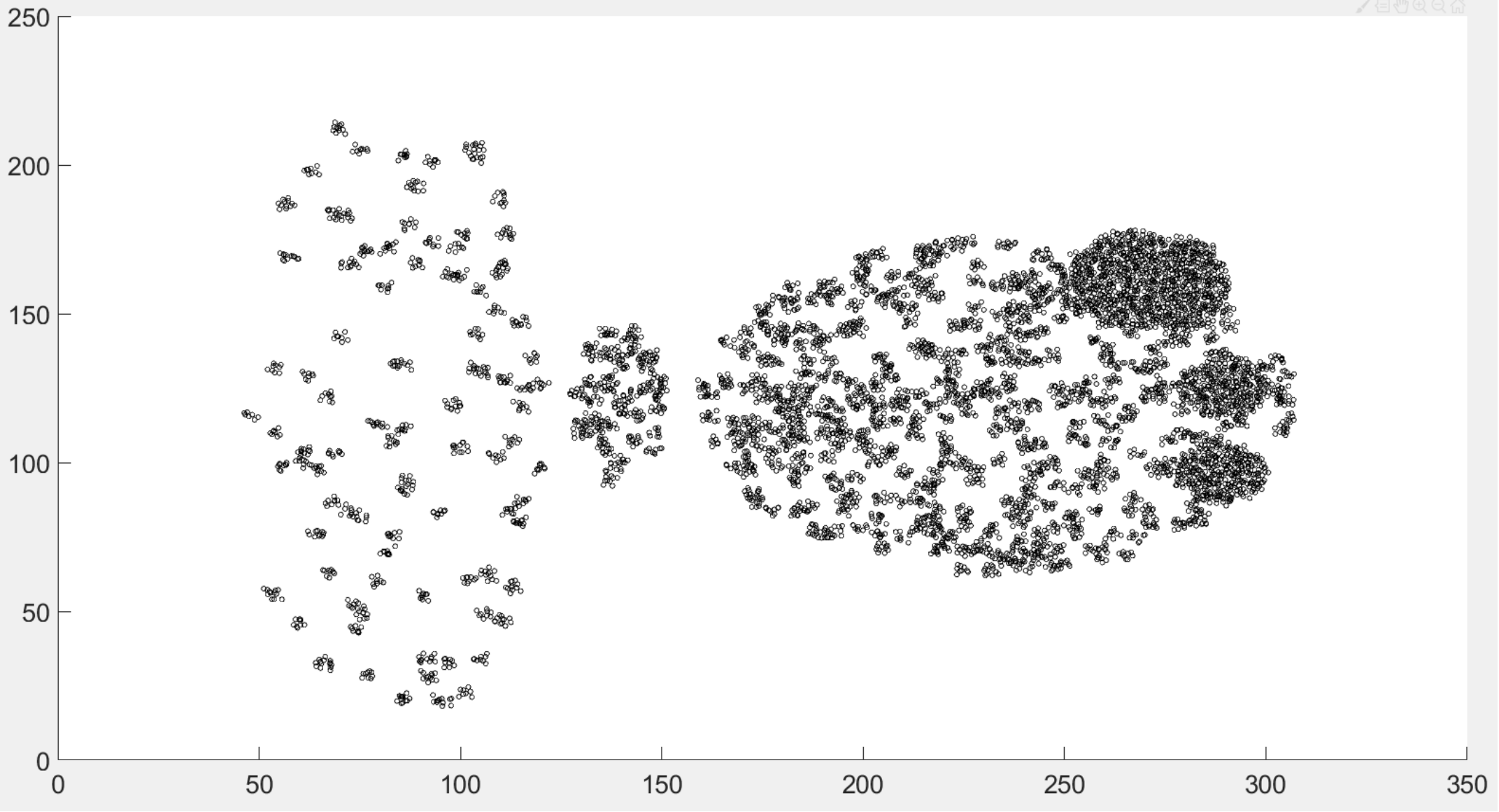


elbow

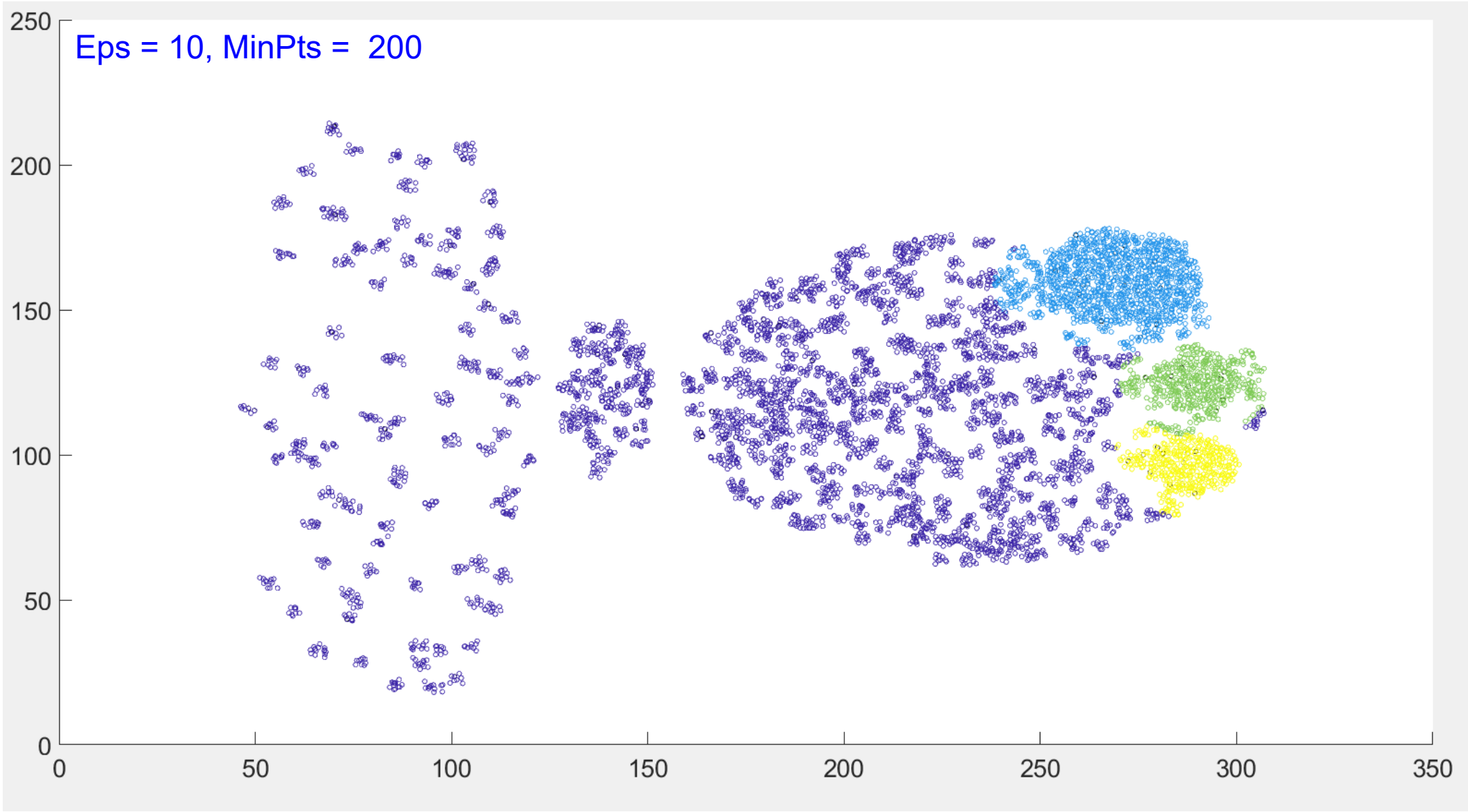
It suddenly become sharp here.
We call it an “elbow” point.

The Eps is set to 3.3 here!

NFL theorem: limitation of DBSCAN



NFL theorem: limitation of DBSCAN

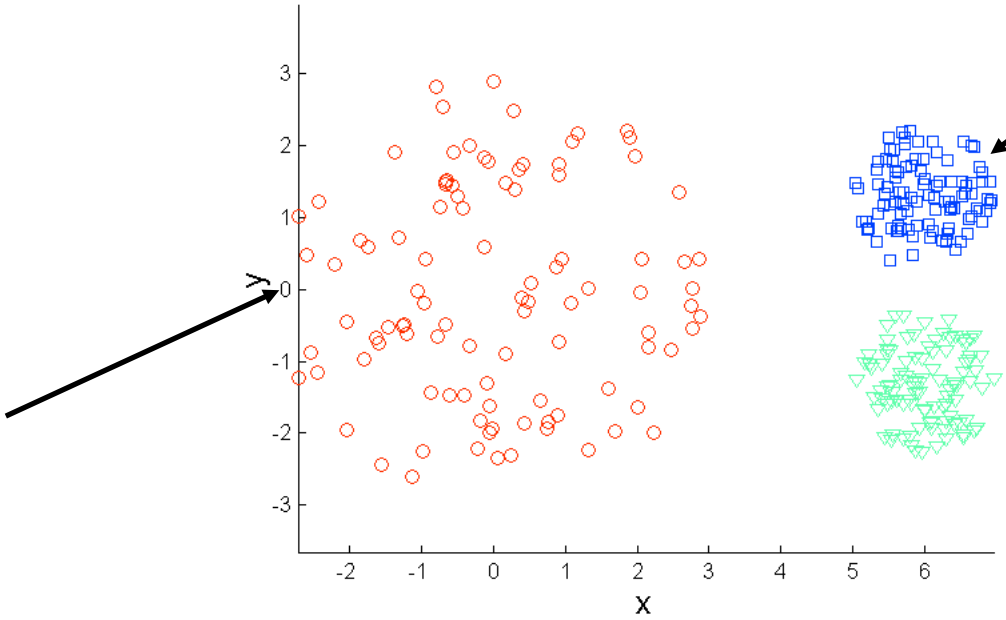


Outline

- Prototype-based clustering
  - k-means
- Density-based clustering
  - DBSCAN
- Hierarchical-based clustering
  - ??

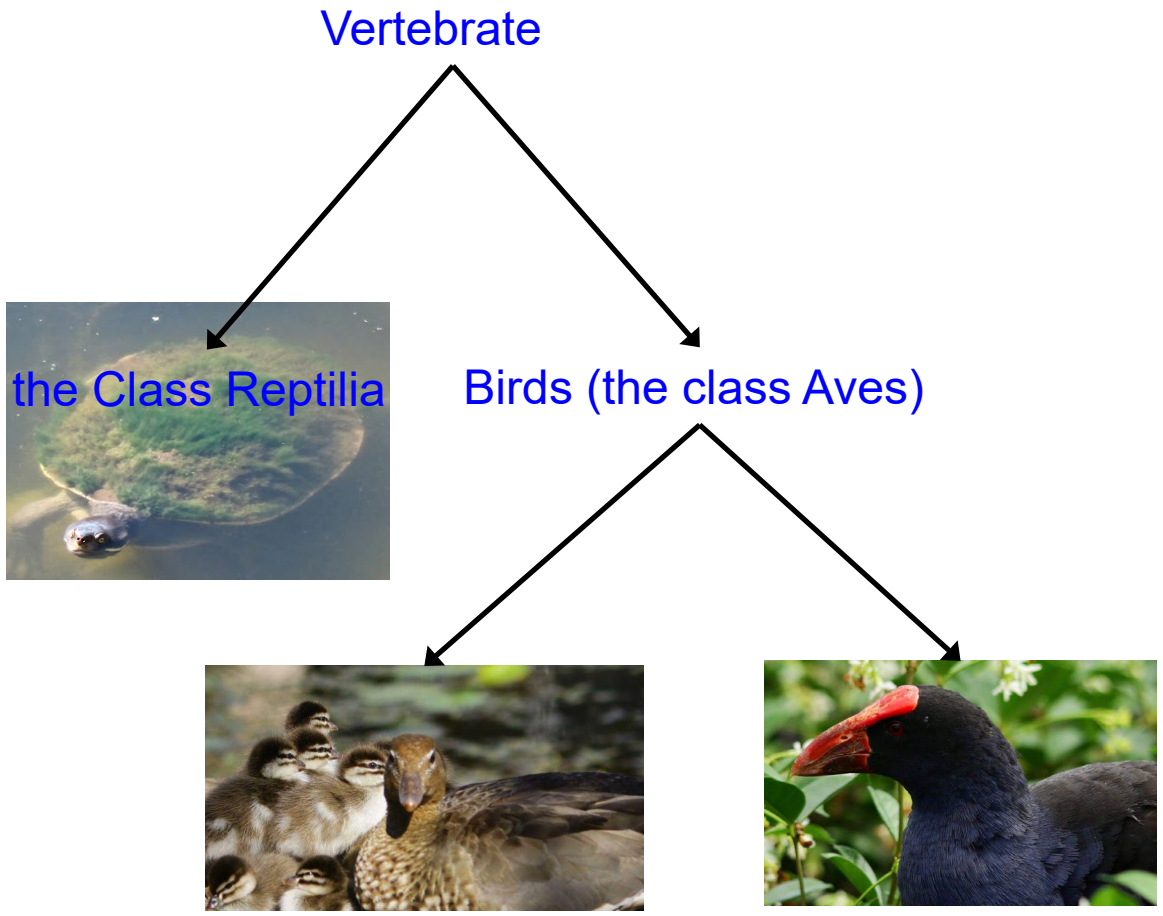
Consider the case

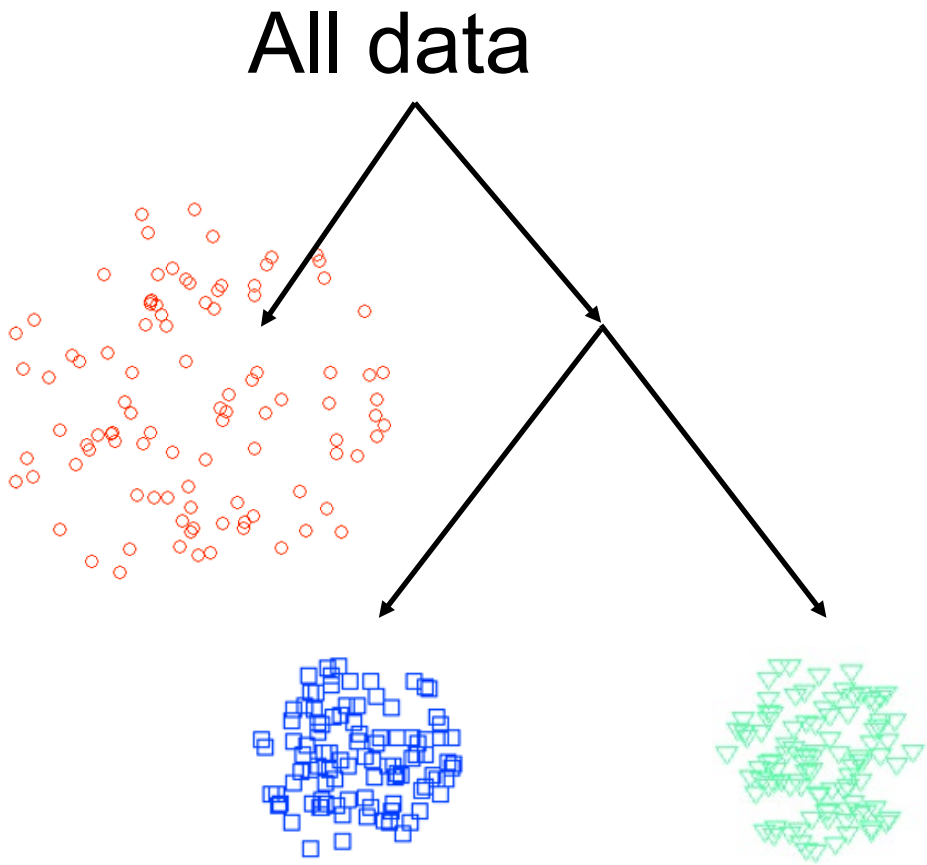
If you collect the wildlife data around the UQ lake, by recording their positions:



Are you satisfied with the clustering result?

