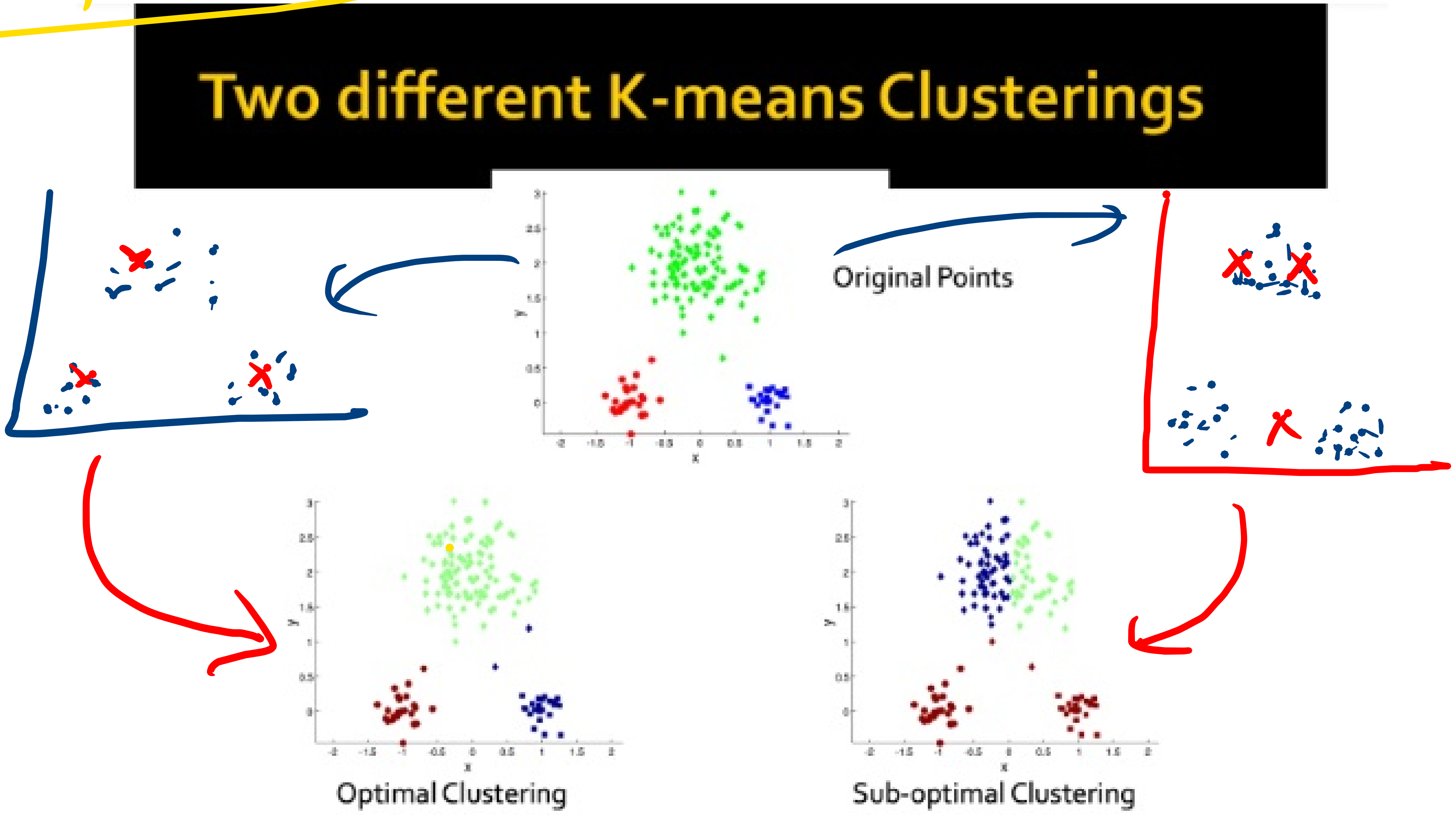
Disadvantage of K-means



→ Initialization of cluster

(1) Run Kmeans multiple times with random initialization of centroids

(2) Kmeans++

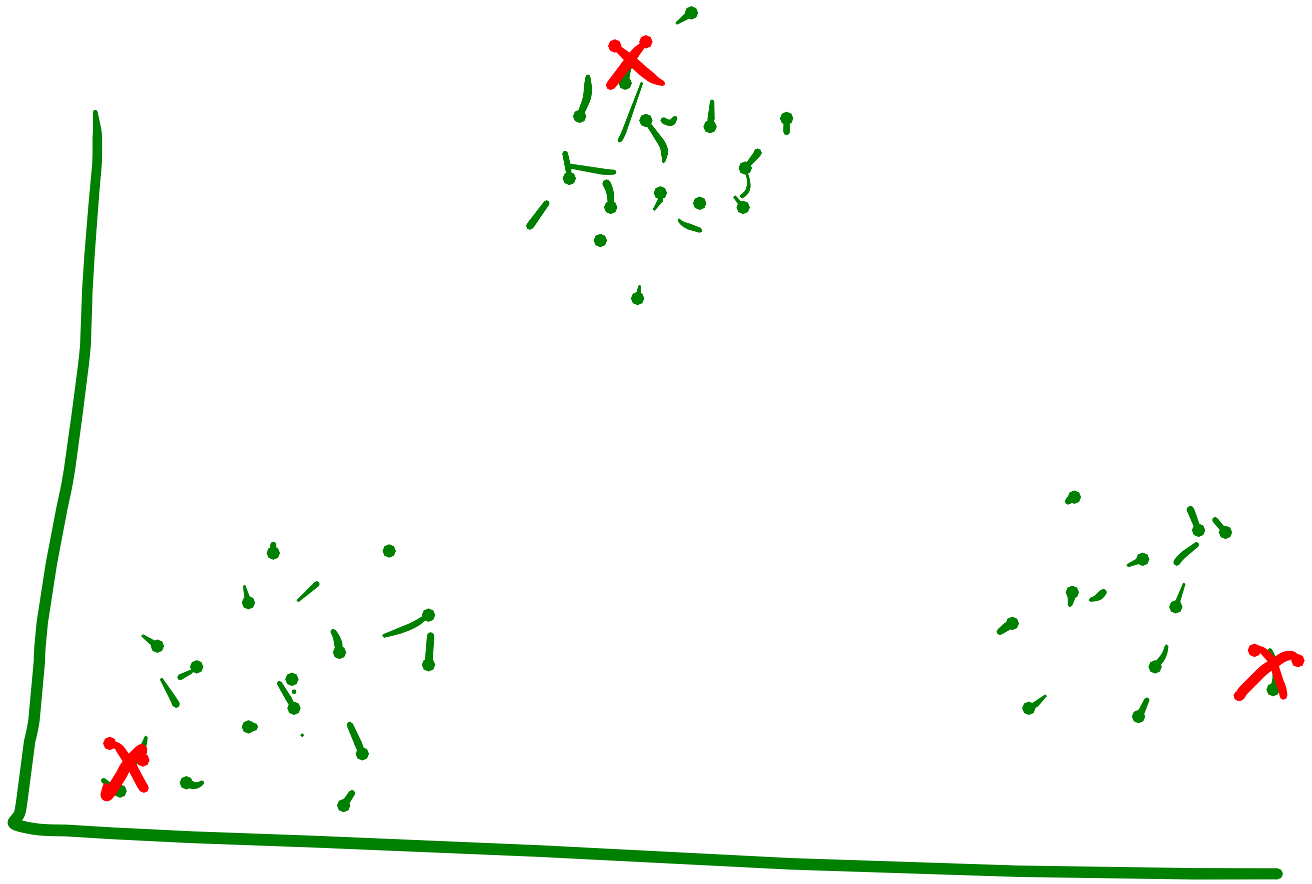
Initialization

↳ Select 1st centroid from data point (randomly)