Another clustering result

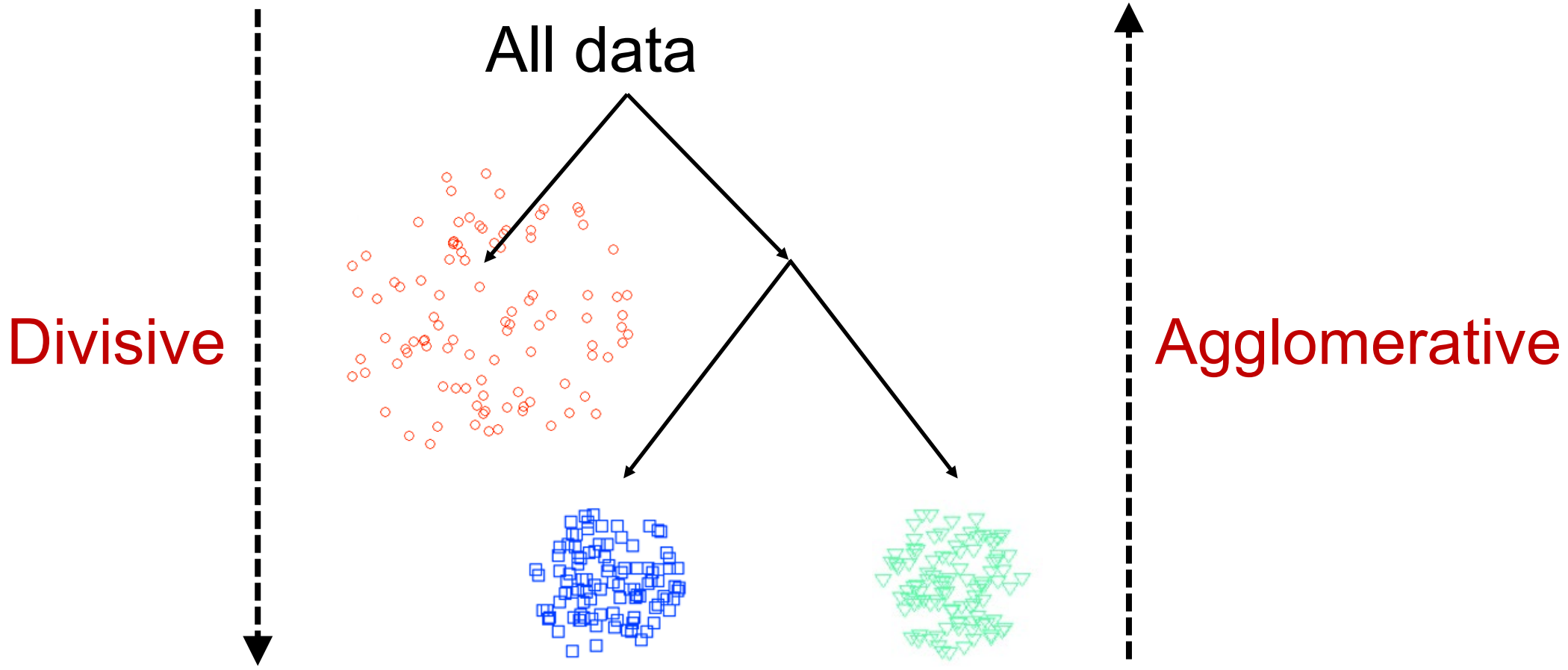




Hierarchical clustering

Two main types

- Agglomerative:
  - Start with the **points** as individual clusters
  - At each step, **merge** the closest pair of clusters until only one cluster (or k clusters) left
- Divisive:
  - Start with **one, all-inclusive** cluster
  - At each step, **split** a cluster until each cluster contains an individual point (or there are k clusters)


All data
Divisive
Agglomerative

AGNES

AGglomerative NESting

1. Compute the *distance matrix*
2. Let *each* data point be a cluster

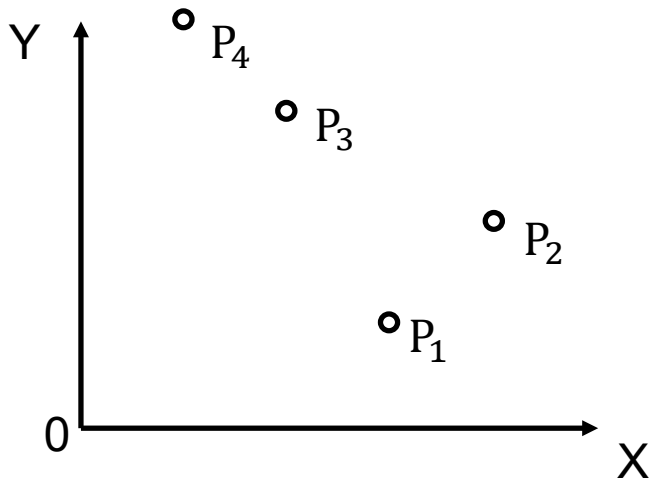
Repeat

3. *merge* the two closest clusters
4. *update* the distance matrix

Until only a *single* cluster remains

An illustration of ANGES

Example:



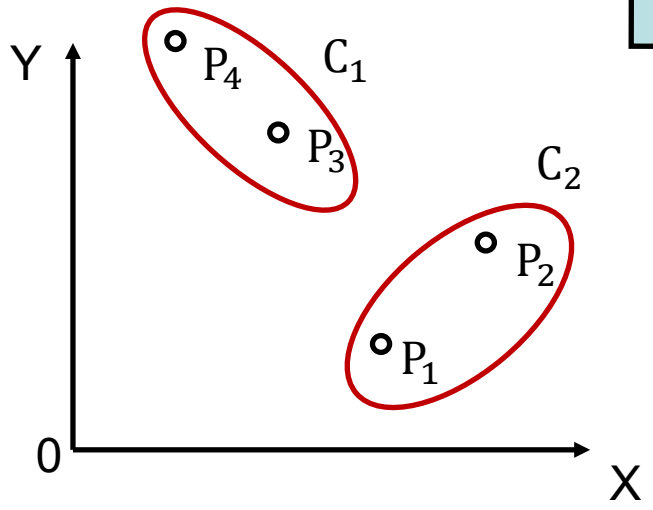
Point	X	Y
P_1	3	1
P_2	4	2
P_3	2	3
P_4	1	4

Step 1: Computer the distance matrix

Step 2: Let each point be a cluster

	P_1	P_2	P_3	P_4
P_1	0	$\sqrt{2}$	$\sqrt{5}$	$\sqrt{13}$
P_2	$\sqrt{2}$	0	$\sqrt{5}$	$\sqrt{13}$
P_3	$\sqrt{5}$	$\sqrt{5}$	0	$\sqrt{2}$
P_4	$\sqrt{13}$	$\sqrt{13}$	$\sqrt{2}$	0

Step 3: Merge the two closest clusters



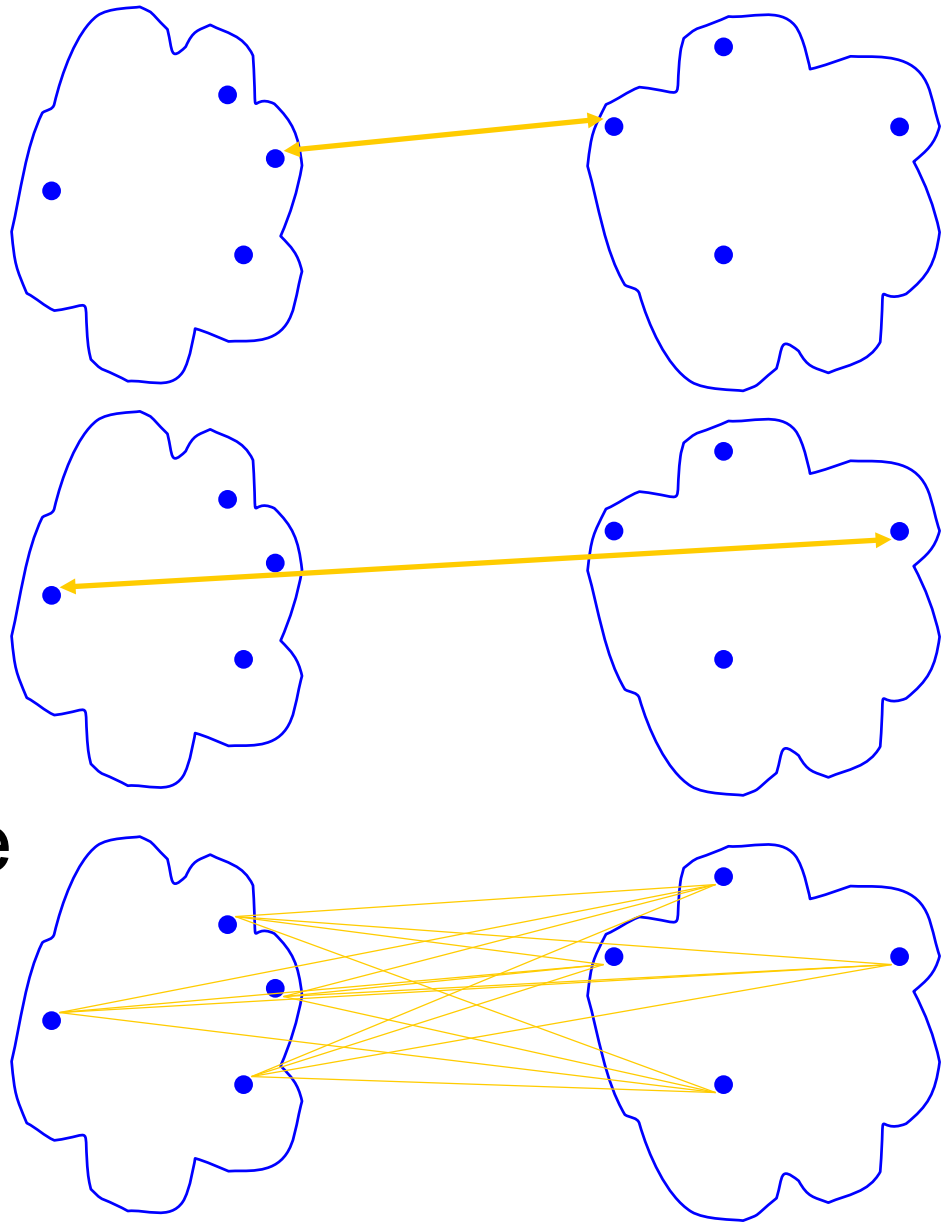
Step 4: Update the distance matrix

	C_1	C_2
C_1	0	?
C_2	?	0

How to update?

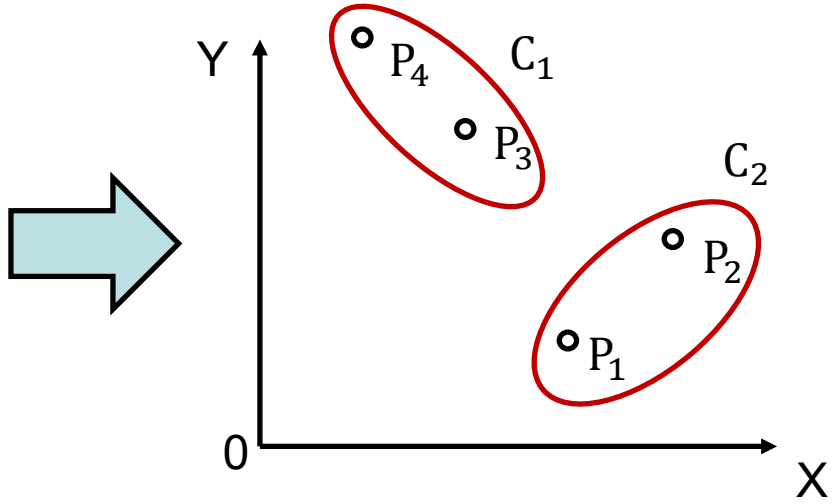
Ways to update the distance matrix

- Min – Single linkage
- Max – Complete linkage
- Average – Average linkage



Update the distance matrix: Min

	P_1	P_2	P_3	P_4
P_1	0	$\sqrt{2}$	$\sqrt{5}$	$\sqrt{13}$
P_2	$\sqrt{2}$	0	$\sqrt{5}$	$\sqrt{13}$
P_3	$\sqrt{5}$	$\sqrt{5}$	0	$\sqrt{2}$
P_4	$\sqrt{13}$	$\sqrt{13}$	$\sqrt{2}$	0



	C_1	C_2
C_1	0	$\sqrt{5}$
C_2	$\sqrt{5}$	0

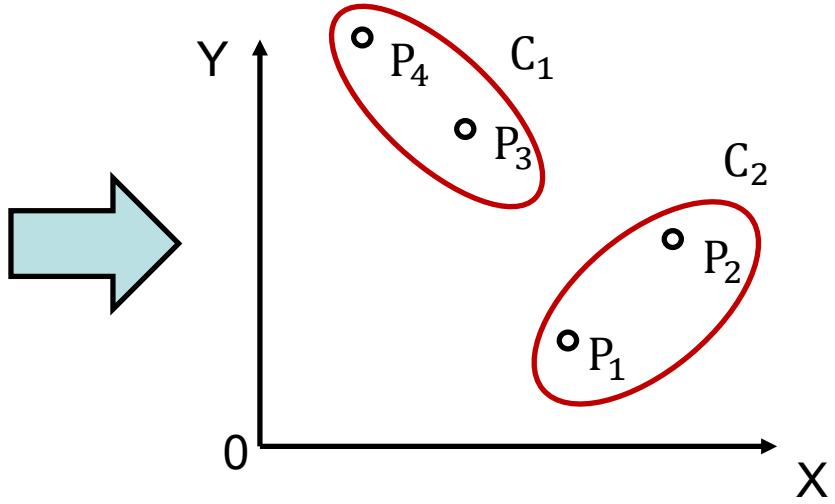
$$dist(C_1, C_2) = \min_{P_i \in C_1, P_j \in C_2} dist(P_i, P_j)$$

$$dist(C_1, C_2) = \min\{ dist(P_1, P_3), dist(P_1, P_4), dist(P_2, P_3), dist(P_2, P_4)\}$$

$$= \min\{ \sqrt{5}, \sqrt{13}, \sqrt{5}, \sqrt{13}\} = \sqrt{5}$$

Update the distance matrix: Max

	P_1	P_2	P_3	P_4
P_1	0	$\sqrt{2}$	$\sqrt{5}$	$\sqrt{13}$
P_2	$\sqrt{2}$	0	$\sqrt{5}$	$\sqrt{13}$
P_3	$\sqrt{5}$	$\sqrt{5}$	0	$\sqrt{2}$
P_4	$\sqrt{13}$	$\sqrt{13}$	$\sqrt{2}$	0



	C_1	C_2
C_1	0	?
C_2	?	0

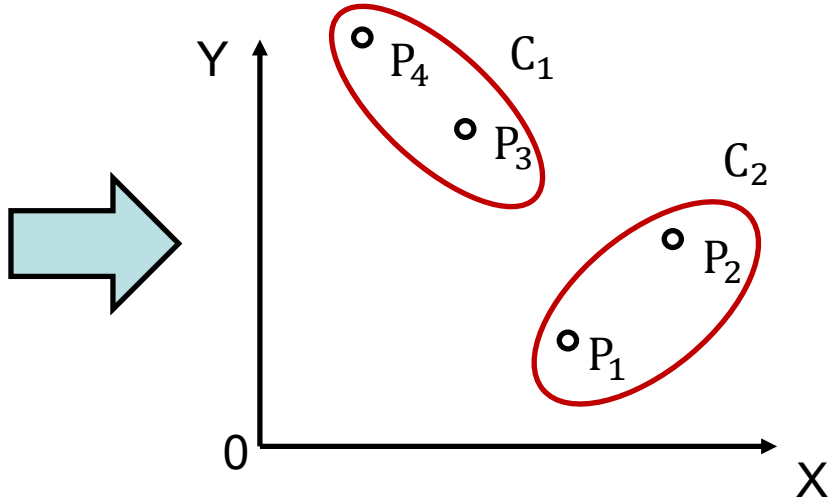
$$dist(C_1, C_2) = \max_{P_i \in C_1, P_j \in C_2} dist(P_i, P_j)$$

$$dist(C_1, C_2) = \max\{dist(P_1, P_3), dist(P_1, P_4), dist(P_2, P_3), dist(P_2, P_4)\}$$

$$= \max\{\sqrt{5}, \sqrt{13}, \sqrt{5}, \sqrt{13}\} = \sqrt{13}$$

Update the distance matrix: Average

	P_1	P_2	P_3	P_4
P_1	0	$\sqrt{2}$	$\sqrt{5}$	$\sqrt{13}$
P_2	$\sqrt{2}$	0	$\sqrt{5}$	$\sqrt{13}$
P_3	$\sqrt{5}$	$\sqrt{5}$	0	$\sqrt{2}$
P_4	$\sqrt{13}$	$\sqrt{13}$	$\sqrt{2}$	0



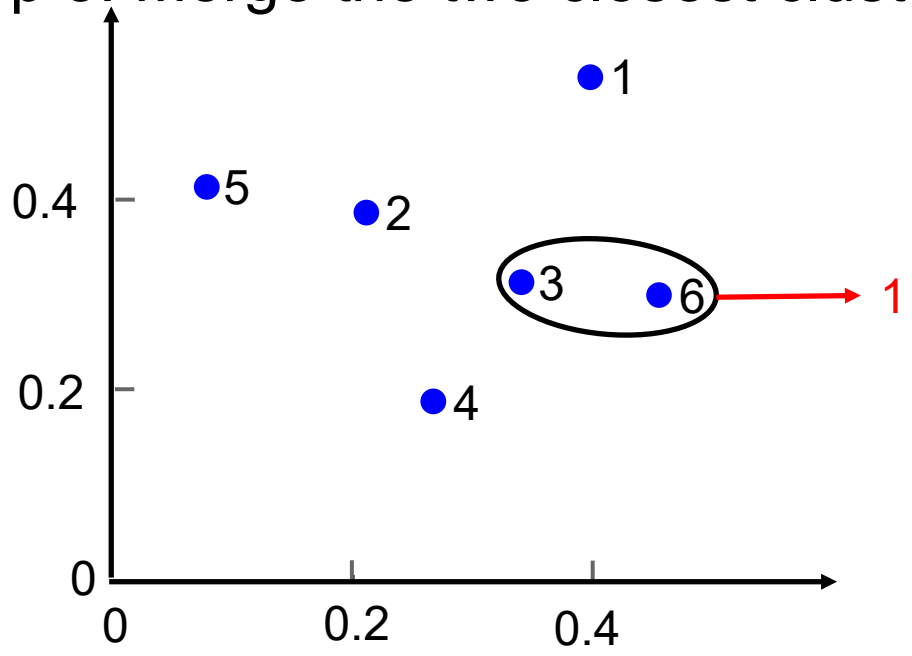
	C_1	C_2
C_1	0	?
C_2	?	0

$$dist(C_1, C_2) = \underset{P_i \in C_1, P_j \in C_2}{\text{average}} \; dist(P_i, P_j)$$

$$dist(C_1, C_2) = \text{average}\{dist(P_1, P_3), dist(P_1, P_4), dist(P_2, P_3), dist(P_2, P_4)\}$$
$$= \text{average}\{\sqrt{5}, \sqrt{13}, \sqrt{5}, \sqrt{13}\} = \quad (\sqrt{13} + \sqrt{5})/2$$

Result on Min: round 1

Step 3: Merge the two closest clusters



Distance Matrix:

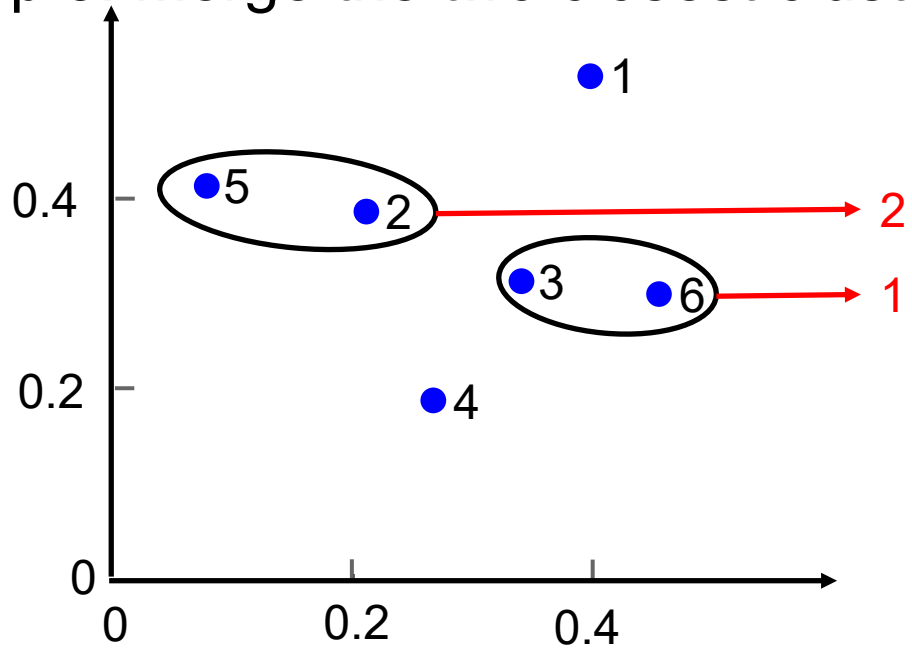
	P_1	P_2	P_3	P_4	P_5	P_6
P_1	0.00	0.24	0.22	0.37	0.34	0.23
P_2	0.24	0.00	0.15	0.20	0.14	0.25
P_3	0.22	0.15	0.00	0.15	0.28	0.11
P_4	0.37	0.20	0.15	0.00	0.29	0.22
P_5	0.34	0.14	0.28	0.29	0.00	0.39
P_6	0.23	0.25	0.11	0.22	0.39	0.00

Step 4: Update the distance matrix

	P_1	P_2	P_3, P_6	P_4	P_5
P_1	0.00	0.24	0.22	0.37	0.34
P_2	0.24	0.00	0.15	0.20	0.14
P_3, P_6	0.22	0.15	0.00	0.15	0.28
P_4	0.37	0.20	0.15	0.00	0.29
P_5	0.34	0.14	0.28	0.29	0.00

Result on Min: round 2

Step 3: Merge the two closest clusters



Distance Matrix:

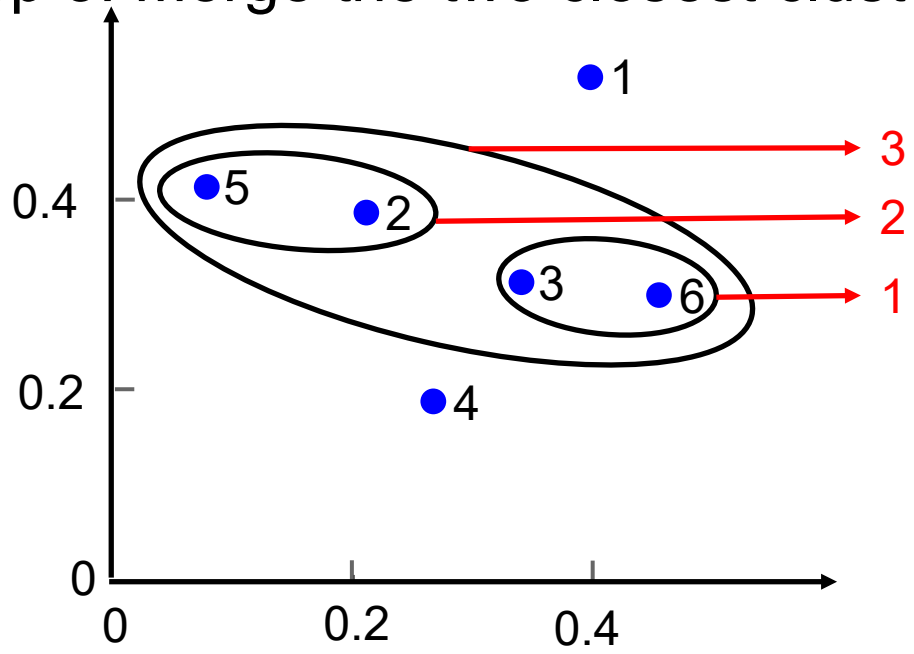
	P_1	P_2	P_3, P_6	P_4	P_5
P_1	0.00	0.24	0.22	0.37	0.34
P_2	0.24	0.00	0.15	0.20	0.14
P_3, P_6	0.22	0.15	0.00	0.15	0.28
P_4	0.37	0.20	0.15	0.00	0.29
P_5	0.34	0.14	0.28	0.29	0.00

Step 4: Update the distance matrix

	P_1	P_2, P_5	P_3, P_6	P_4
P_1	0.00	0.24	0.22	0.37
P_2, P_5	0.24	0.00	0.15	0.20
P_3, P_6	0.22	0.15	0.00	0.15
P_4	0.37	0.20	0.15	0.00

Result on Min: round 3

Step 3: Merge the two closest clusters



Distance Matrix:

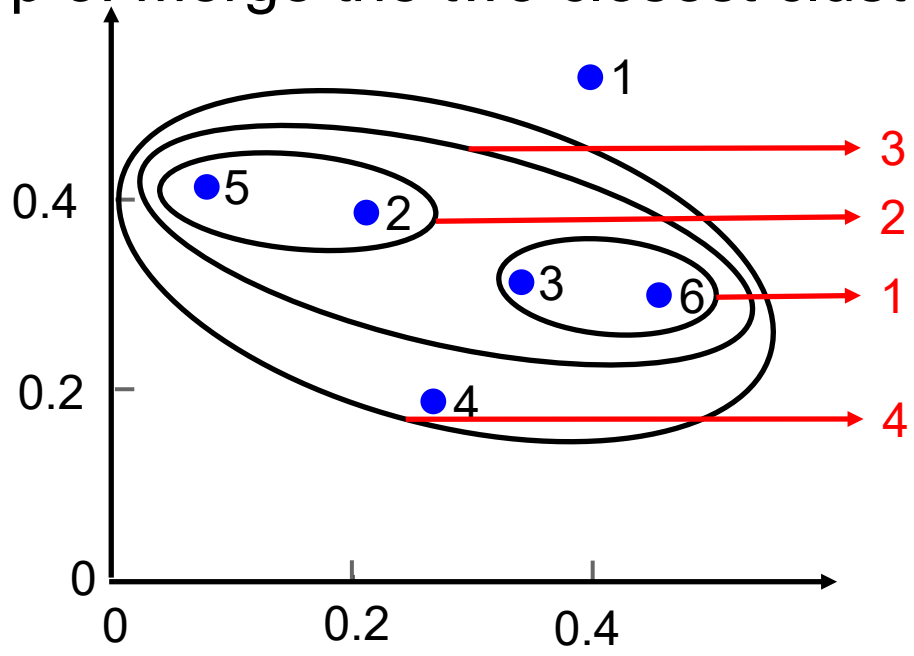
	P_1	P_2, P_5	P_3, P_6	P_4
P_1	0.00	0.24	0.22	0.37
P_2, P_5	0.24	0.00	0.15	0.20
P_3, P_6	0.22	0.15	0.00	0.15
P_4	0.37	0.20	0.15	0.00

Step 4: Update the distance matrix

	P_1	P_2, P_5, P_3, P_6	P_4
P_1	0.00	0.22	0.37
P_2, P_5, P_3, P_6	0.22	0.00	0.15
P_4	0.37	0.15	0.00

Result on Min: round 4

Step 3: Merge the two closest clusters



Distance Matrix:

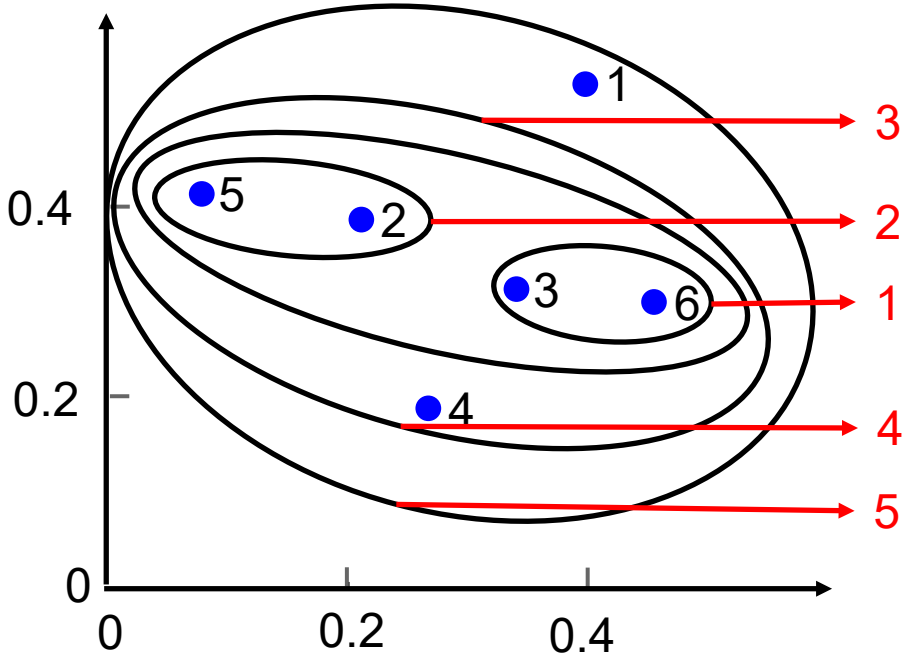
	P_1	P_2, P_5, P_3, P_6	P_4
P_1	0.00	0.22	0.37
P_2, P_5, P_3, P_6	0.22	0.00	0.15
P_4	0.37	0.15	0.00

Step 4: Update the distance matrix

	P_1	P_2, P_5, P_3, P_6, P_4
P_1	0.00	0.22
P_2, P_5, P_3, P_6, P_4	0.22	0.00

Result on Min: round 5

Step 3: Merge the two closest clusters

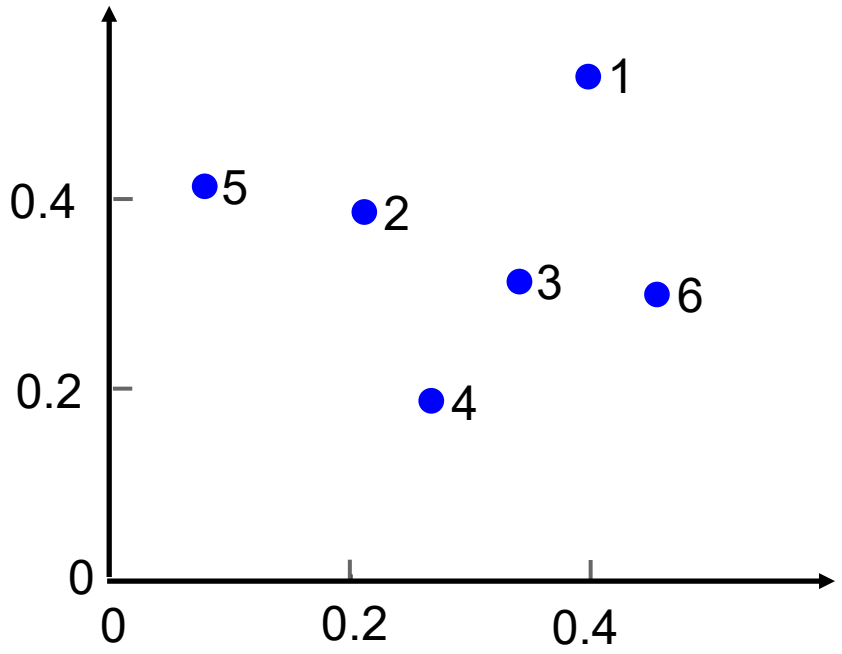


Distance Matrix:

	P_1	P_2, P_5, P_3, P_6, P_4
P_1	0.00	0.22
P_2, P_5, P_3, P_6, P_4	0.22	0.00

Activity

- The hierarchical clustering result by ANGES of the following data, using Max and Average:

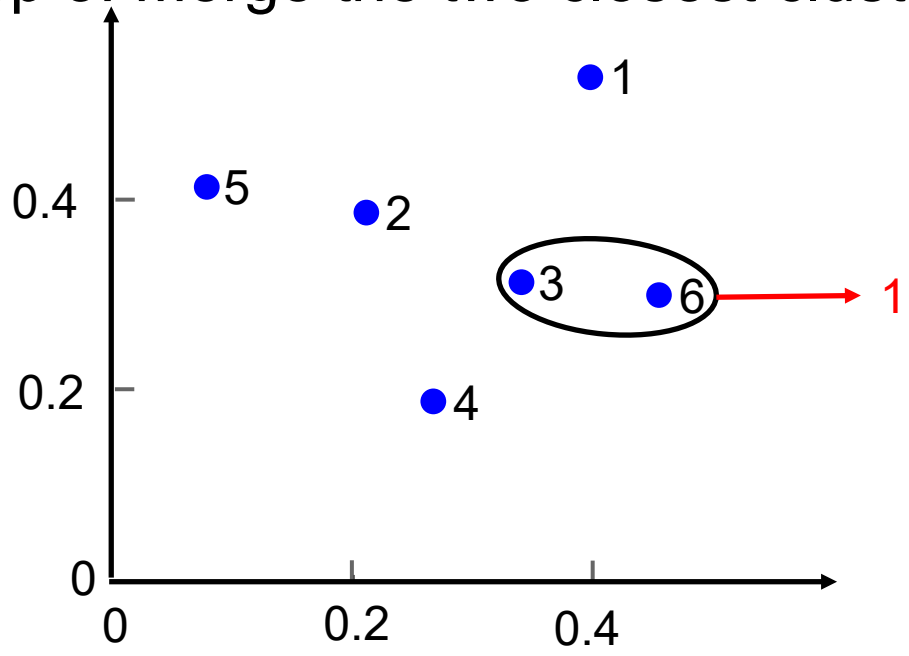


Distance Matrix:

	P_1	P_2	P_3	P_4	P_5	P_6
P_1	0.00	0.24	0.22	0.37	0.34	0.23
P_2	0.24	0.00	0.15	0.20	0.14	0.25
P_3	0.22	0.15	0.00	0.15	0.28	0.11
P_4	0.37	0.20	0.15	0.00	0.29	0.22
P_5	0.34	0.14	0.28	0.29	0.00	0.39
P_6	0.23	0.25	0.11	0.22	0.39	0.00

Result on Max: round 1

Step 3: Merge the two closest clusters



Distance Matrix:

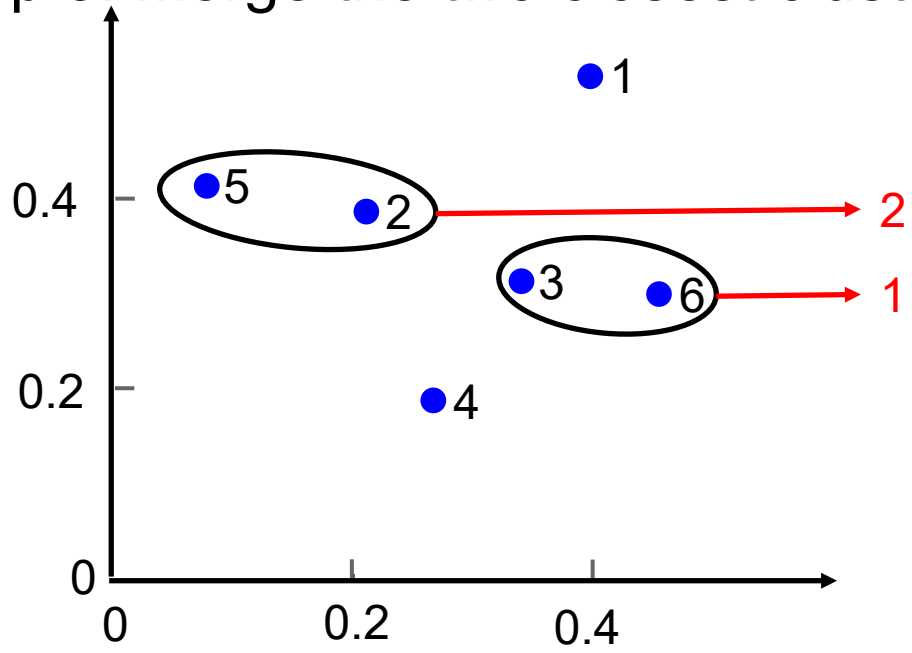
	P_1	P_2	P_3	P_4	P_5	P_6
P_1	0.00	0.24	0.22	0.37	0.34	0.23
P_2	0.24	0.00	0.15	0.20	0.14	0.25
P_3	0.22	0.15	0.00	0.15	0.28	0.11
P_4	0.37	0.20	0.15	0.00	0.29	0.22
P_5	0.34	0.14	0.28	0.29	0.00	0.39
P_6	0.23	0.25	0.11	0.22	0.39	0.00

Step 4: Update the distance matrix

	P_1	P_2	P_3, P_6	P_4	P_5
P_1	0.00	0.24	0.23	0.37	0.34
P_2	0.24	0.00	0.25	0.20	0.14
P_3, P_6	0.23	0.25	0.00	0.22	0.39
P_4	0.37	0.20	0.22	0.00	0.29
P_5	0.34	0.14	0.39	0.29	0.00