↳ Select 2nd centroid which is farthest from 1st one

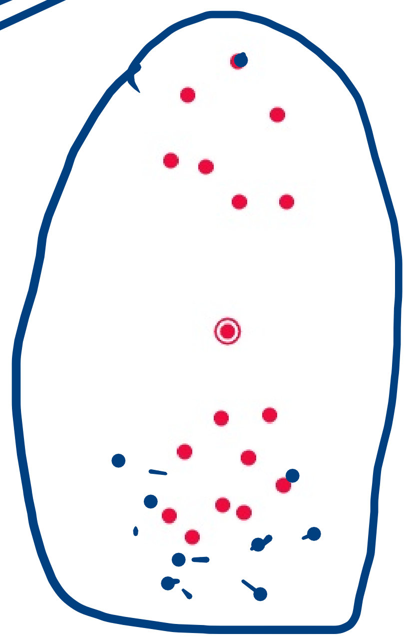
↳ Select 3rd centroid which is farthest from both 1st & 2nd centroid

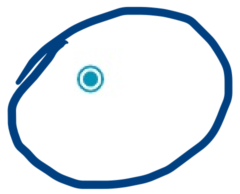
→ Run Kmeans



→ Outlier

Case 1



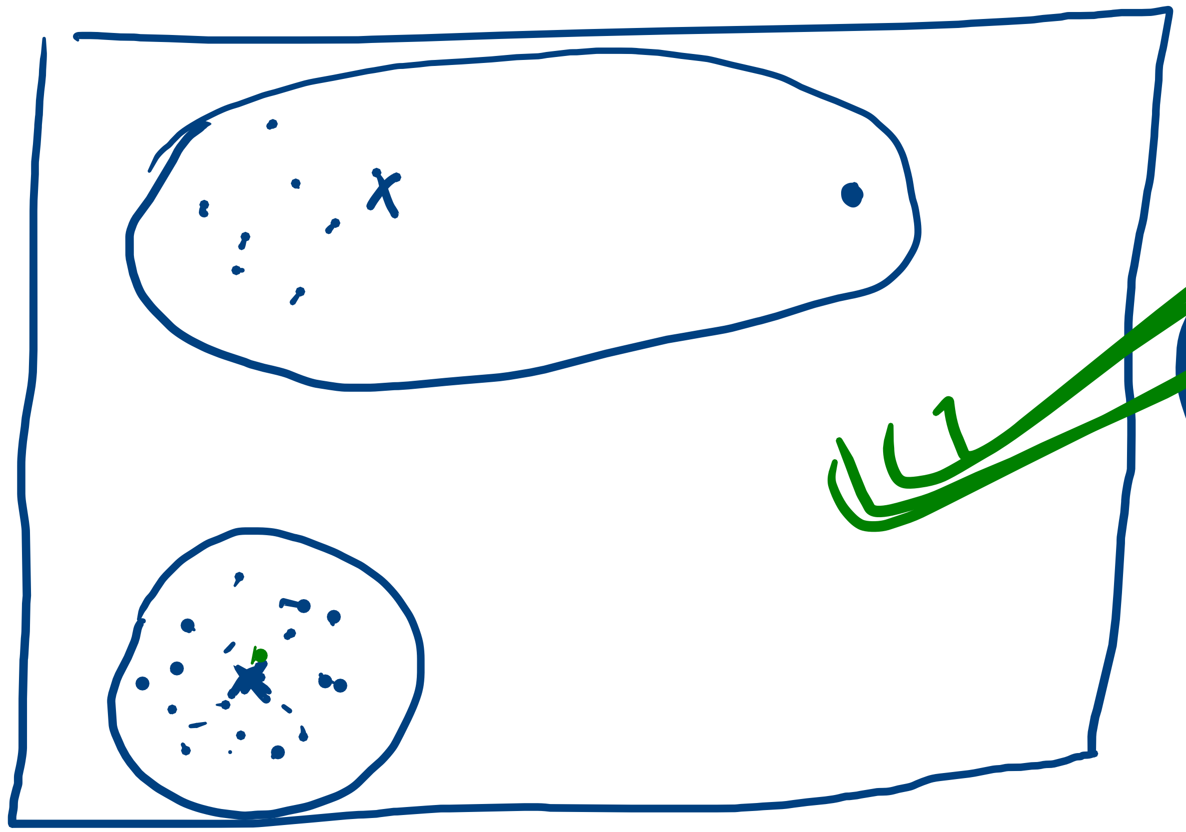


WCSS ↑

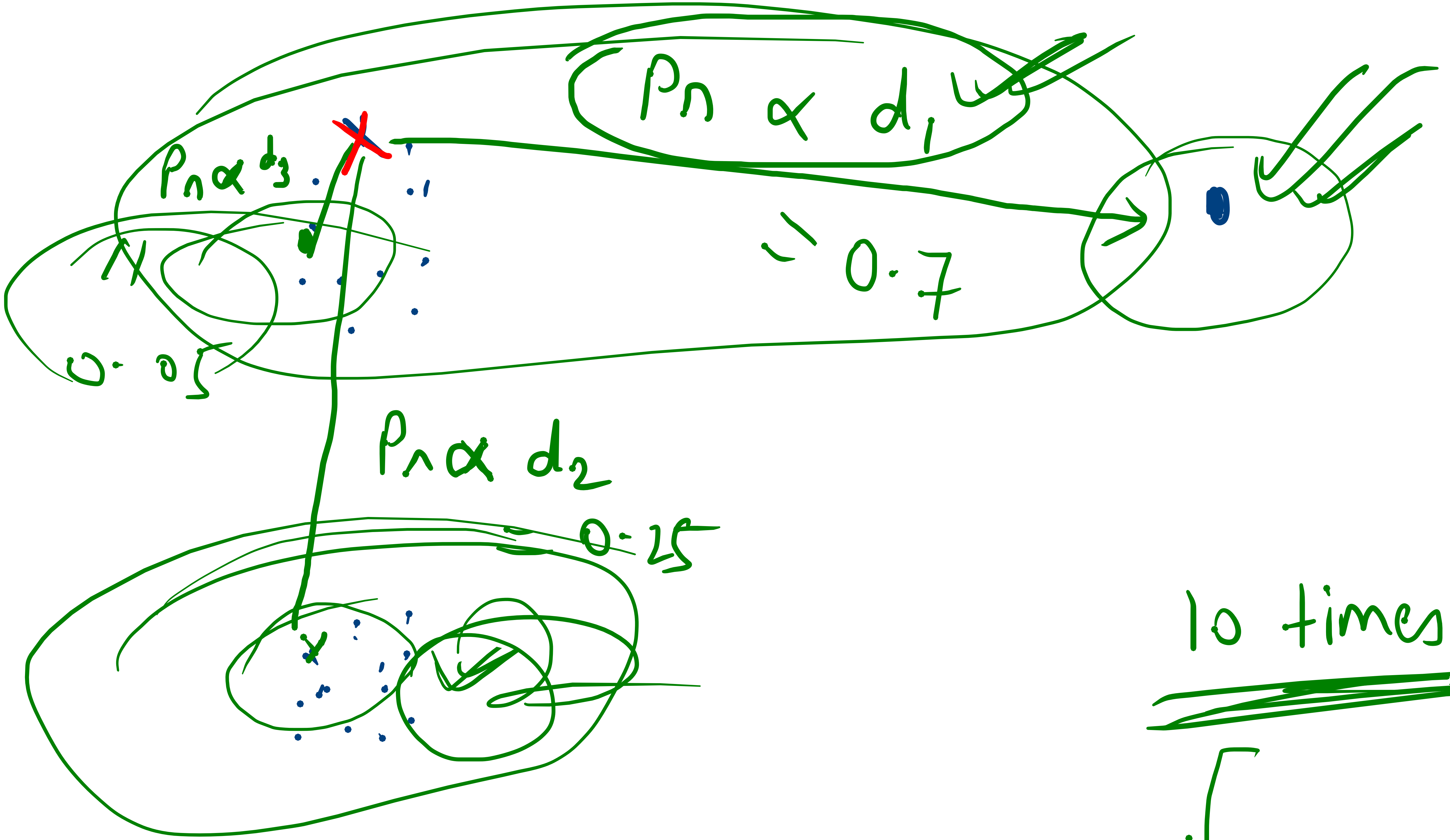
→ lowest WCSS

↳ Best cluster

Case 2

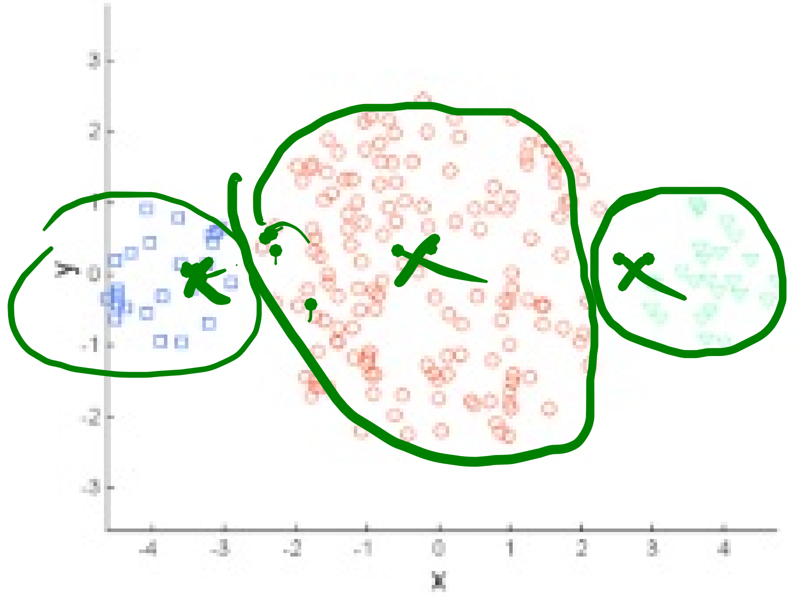


WCSS ↓

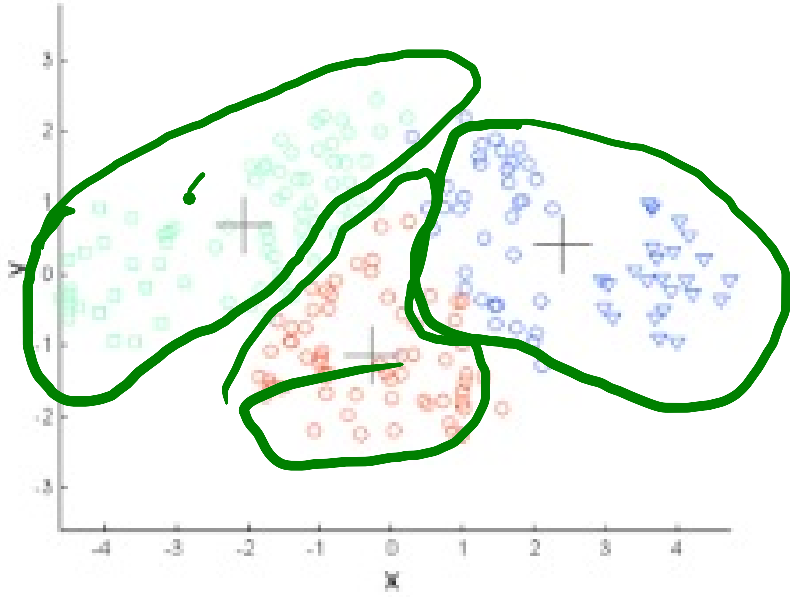

$P_n \propto d_1$
≈ 0.7
$P_n \propto d_3$
0.05
$P_n \propto d_2$
≈ 0.25

10 times

Limitations of K-means: Differing Sizes

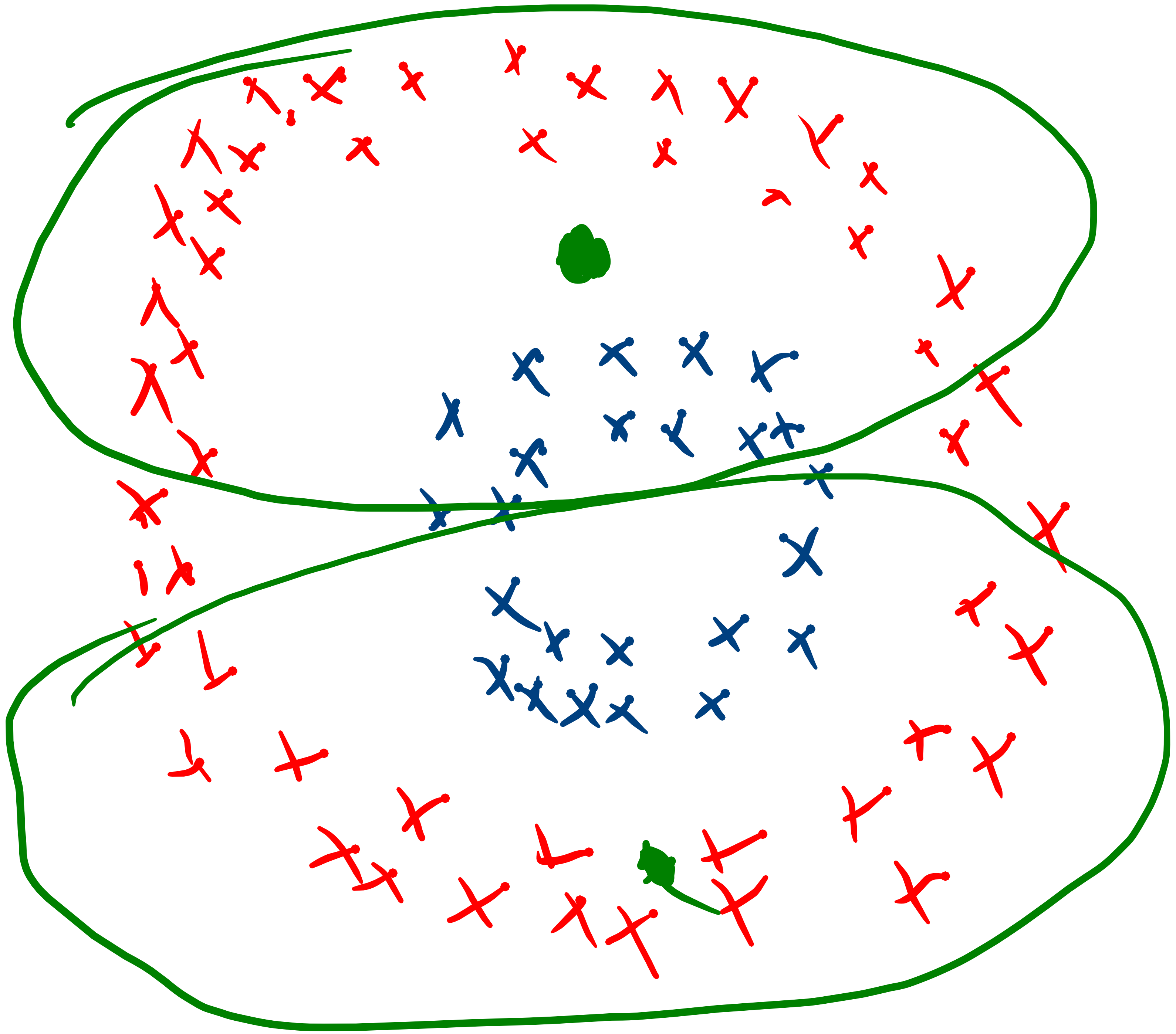


Original Points

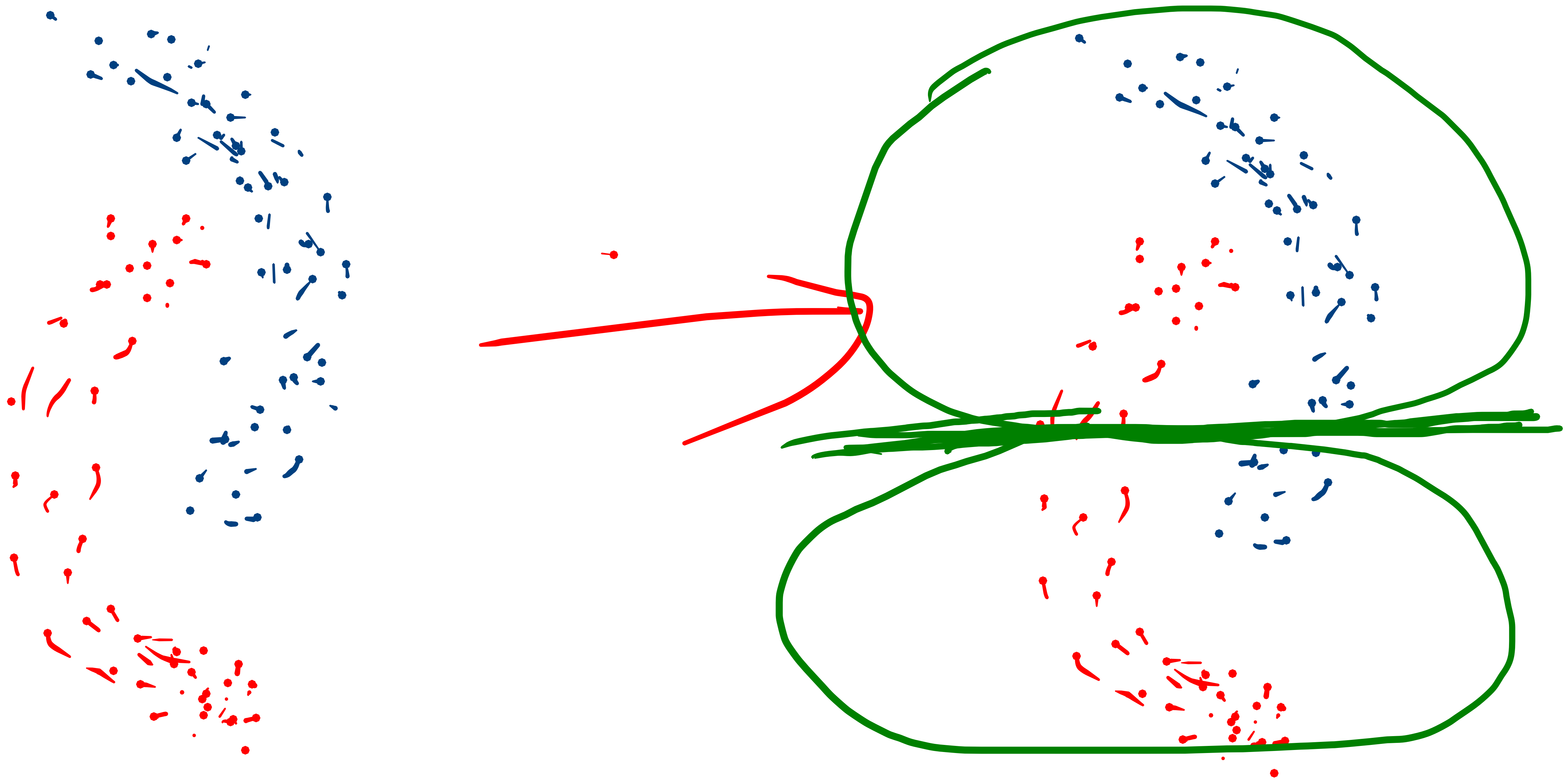