Result on Max: round 2

Step 3: Merge the two closest clusters



Distance Matrix:

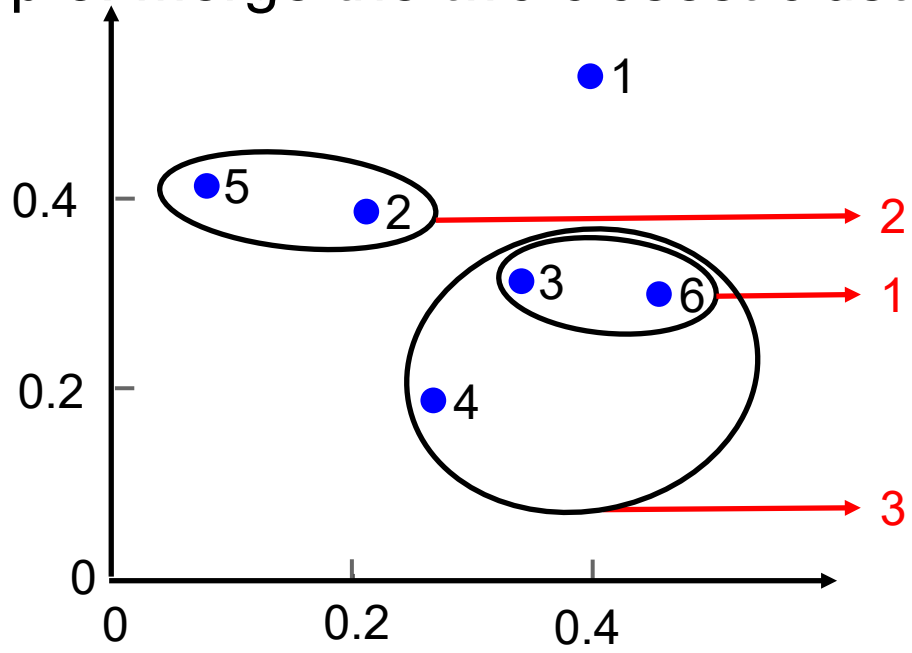
	P_1	P_2	P_3, P_6	P_4	P_5
P_1	0.00	0.24	0.23	0.37	0.34
P_2	0.24	0.00	0.25	0.20	0.14
P_3, P_6	0.23	0.25	0.00	0.22	0.39
P_4	0.37	0.20	0.22	0.00	0.29
P_5	0.34	0.14	0.39	0.29	0.00

Step 4: Update the distance matrix

	P_1	P_2, P_5	P_3, P_6	P_4
P_1	0.00	0.34	0.23	0.37
P_2, P_5	0.34	0.00	0.39	0.29
P_3, P_6	0.23	0.39	0.00	0.22
P_4	0.37	0.29	0.22	0.00

Result on Max: round 3

Step 3: Merge the two closest clusters



Distance Matrix:

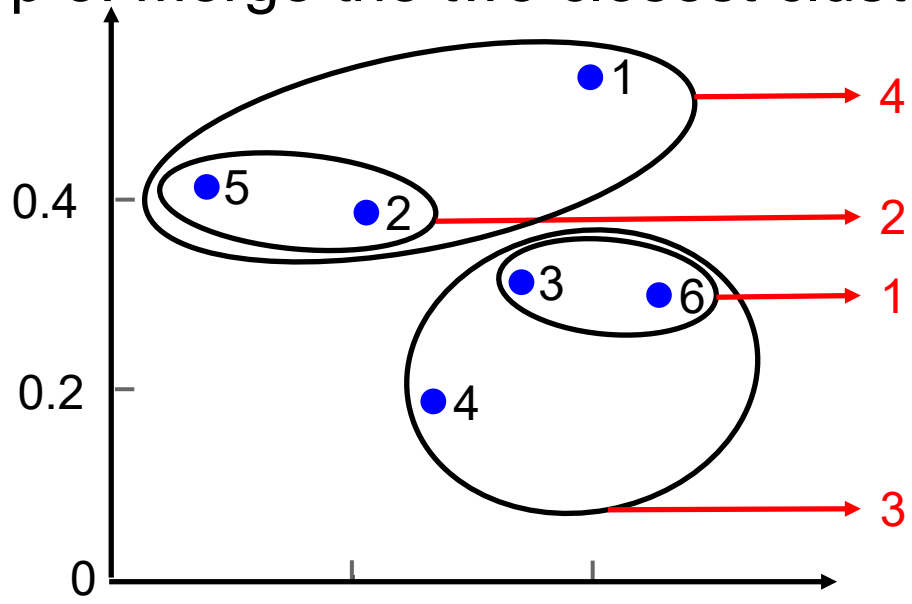
	P_1	P_2, P_5	P_3, P_6	P_4
P_1	0.00	0.34	0.23	0.37
P_2, P_5	0.34	0.00	0.39	0.29
P_3, P_6	0.23	0.39	0.00	0.22
P_4	0.37	0.29	0.22	0.00

Step 4: Update the distance matrix

	P_1	P_2, P_5	P_3, P_6, P_4
P_1	0.00	0.34	0.37
P_2, P_5	0.34	0.00	0.39
P_3, P_6, P_4	0.37	0.39	0.00

Result on Max: round 4

Step 3: Merge the two closest clusters



Distance Matrix:

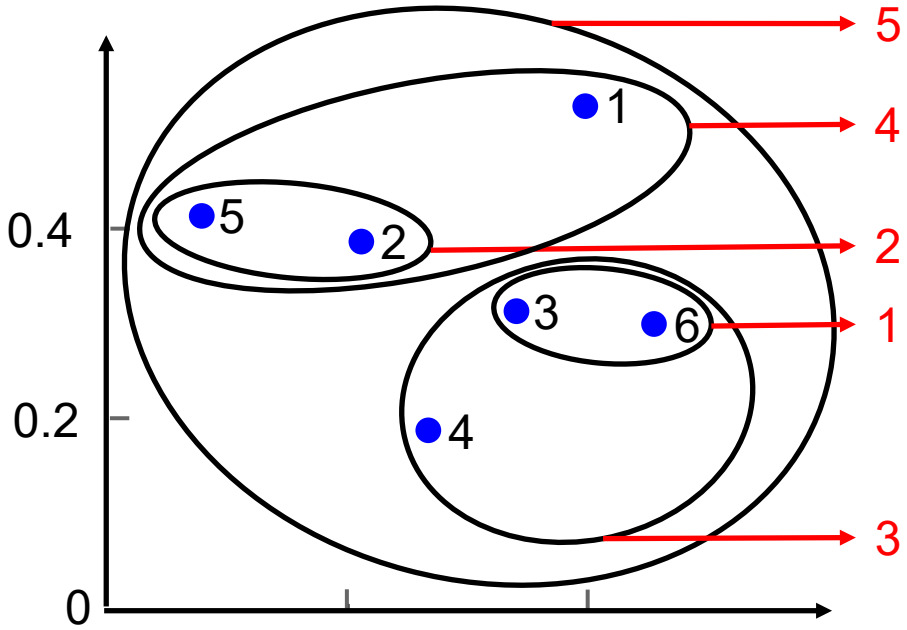
	P_1	P_2, P_5	P_3, P_6, P_4
P_1	0.00	0.34	0.37
P_2, P_5	0.34	0.00	0.39
P_3, P_6, P_4	0.37	0.39	0.00

Step 4: Update the distance matrix

	P_1, P_2, P_5	P_3, P_6, P_4
P_1, P_2, P_5	0.00	0.39
P_3, P_6, P_4	0.39	0.00

Result on Max: round 5

Step 3: Merge the two closest clusters

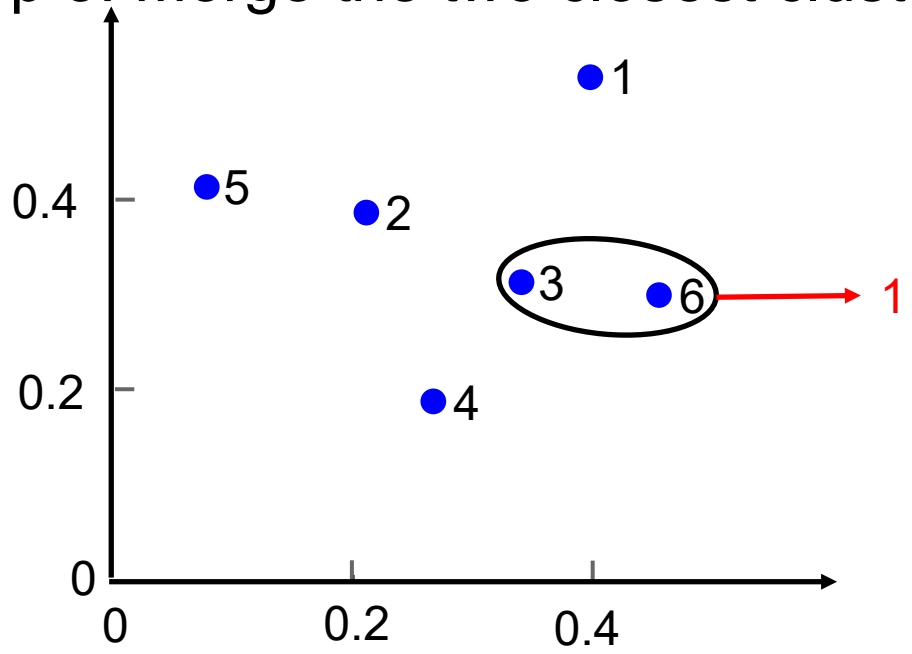


Distance Matrix:

	P_1, P_2, P_5	P_3, P_6, P_4
P_1, P_2, P_5	0.00	0.39
P_3, P_6, P_4	0.39	0.00

Result on Average: round 1

Step 3: Merge the two closest clusters



Distance Matrix:

	P_1	P_2	P_3	P_4	P_5	P_6
P_1	0.00	0.24	0.22	0.37	0.34	0.23
P_2	0.24	0.00	0.15	0.20	0.14	0.25
P_3	0.22	0.15	0.00	0.15	0.28	0.11
P_4	0.37	0.20	0.15	0.00	0.29	0.22
P_5	0.34	0.14	0.28	0.29	0.00	0.39
P_6	0.23	0.25	0.11	0.22	0.39	0.00

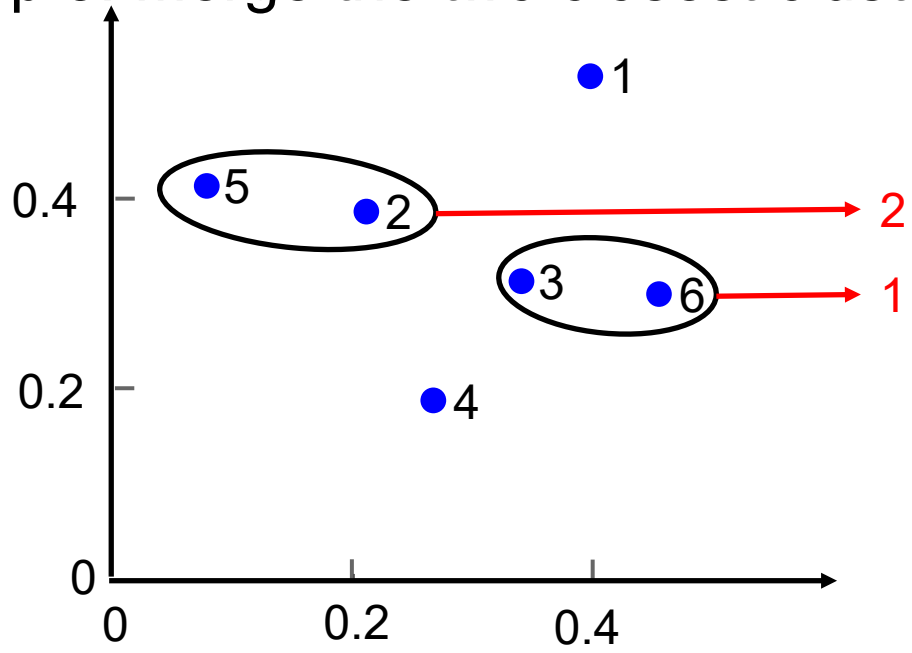
Step 4: Update the distance matrix

$$\text{average}\{dist(P_1, P_3), dist(P_1, P_6)\} = \text{average}\{0.22, 0.23\} = 0.225$$

	P_1	P_2	P_3, P_6	P_4	P_5
P_1	0.00	0.24	0.225	0.37	0.34
P_2	0.24	0.00	0.20	0.20	0.14
P_3, P_6	0.225	0.20	0.00	0.185	0.335
P_4	0.37	0.20	0.185	0.00	0.29
P_5	0.34	0.14	0.335	0.29	0.00

Result on Average: round 2

Step 3: Merge the two closest clusters



Distance Matrix:

	P_1	P_2	P_3, P_6	P_4	P_5
P_1	0.00	0.24	0.225	0.37	0.34
P_2	0.24	0.00	0.20	0.20	0.14
P_3, P_6	0.225	0.20	0.00	0.185	0.335
P_4	0.37	0.20	0.185	0.00	0.29
P_5	0.34	0.14	0.335	0.29	0.00

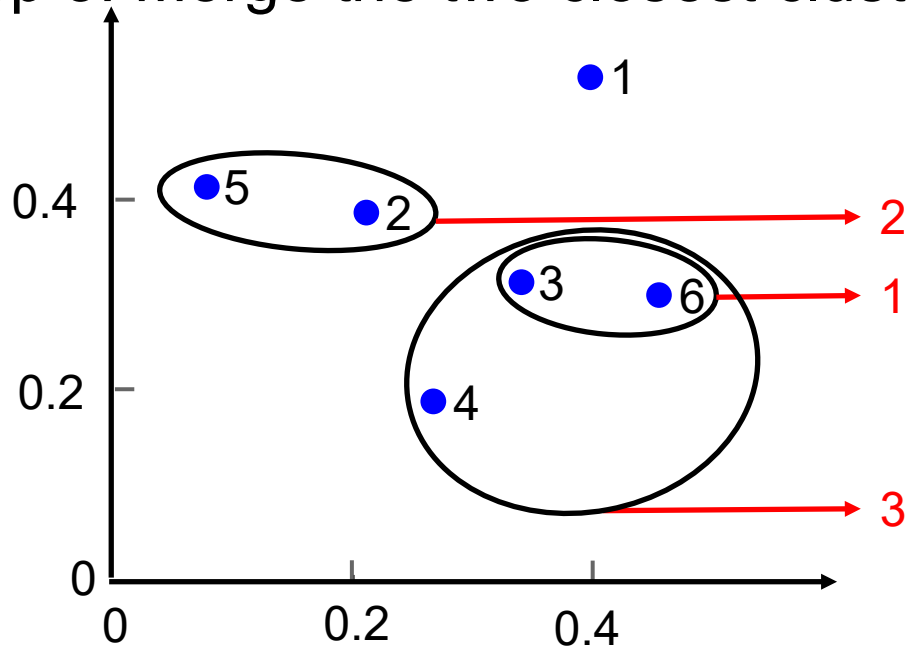
Step 4: Update the distance matrix

$$\text{average}\{dist(P_2, P_3), dist(P_2, P_6), dist(P_5, P_3), dist(P_5, P_6)\}$$
$$= \text{average}\{0.15, 0.25, 0.28, 0.39\} = 0.2675$$

	P_1	P_2, P_5	P_3, P_6	P_4
P_1	0.00	0.29	0.225	0.37
P_2, P_5	0.29	0.00	0.2675	0.245
P_3, P_6	0.225	0.2675	0.00	0.185
P_4	0.37	0.245	0.185	0.00

Result on Average: round 3

Step 3: Merge the two closest clusters



Distance Matrix:

	P_1	P_2, P_5	P_3, P_6	P_4
P_1	0.00	0.29	0.225	0.37
P_2, P_5	0.29	0.00	0.2675	0.245
P_3, P_6	0.225	0.2675	0.00	0.185
P_4	0.37	0.245	0.185	0.00

Step 4: Update the distance matrix

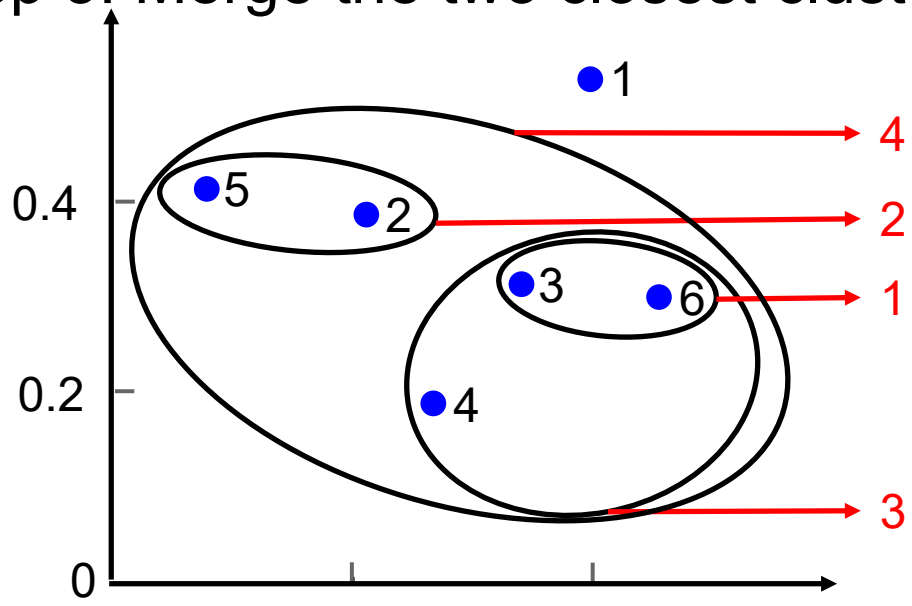
	P_1	P_2, P_5	P_3, P_6, P_4
P_1	0.00	0.29	0.273
P_2, P_5	0.29	0.00	0.26
P_3, P_6, P_4	0.273	0.26	0.00

$$\text{average}\{dist(P_1, P_3), dist(P_1, P_6), dist(P_1, P_4)\}$$
$$= \text{average}\{0.22, 0.23, 0.37\} \approx 0.273$$

$$\text{average}\{dist(P_2, P_3), dist(P_2, P_6), dist(P_2, P_4), dist(P_5, P_3), dist(P_5, P_6), dist(P_5, P_4)\}$$
$$= \text{average}\{0.15, 0.25, 0.20, 0.28, 0.39, 0.29\} = 0.26$$