K-means (3 Clusters)

Works Clusters are similar size



K-means always form spherical cluster



linear decision boundary

Kmedian → less sensitive to outliers

→ Intialize k cluster centre

→ Repeat

→ Assignment

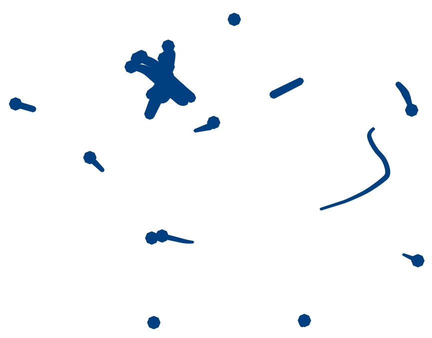
→ Update → Medians of points

$$\frac{1}{n_k} \sum_{i \in C_i} x_i$$

K-means

→ Number of clusters need to be decided initially

Does the centroid in K-means / K-median need to be a data point??



Always guaranteed that mediod is a datapoint

K-mediods (less sensitive to outliers)

- Select K points as mediods

Repeat

- Assign each point to the cluster which is closest and calculate total sum of distance (dissimilarity)

dold

- Swap a mediod with non-mediod and calculate total sum of distance (dissimilarity)

dnew

- S = dnew − dold

S<0 → Select this point as better mediod

Eg Datapoints → Manhattan

(2,6), (3,4), (3,8), (4,7) (6,2) (6,4)

(7,3) (7,4) (8,5) (7,6)

	Dist from (3,4)	Dist from (7,4)
(2,6)	$\|3-2\|+\|4-6\| = 3$	$\|7-2\|+\|4-6\| = 7$
(3,8)	4	8
(4,7)	4	6
(6,2)	5	3
(6,4)	3	1
(7,3)	5	1
(8,5)	6	2
(7,6)	6	2

old dist

~~Dissimilarity~~ $= 3+4+4+3+1+1+2+2$

$= 20$

	Distance from (3,4)	Distance from (7,3)
(2,6)	$\lvert 3-2\rvert + \lvert 4-6\rvert = 3$	$\lvert 2-7\rvert + \lvert 6-3\rvert = 8$
(3,8)	4	9
(4,7)	4	7
(6,2)	5	2
(6,4)	3	2
(7,4)	4	1
(8,5)	6	3
(7,6)	6	3