Result on Average: round 4

Step 3: Merge the two closest clusters



Distance Matrix:

	P_1	P_2, P_5	P_3, P_6, P_4
P_1	0.00	0.29	0.273
P_2, P_5	0.29	0.00	0.26
P_3, P_6, P_4	0.273	0.26	0.00

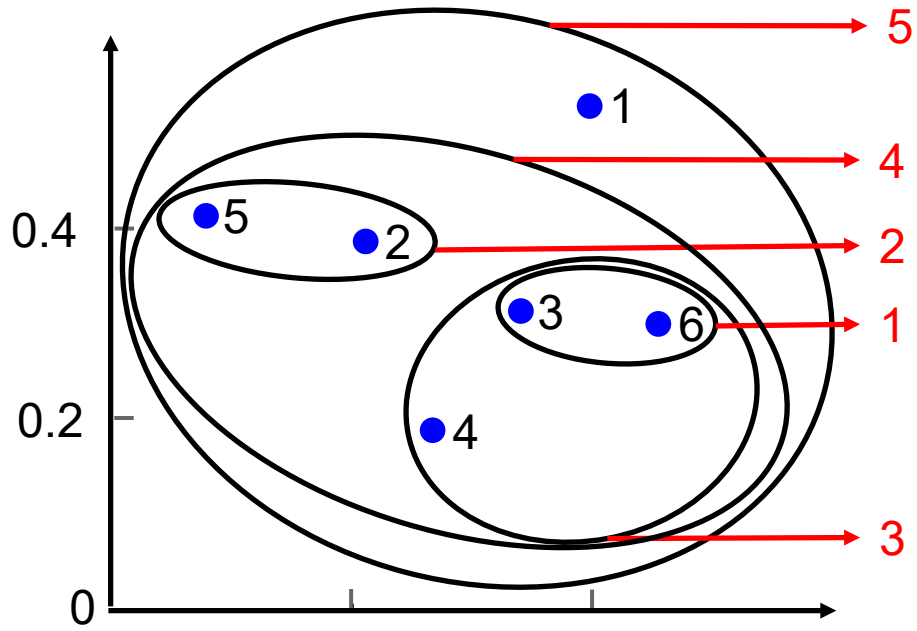
Step 4: Update the distance matrix

	P_1	P_2, P_5, P_3, P_6, P_4
P_1	0.00	0.28
P_2, P_5, P_3, P_6, P_4	0.28	0.00

$$\text{average}\{dist(P_1, P_3), dist(P_1, P_6)\}, dist(P_1, P_2), dist(P_1, P_5), dist(P_1, P_4)\}$$
$$= \text{average}\{0.22, 0.23, 0.24, 0.34, 0.37\} = 0.225$$

Result on Average: round 5

Step 3: Merge the two closest clusters



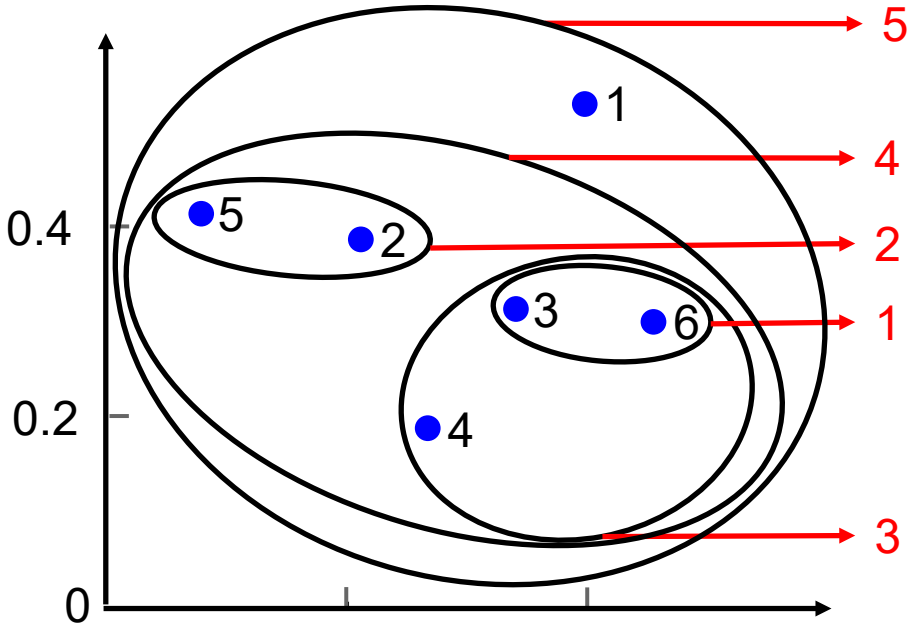
Distance Matrix:

	P_1	P_2, P_5, P_3, P_6, P_4
P_1	0.00	0.28
P_2, P_5, P_3, P_6, P_4	0.28	0.00

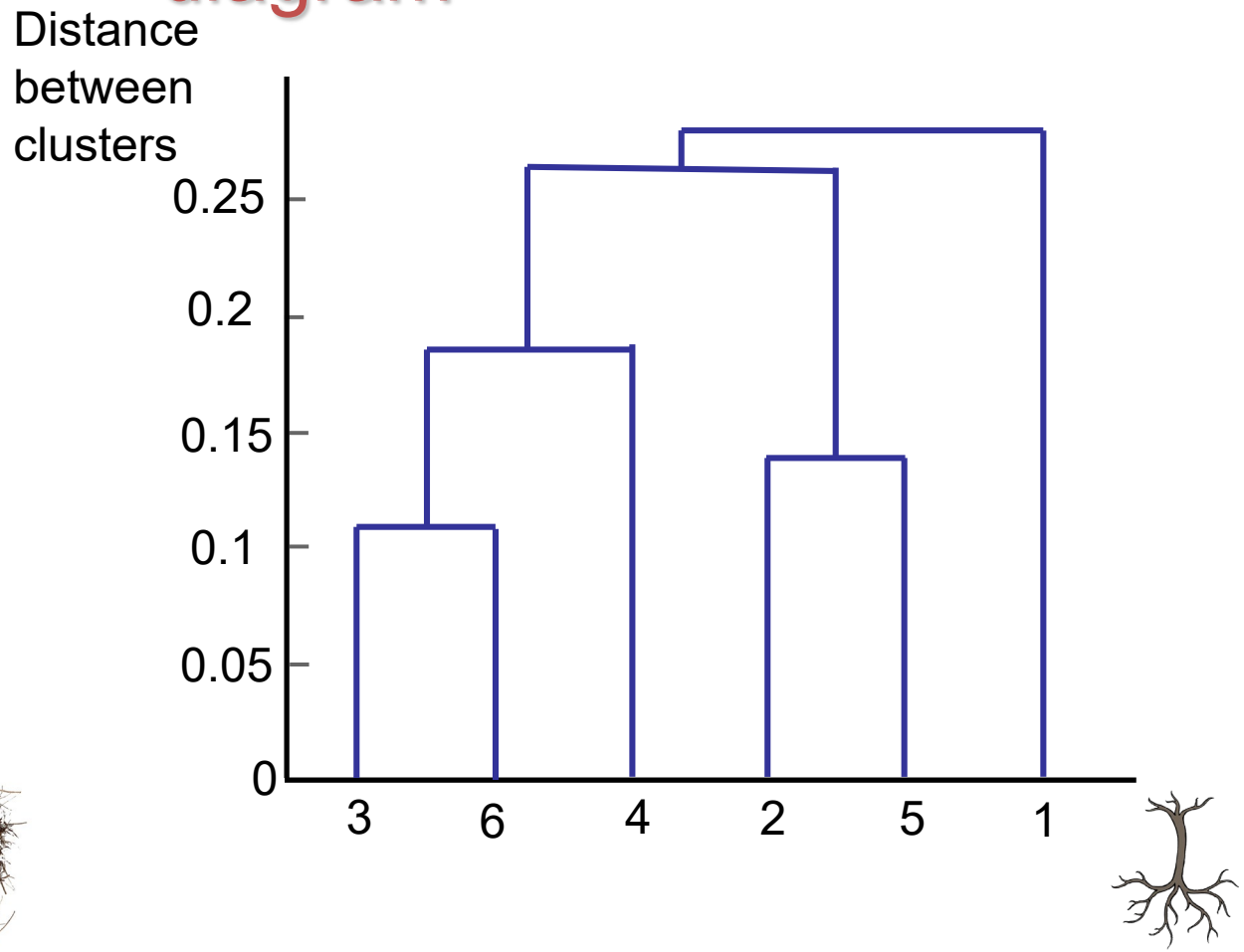
How to represent the hierarchical clustering result?

Representation: dendrogram

Nested cluster diagram

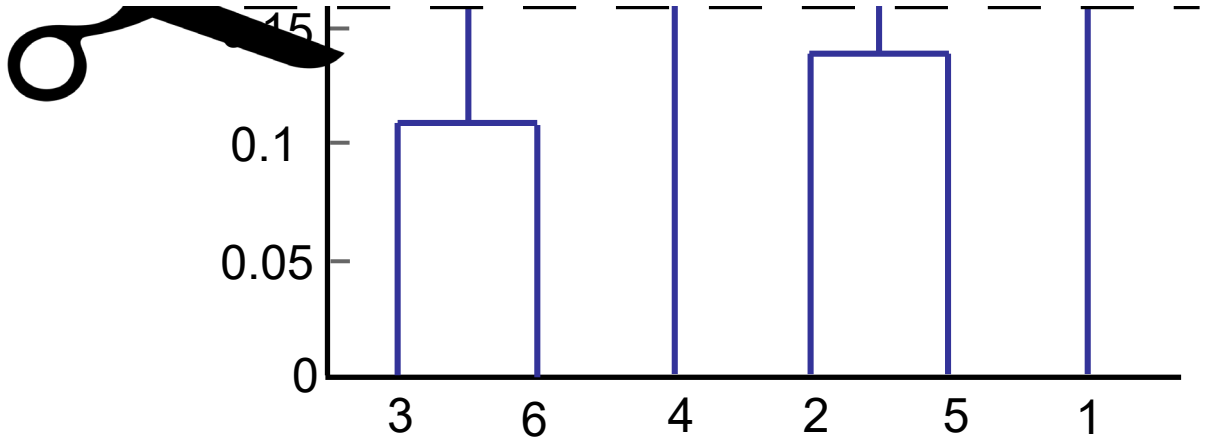


Dendrogram: a tree-like diagram



Clustering result by dendrogram (1)

- Cutting at a particular level produces corresponding clustering result

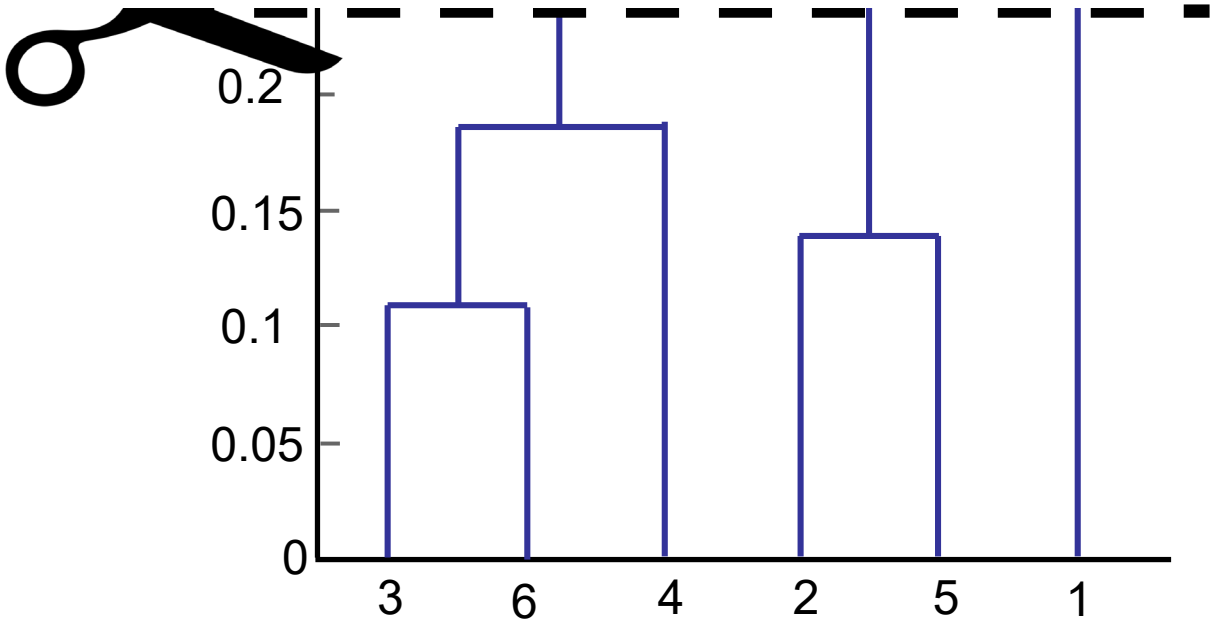


Clustering result:

{3,6}, {4}, {2,5}, {1}

Clustering result by dendrogram (2)

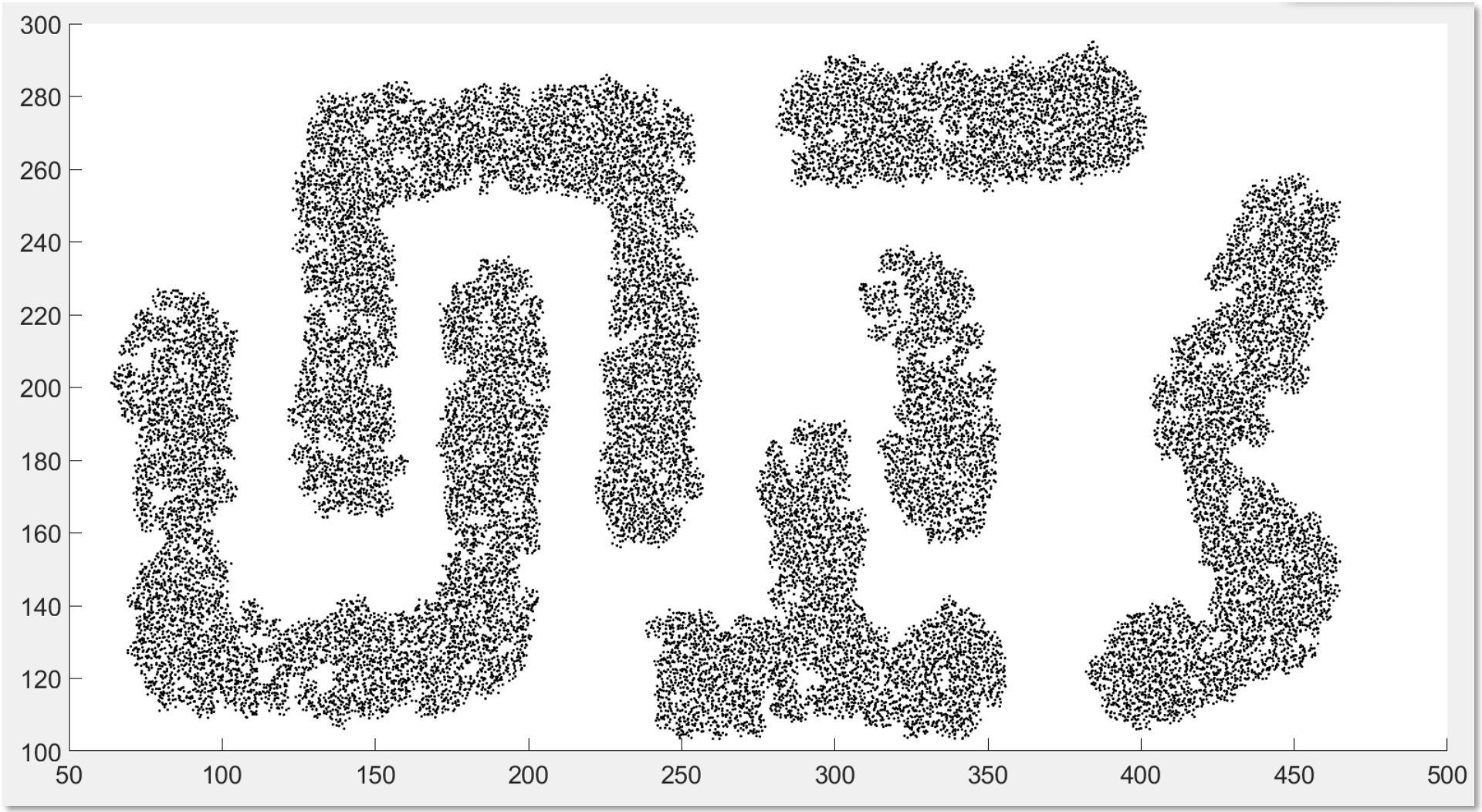
- Cutting at a particular level produces corresponding clustering result



Clustering result:

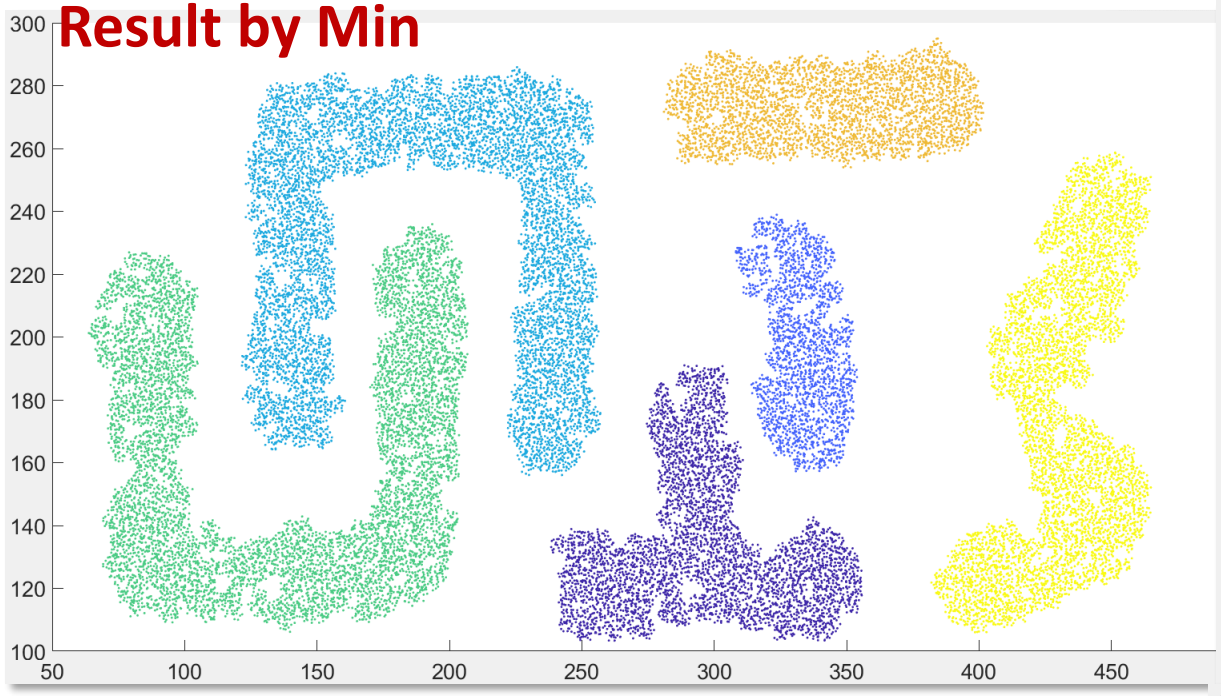
{3,6,4}, {2,5}, {1}

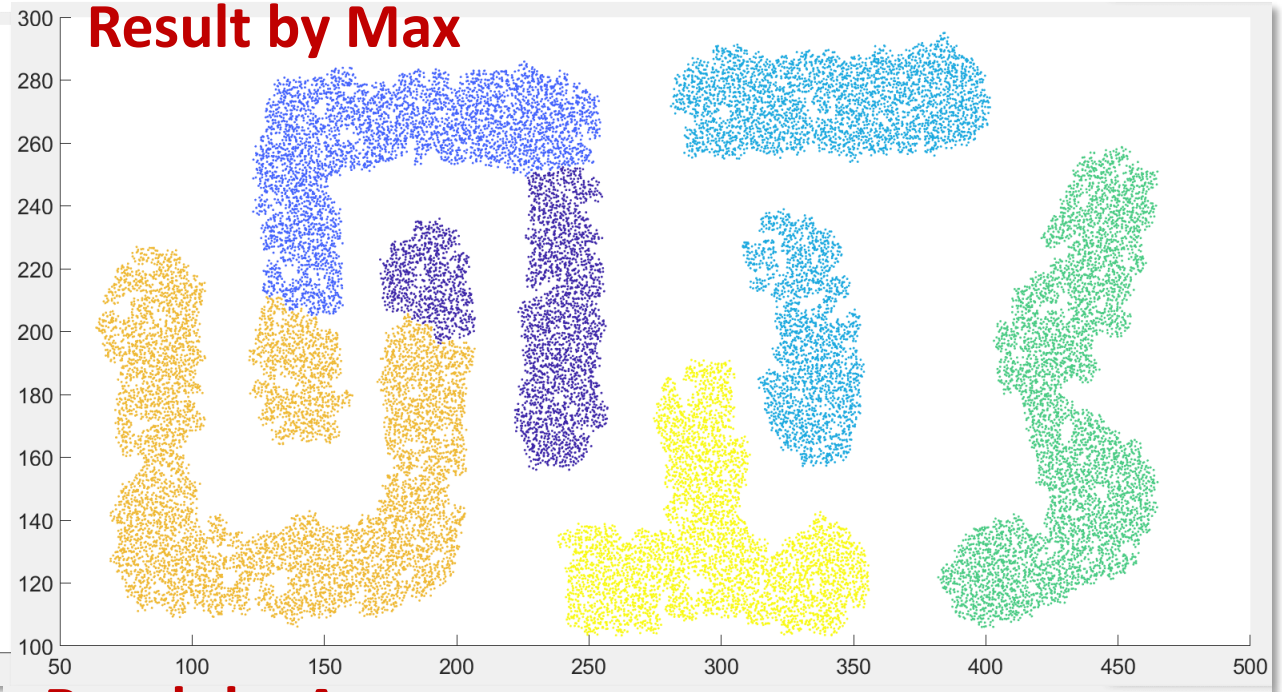
What are the clustering results (1)

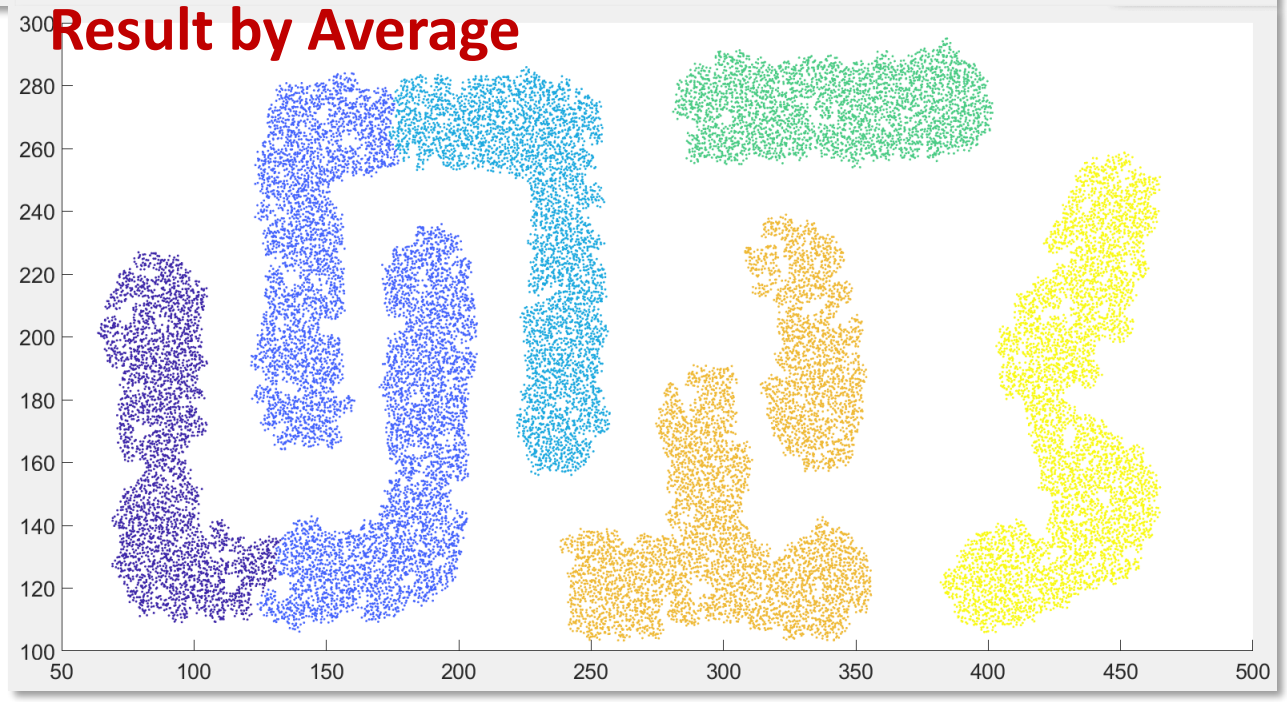


382.8340	294.0955
384.4406	294.9963
384.0002	294.9348
381.6047	292.3745
382.2935	292.2753
383.1847	293.2622
384.3725	293.3131
385.6911	293.5673
381.7935	291.3870
383.0776	292.9254
383.8384	292.9536
385.3704	292.3577
385.4089	291.6196
298.7562	290.8543
298.0548	290.7459
300.3409	290.0543
304.8346	290.8732
306.1174	291.2395
306.2808	290.4160
307.3962	290.1910

31275 2D points

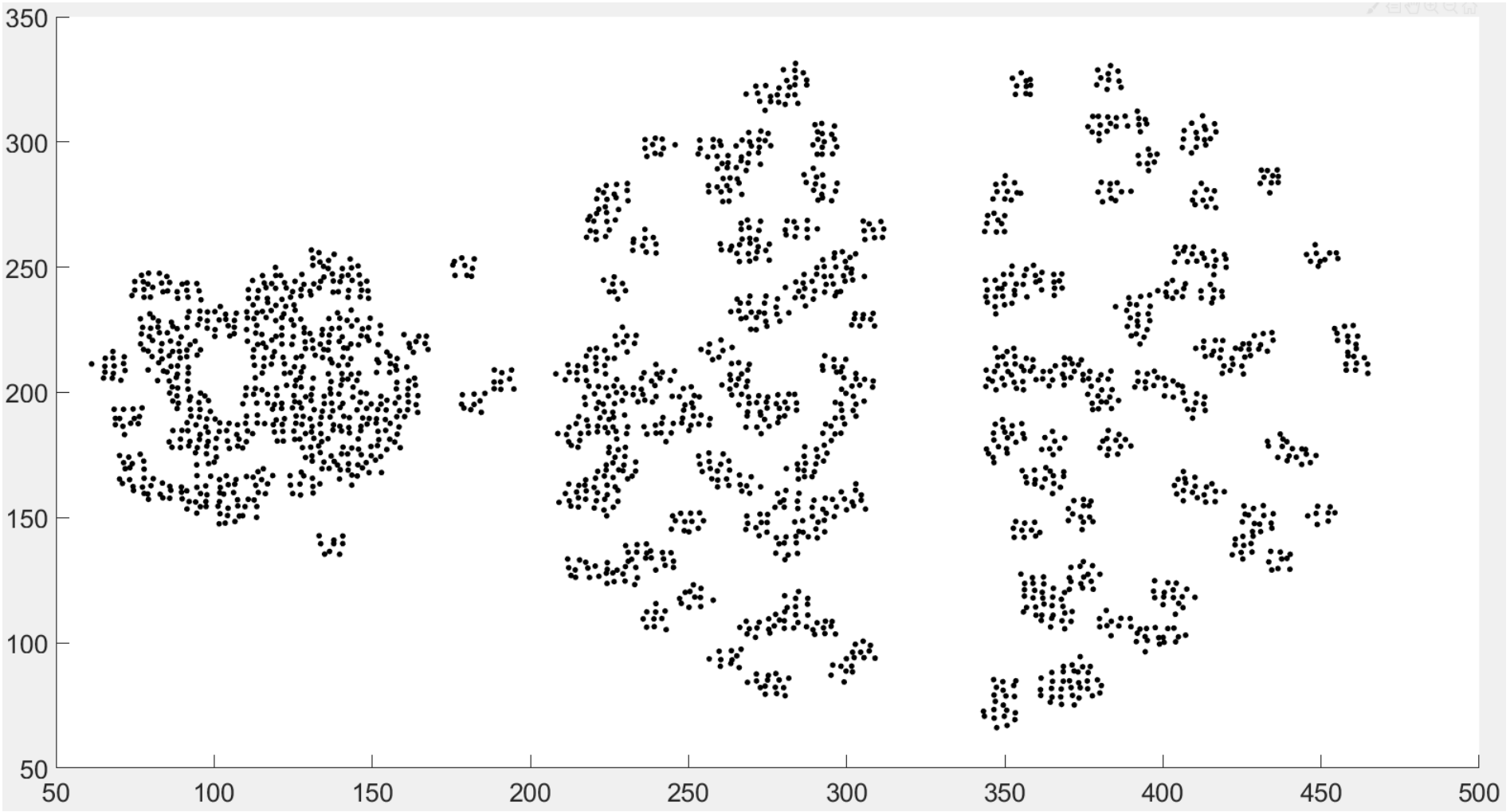






- **Min** is best at handling *non-elliptical shapes*

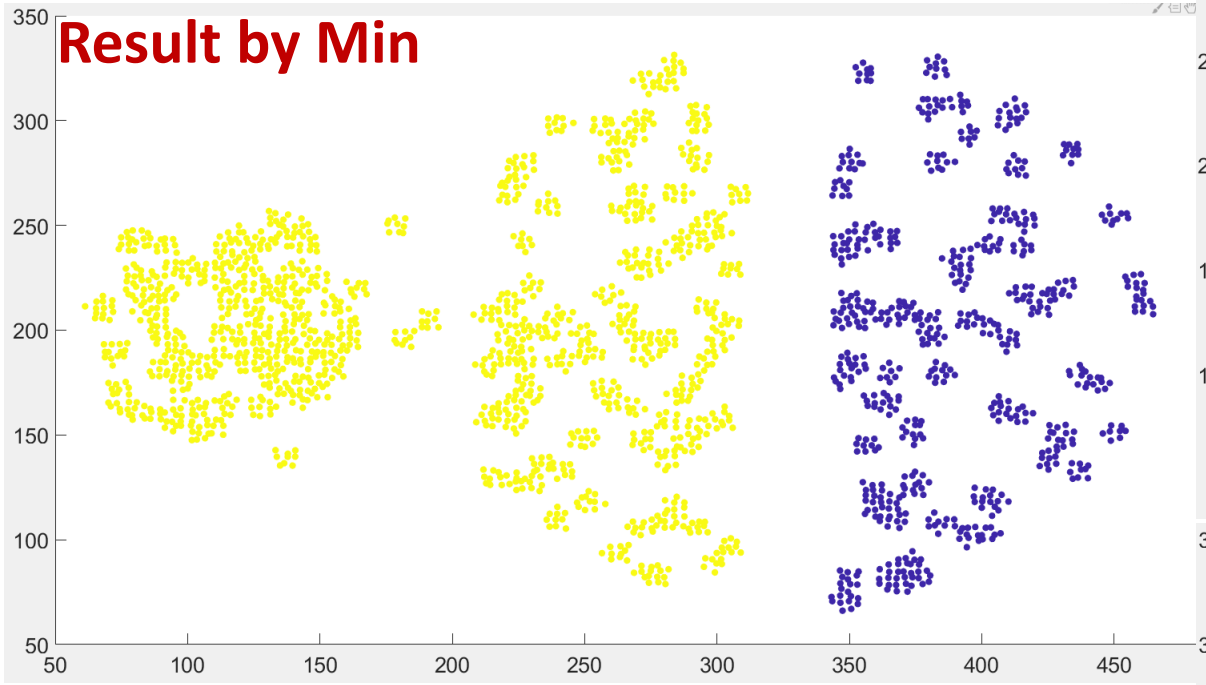
What are the clustering results (2)