New dist = Dissimilarity (total dist) $= 3+4+4+2+2+1+3+3$

$= 22$

$S = \text{New} - \text{Old} = 22 - 20 = 2$

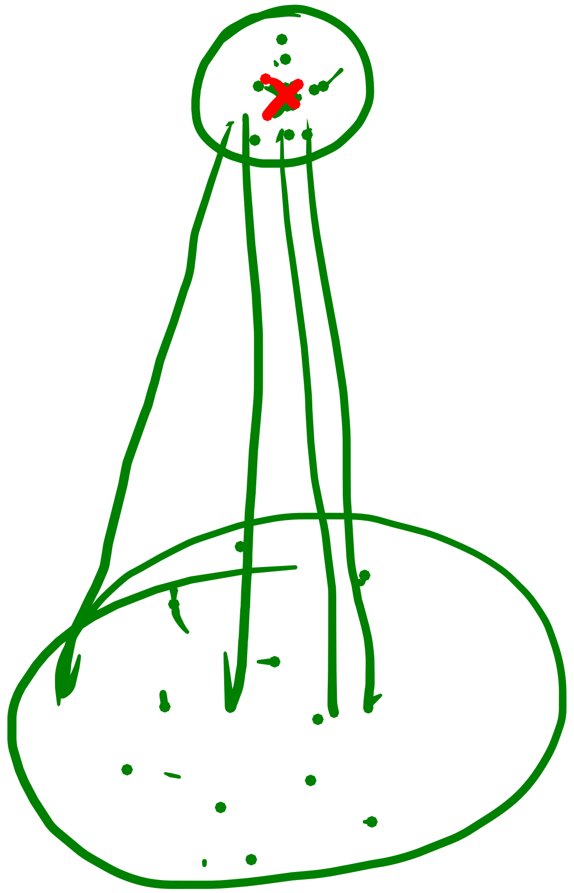
Advantage of K-mediod

1. Robust to outlier
2. Manhatten or cosine metric can be used for calculating dist.

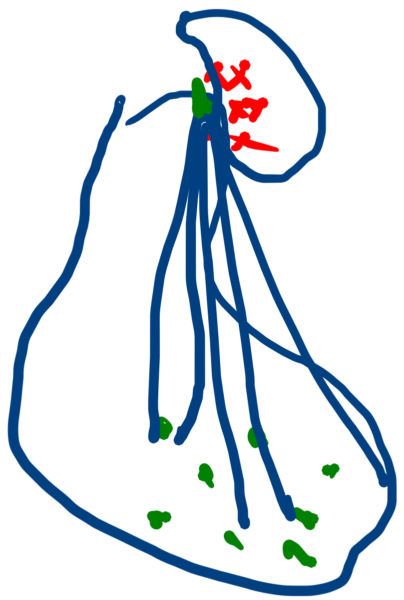
Disadvantage

1. Time complexity
2. Not suitable for non-spherical clusters.

$$S.H = \frac{b-a}{\max(b,a)} = \frac{b-0}{b} = 1$$



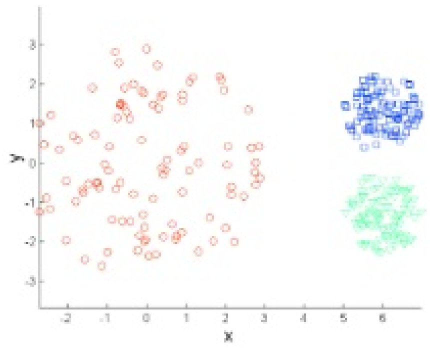
$$\boxed{b \gg a}$$



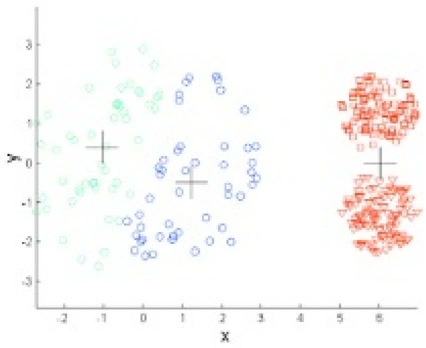
$$a \gg b$$

$$\frac{0-a}{\max(0,a)} = \frac{-a}{a} = -1$$

Limitations of K-means: Differing Density

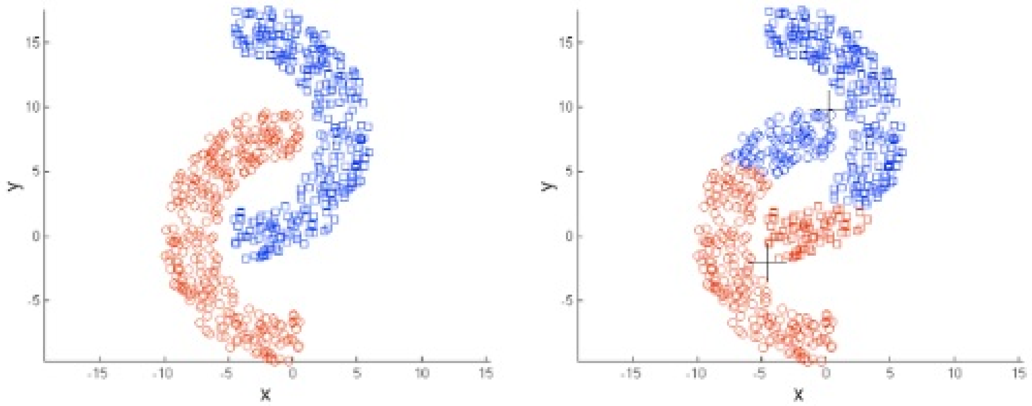


Original Points



K-means (3 Clusters)