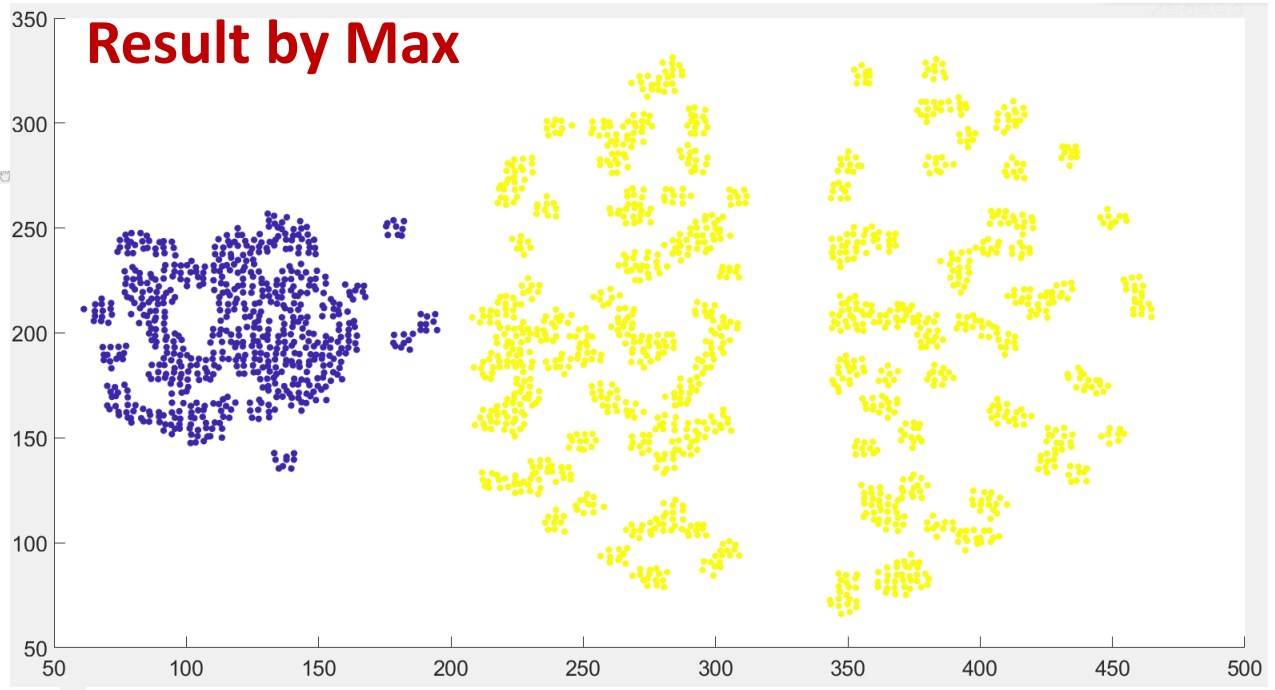


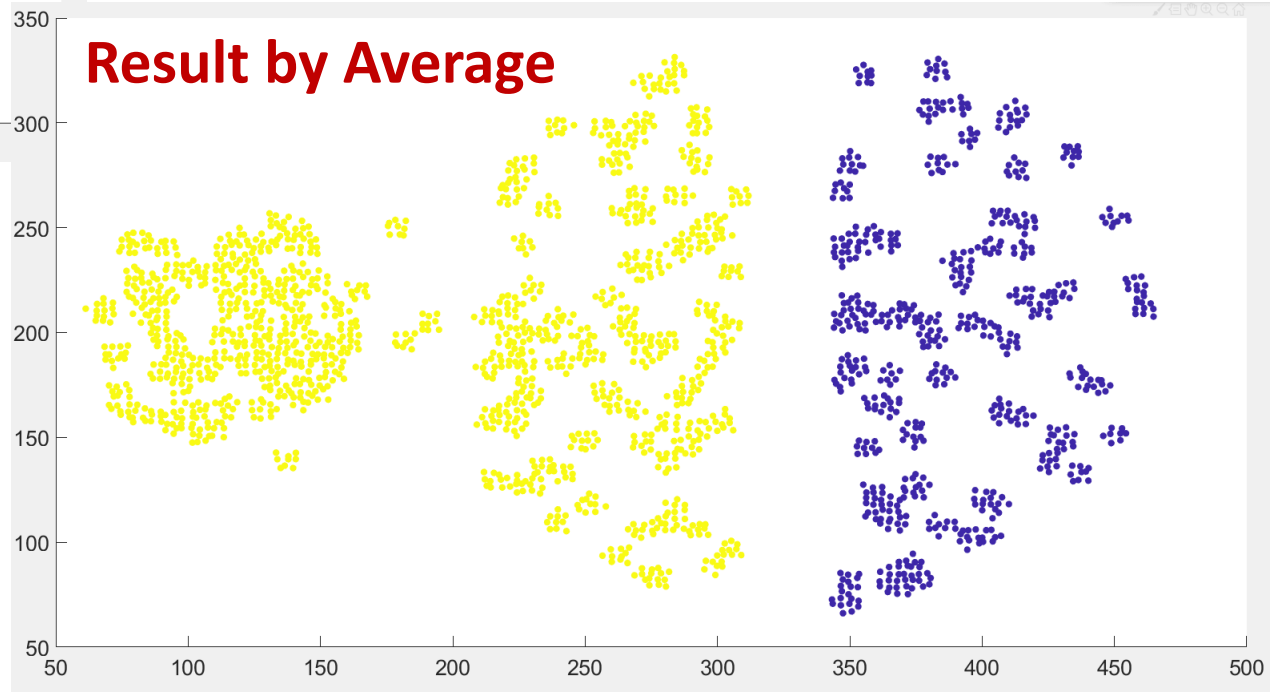
283.8340	331.3622
383.4406	330.4740
280.0002	328.8514
283.6047	328.5624
286.2935	327.6148
355.1847	327.6099
379.3725	328.7623
382.6911	327.2522
385.7935	328.2669
281.0776	324.5554
283.8384	325.6544
287.3704	324.7290
352.4089	325.4667
356.7562	324.3848
358.0548	324.9629
380.3409	325.6072
382.8346	324.7940
386.1174	324.3231
271.2808	322.2980
274.3962	322.5067
278.6015	321.5604

2104 2D points

2 Cluster:

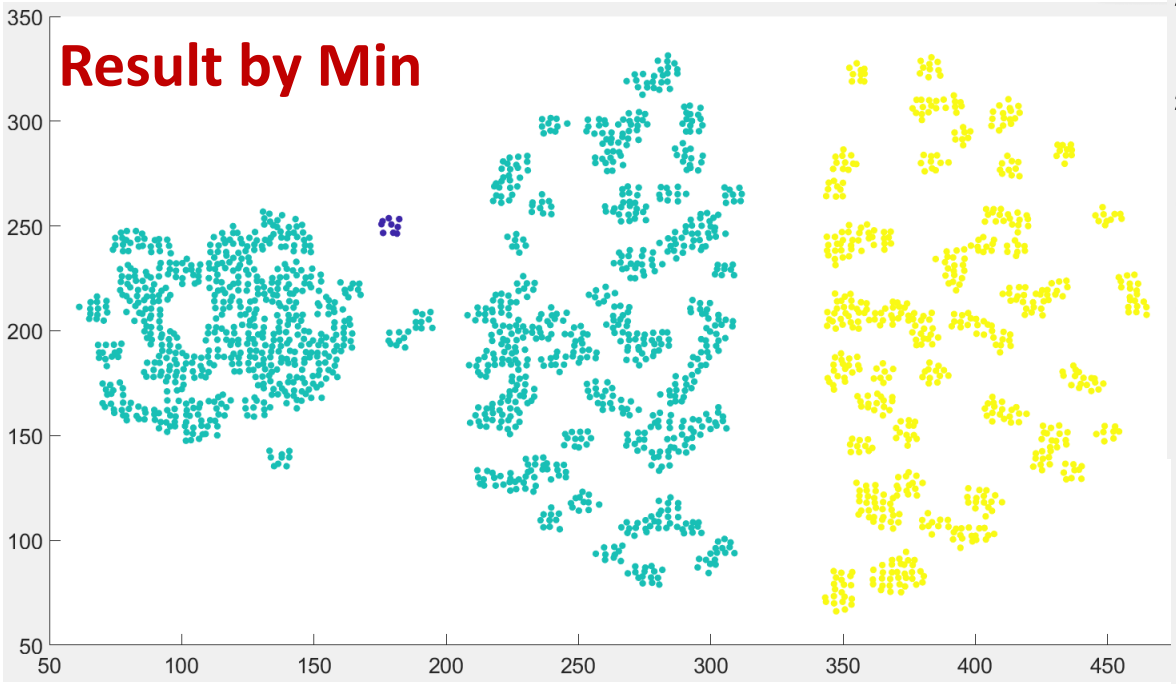


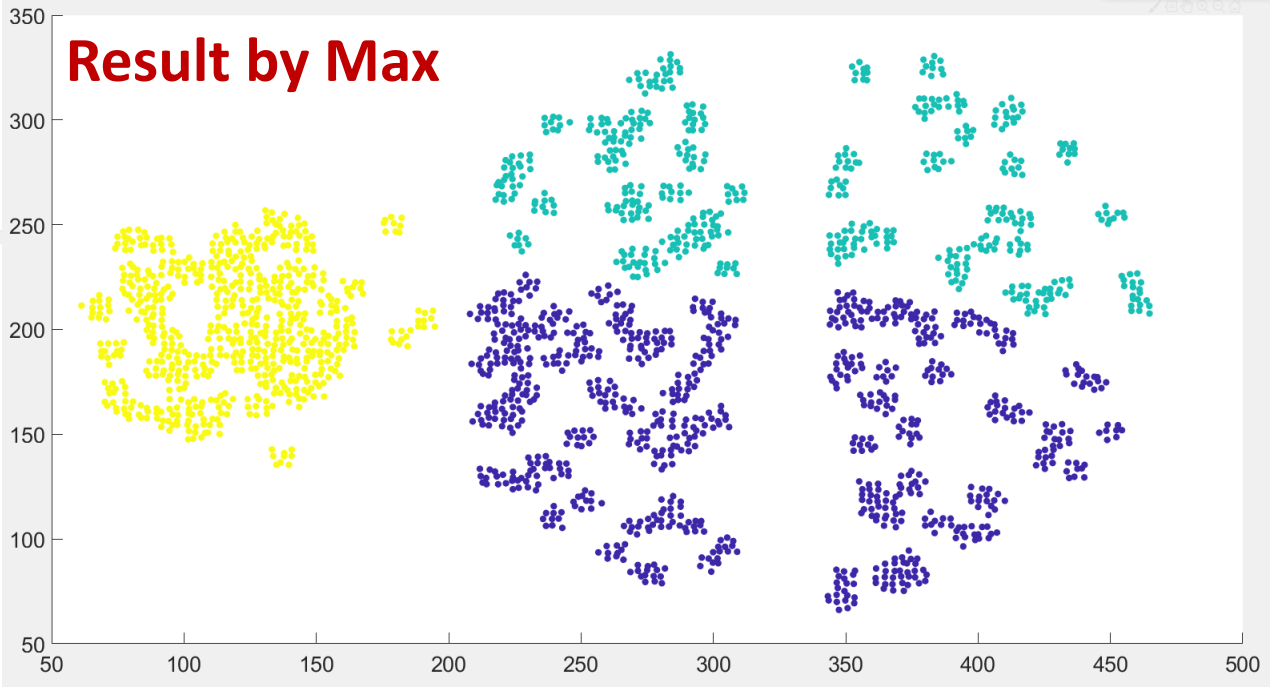


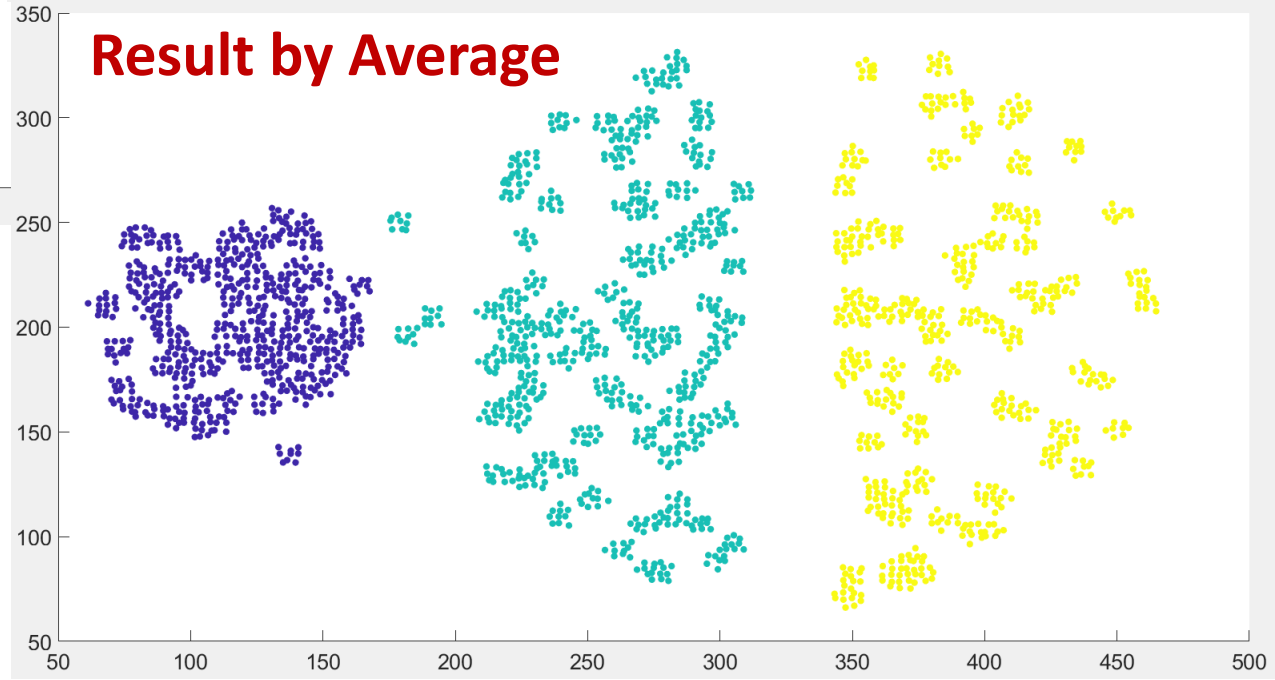


- **Max** tends to form globular shape

3 Cluster:

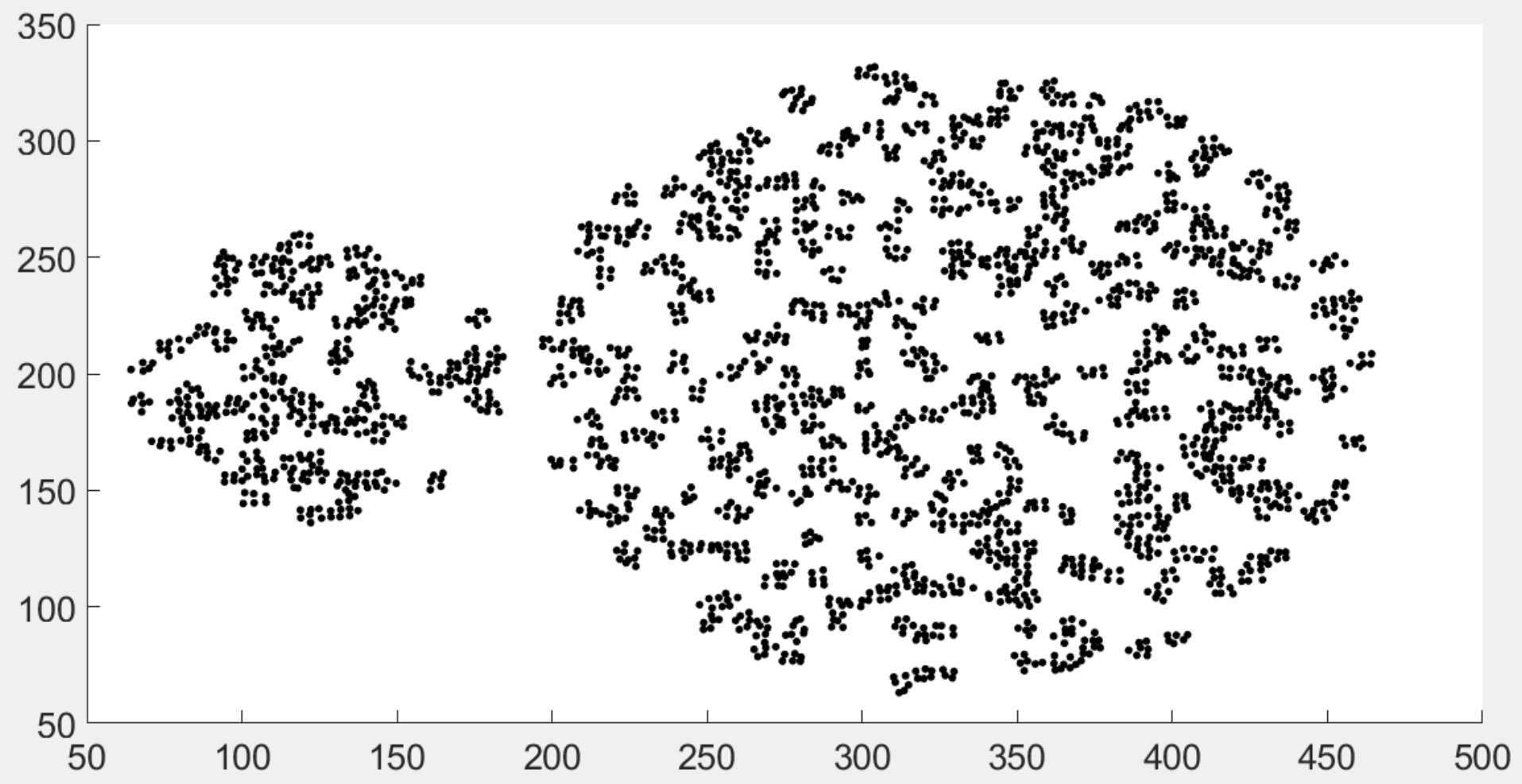






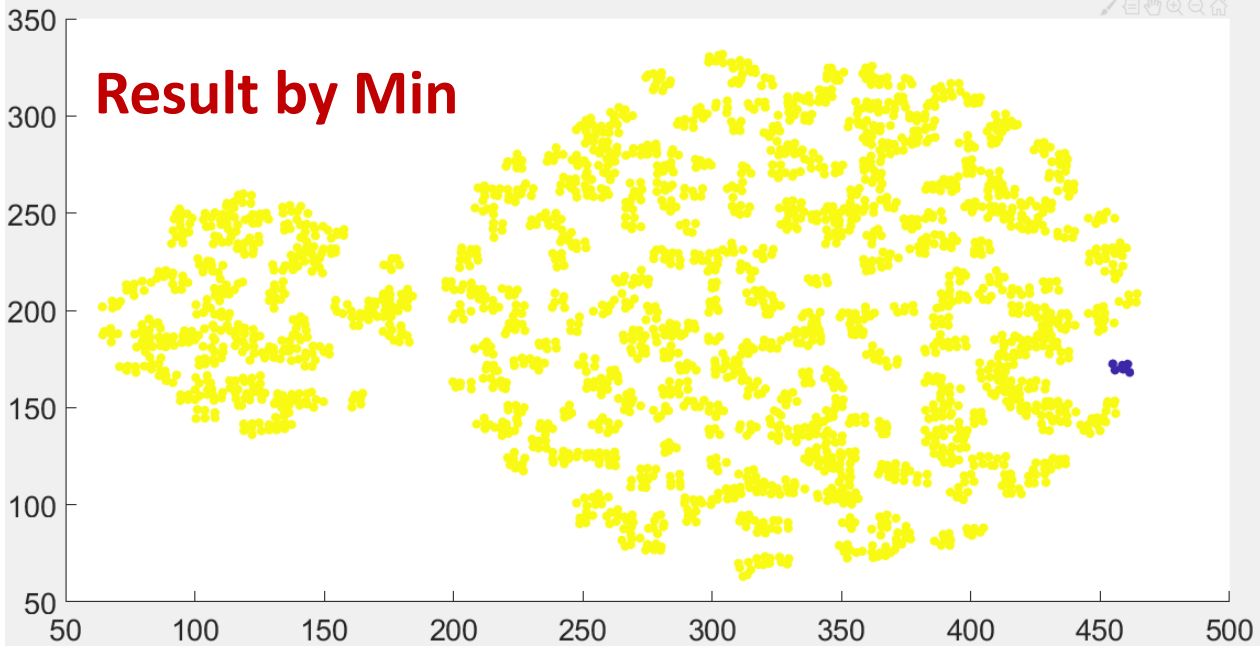
- **Min** is sensitive to noise
- **Max** tends to break large clusters