Limitations of K-means: Non-globular Shapes



Kmeans → No of clusters

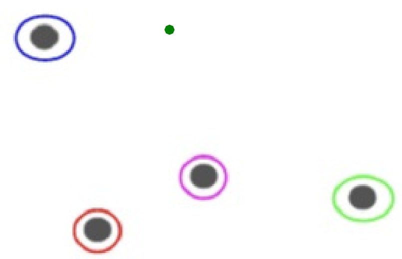
Hierarchial clustering

1. Agglomerative
   ↳ bottom up

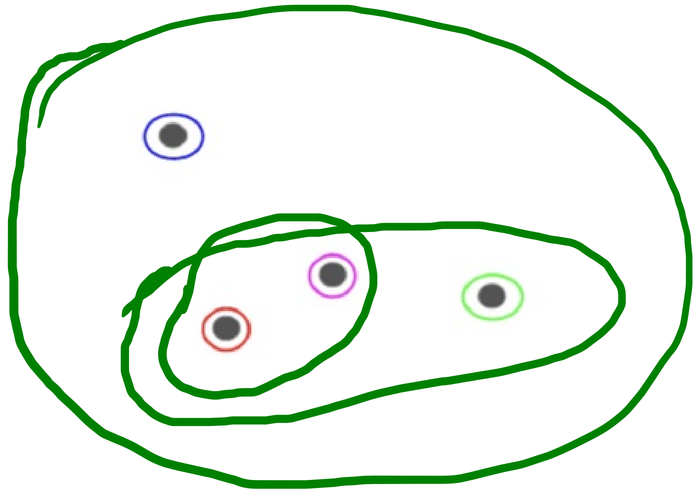
2. Divisive
   ↳ top-down

Agglomerative

1. Each point as a cluster



2. At each iteration, we merge closest pair of cluster & repeat until 1 cluster is left



Divisive

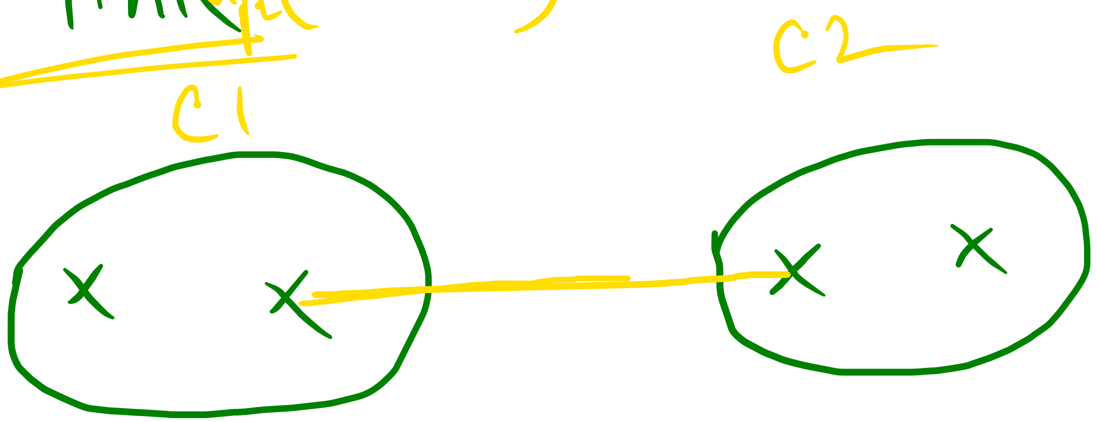
1. Exact opposite to Agglomerative

Algo (Agglomerative)

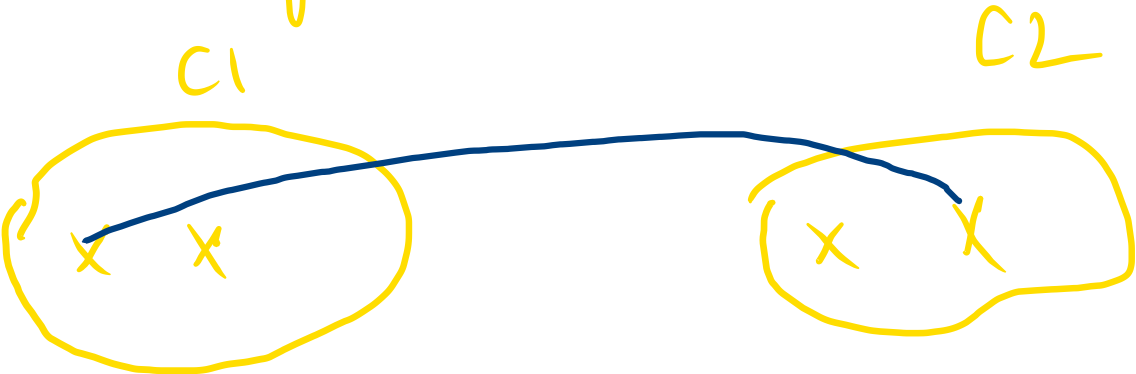
1. Assign each point in cluster (n_datapoints → n_cluster)
2. Compute Proximity matrix
3. Repeat until single cluster
   a. Merge the closest Clusters
   b. Update the proximity matrix

Linkage

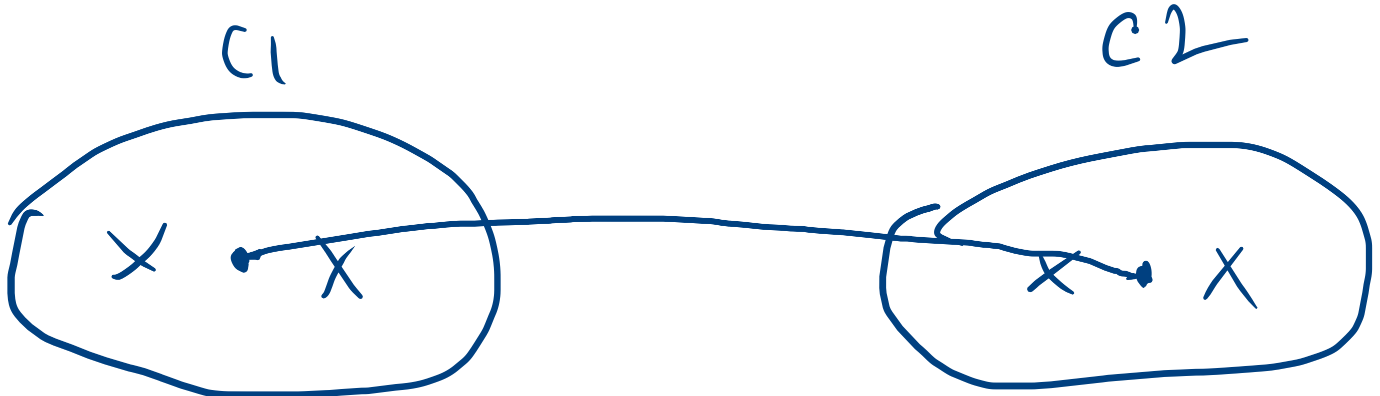
- Single linkage (MIN)

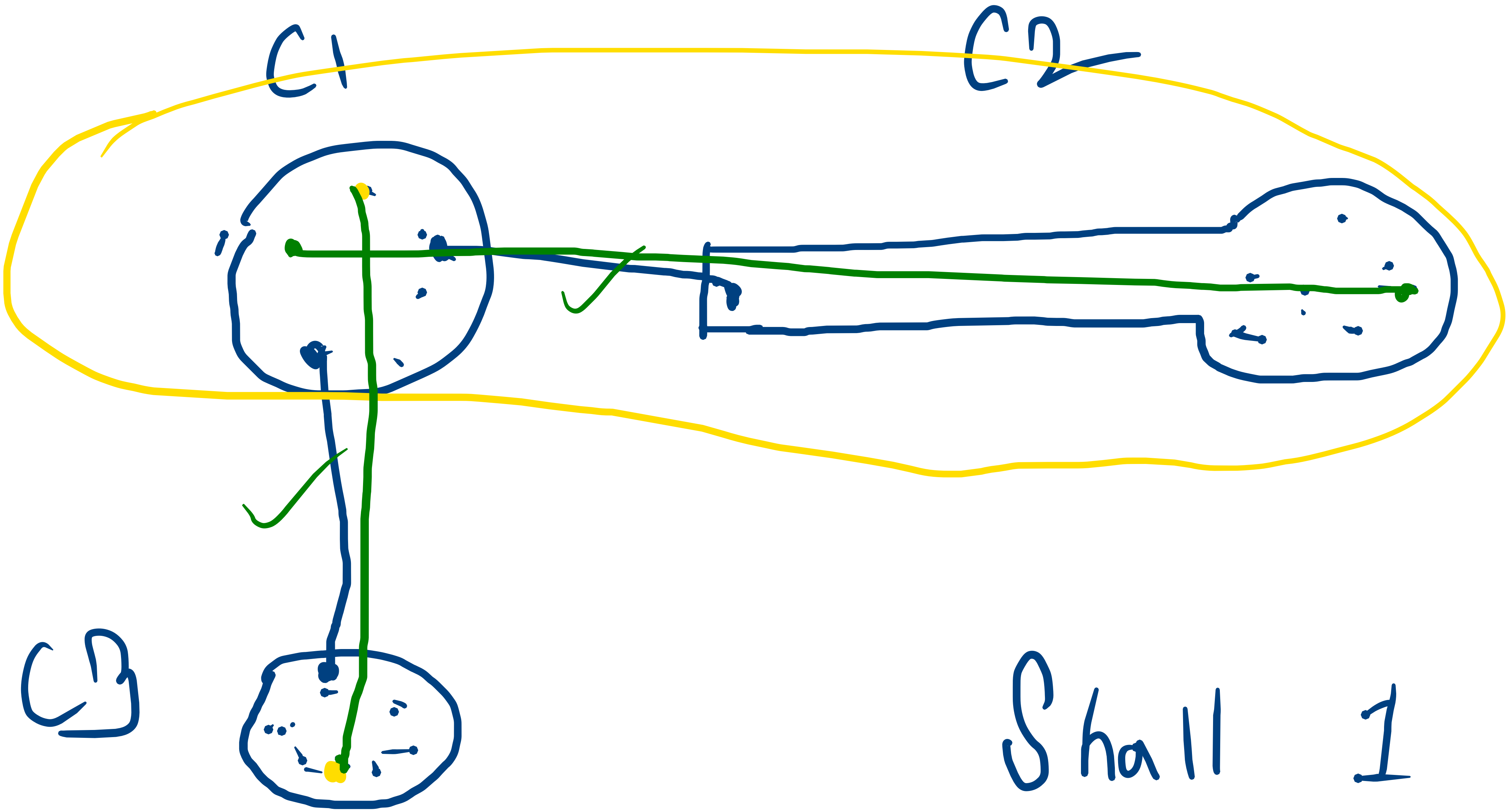


- Complete linkage



- Centroid linkage





Shall 1 j

C1 & C2

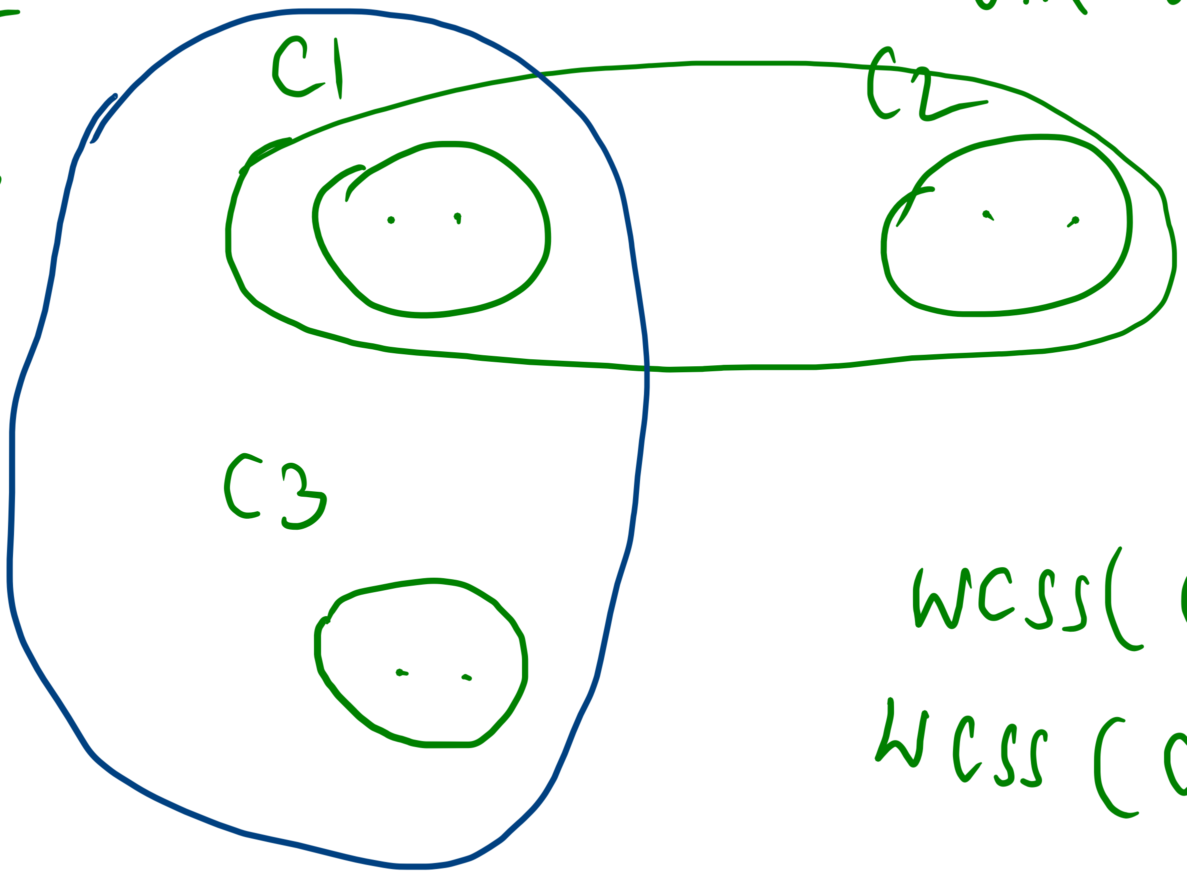
C1 & C3

Cross linkage

Ward Linkage → WCSS (Within Cluster Sum of Square)

C1 & C2

C1 & C3



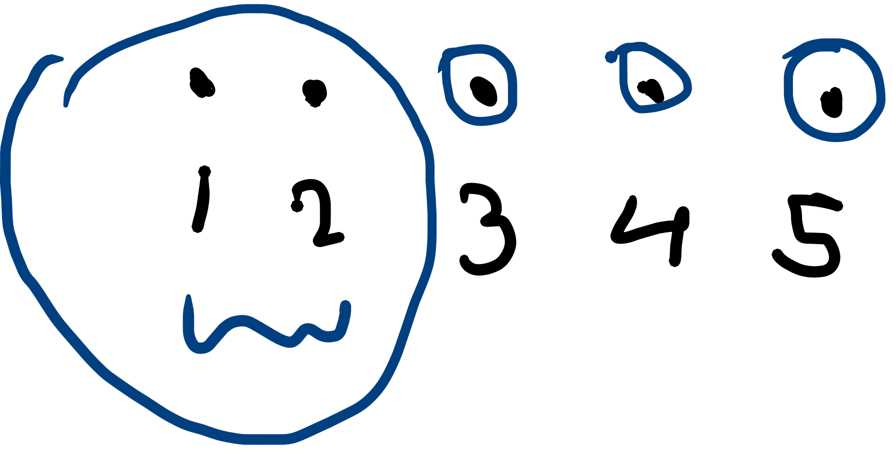
WCSS(C1) WCSS(C2)

WCSS(Combined C1 + C2)

$$WCSS_{increase}^{C1C2} = WCSS(\text{Combine } C1 + C2) - WCSS(C1) - WCSS(C2)$$

$$WCSS_{increase}^{C1C3} = WCSS(\text{Combine } C1 + C3) - WCSS(C1) - WCSS(C3)$$

Break until 10:15pm



Student_ID	Marks
1	10
2	7
3	28
4	20
5	35

Proximity $\longrightarrow$ Distance b/w clusters

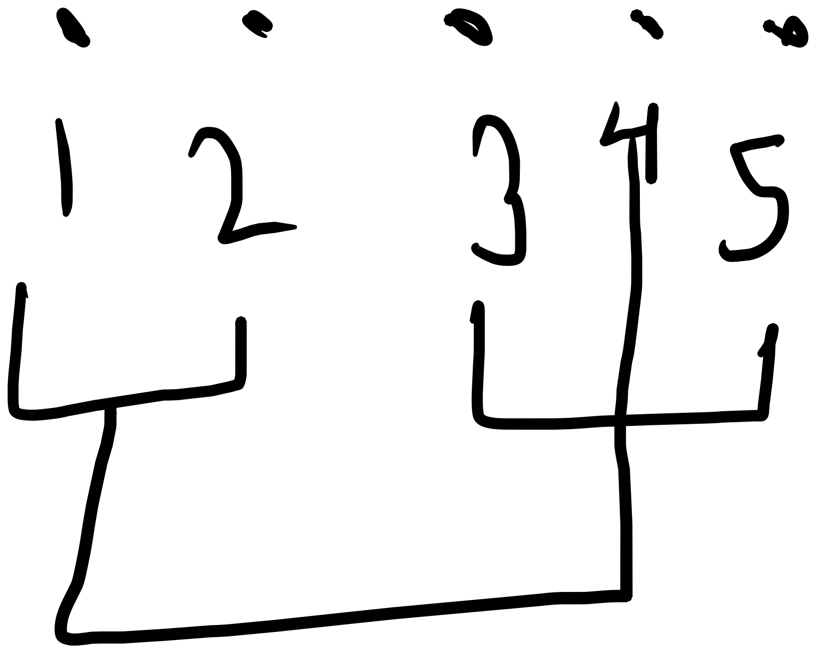
ID	1	2	3	4	5
1	0	3	18	10	25
2	3	0	21	13	28
3	18	21	0	8	7
4	10	13	8	0	15
5	25	28	7	15	0