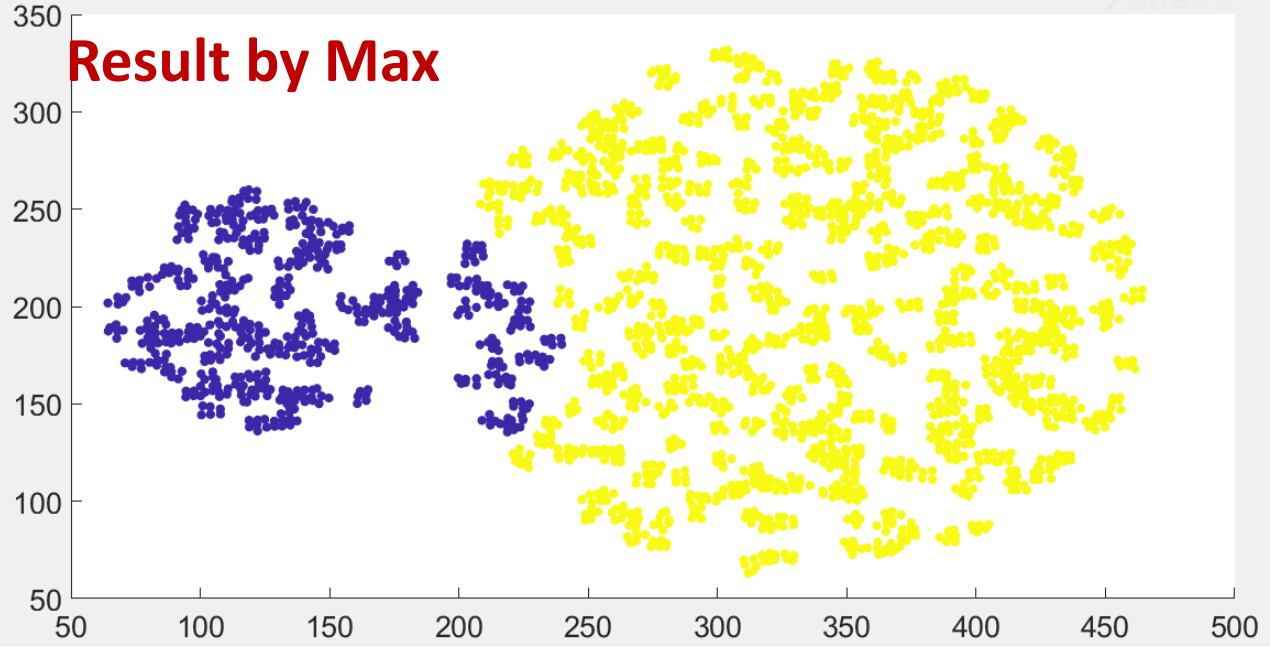
What are the clustering results (3)

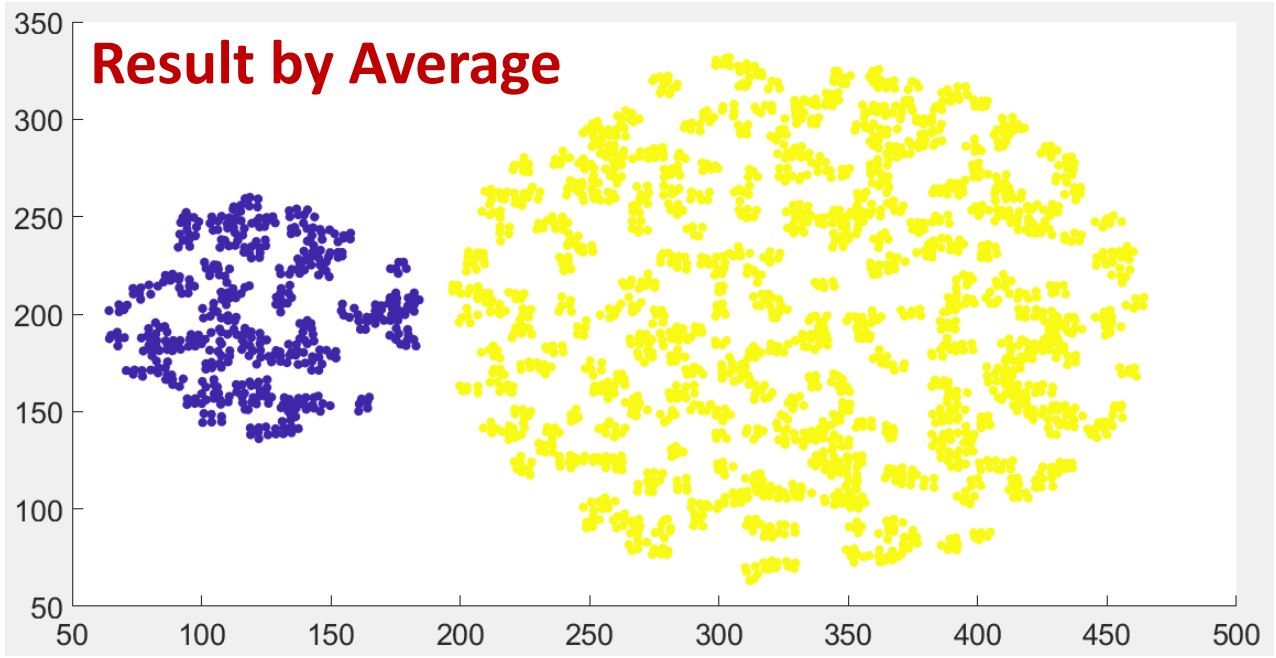


298.8340	330.4028
302.4406	331.1778
304.0002	331.6850
298.6047	327.6734
301.2935	328.1729
304.1847	327.3245
307.3725	327.3892
310.6911	328.5227
313.7935	327.3002
308.0776	324.5318
310.8384	325.3790
314.3704	324.0302
316.4089	324.0644
344.7562	324.5751
346.0548	324.6732
359.3409	324.7244
361.8346	325.6144
275.1174	321.0720

2409 2D points



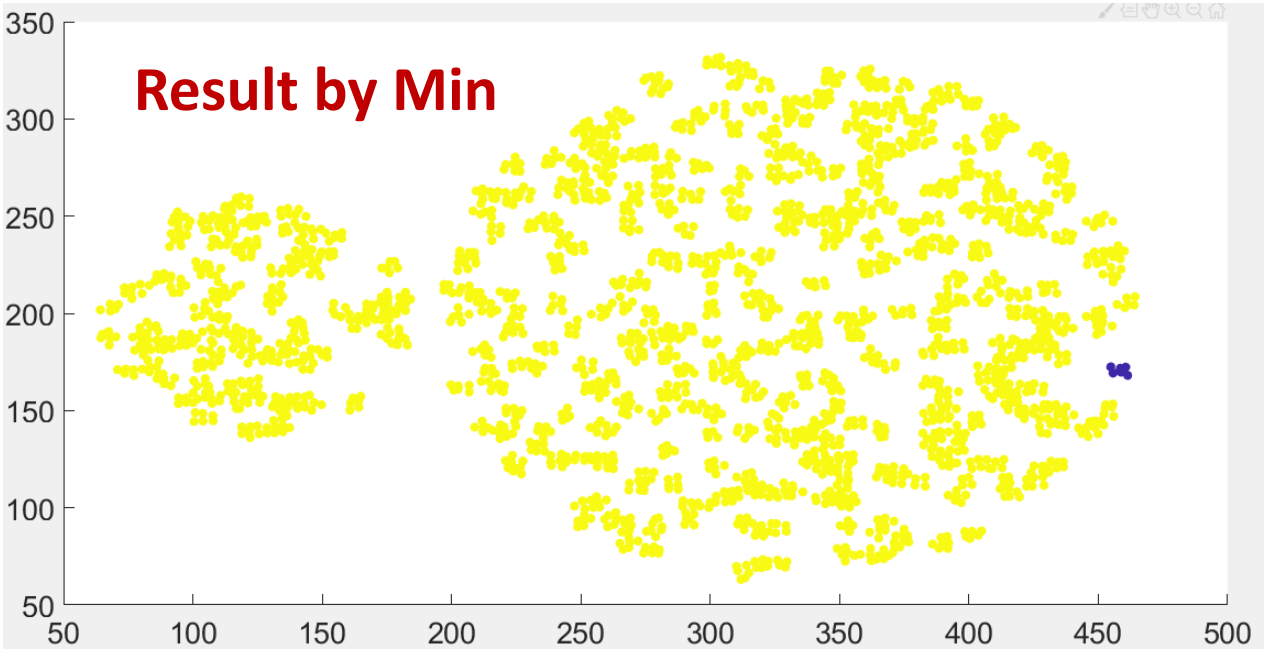


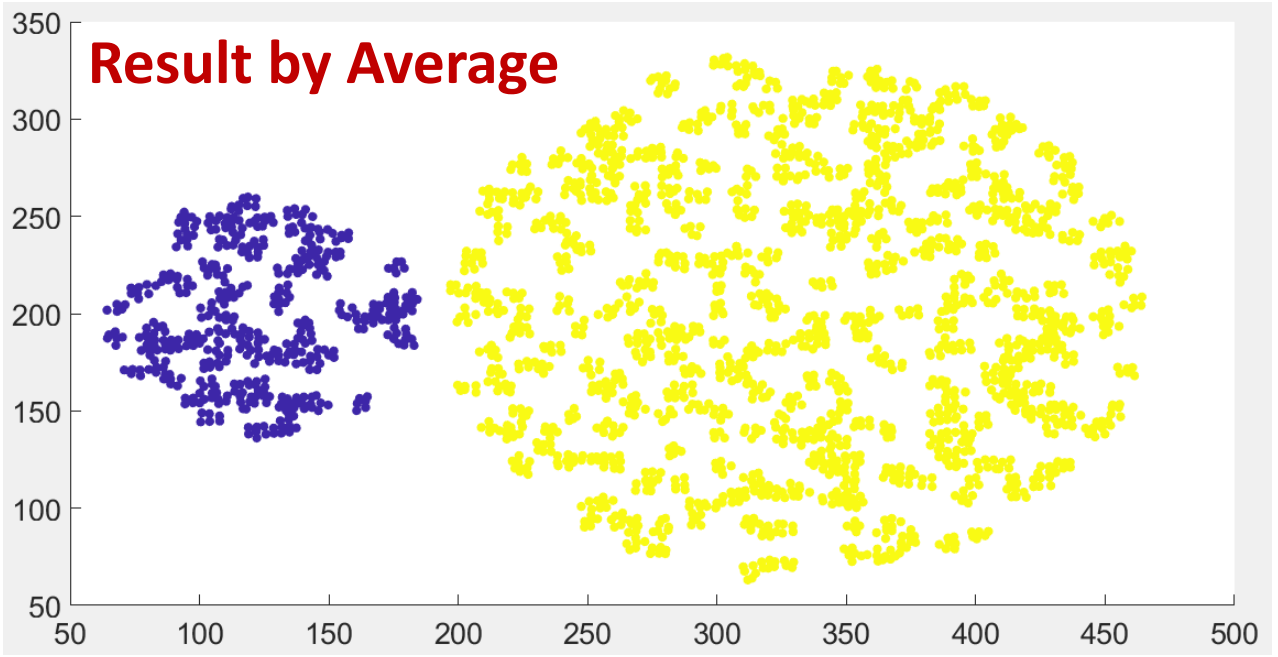


- **Min** is sensitive to noise
- **Max** tends to break large clusters
- **Max** tends to form globular shape

How to determine which to use?

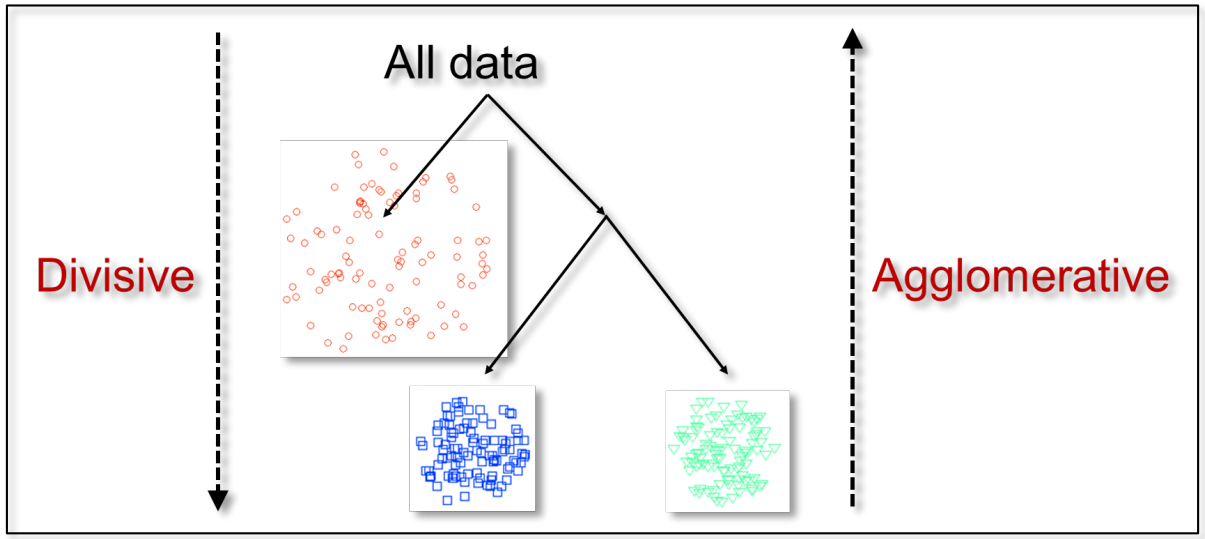
- By the clustering validity index
  - SSE
- By "explanation"





Please always remember ***the NFL theorem****!*

Summarization of hierarchical-based clustering



AGNES

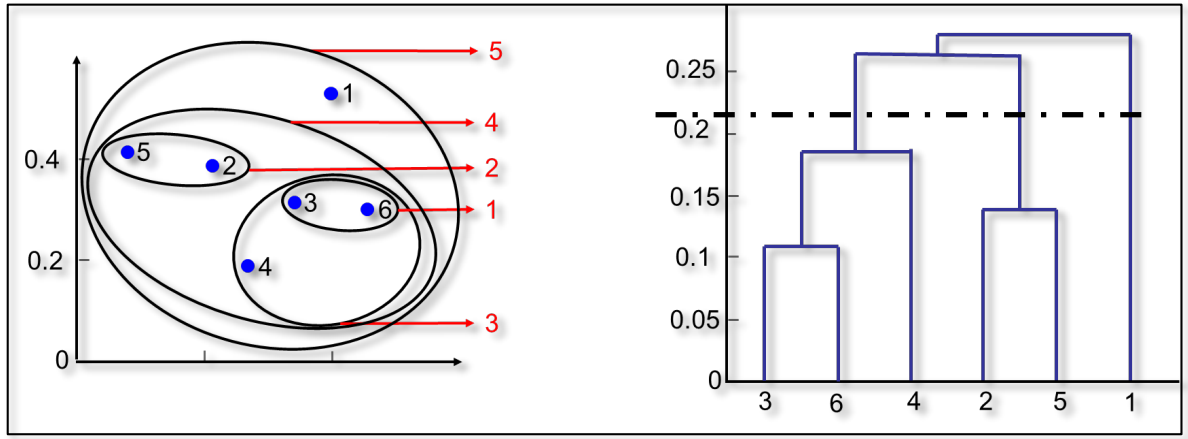
AGglomerative NESting

1. Compute the *distance matrix*
2. Let *each* data point be a cluster

Repeat

3. *merge* the two closest clusters
4. *update* the distance matrix

Until only a *single* cluster remains



the NFL theorem

Ways to update the distance matrix

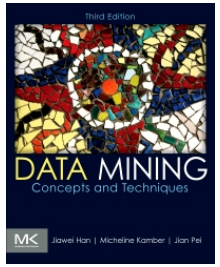
- Min – Single linkage
- Max – Complete linkage
- Average – Average linkage

Summary

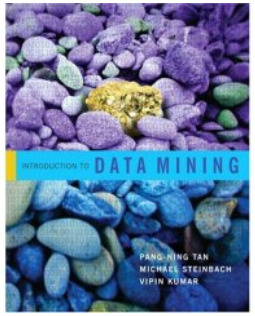
- **Prototype-based clustering** (*k*-means)
  - Easy and simple
  - Need to specify *k*, cannot deal with non-globular shape
- **Density-based clustering** (DBSCAN)
  - Can deal with non-globular shape
  - Need to specify MinPts and Eps, density-sensitive
- **Hierarchical-based clustering** (AGNES)
  - Hierarchical structure, deal with non-globular shape
  - Select within Min, Max, Average, each has its own disadvantage

Recommended reading (not required)

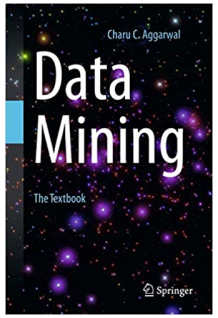
- [Han et al., 2012]
  - Sec. 10.3, 10.4



- [Tan et al., 2005]
  - Sec. 8.3, 8.4



- [Aggarwal, 2015]
  - Sec. 6.4, 6.6.2



Next week: anomaly detection