1 2 3 4 5

Student_ID	Marks
(1,2)	10
3	28
4	20
5	35

Proximity

ID	(1,2)	3	4	5
(1,2)	0	18	10	25
3	18	0	8	7
4	10	8	0	15
5	25	7	15	0

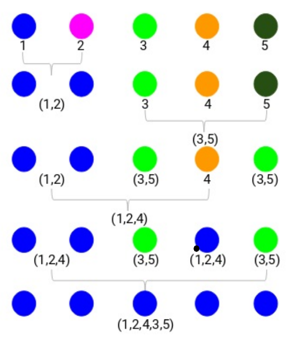


(1,2)	10
(3,5)	35
4	20

	(1,2)	(3,5)	4
(1,2)	0	25	10
(3,5)	25	0	15
4	10	15	0

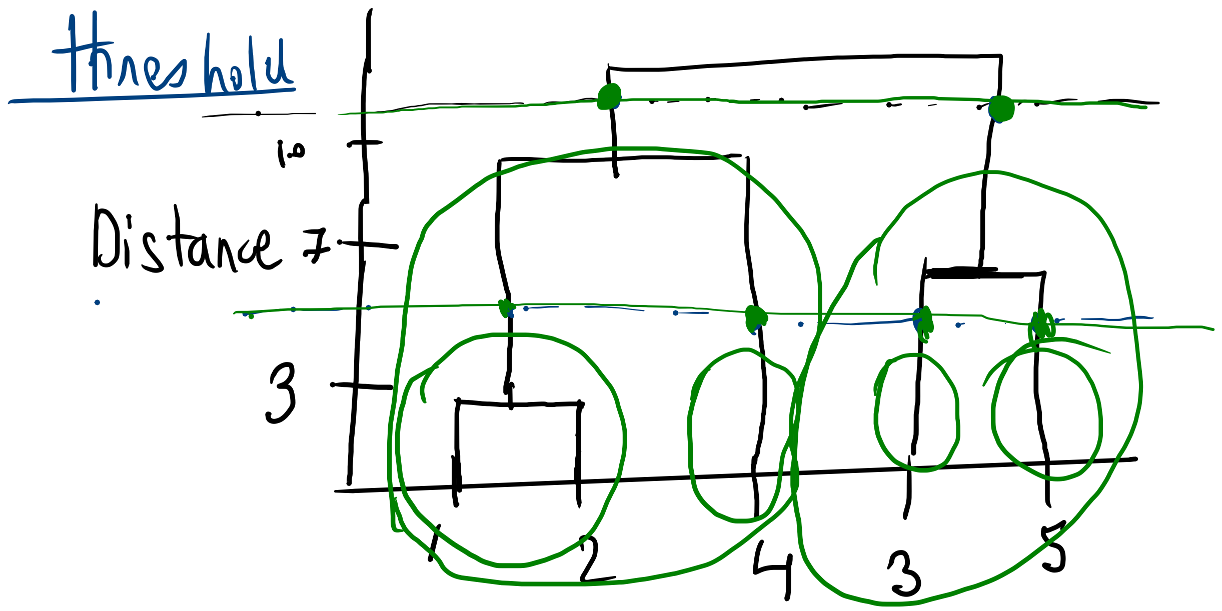
$$\sqrt{d^2}$$

$$|d|$$



Dendrogram

→ Tree like diagram that records the sequence of merges



1

→ No of cluster = No of vertical lines intersected by threshold

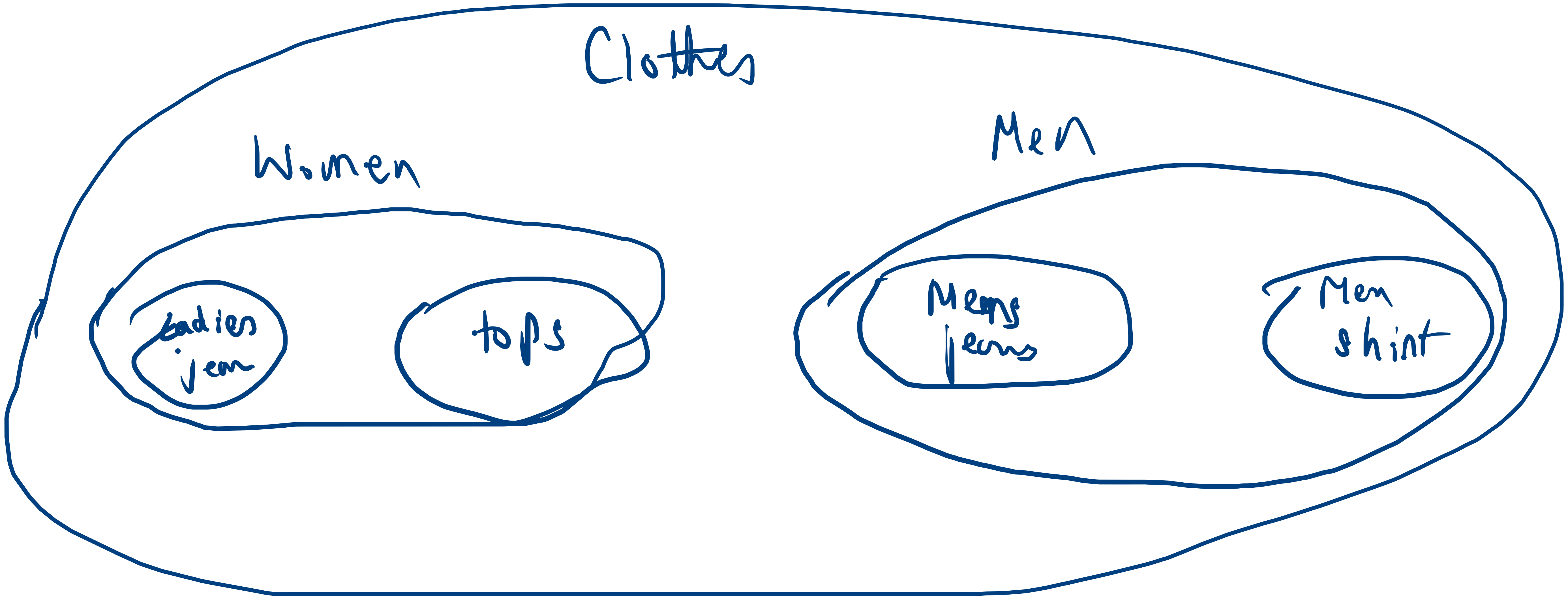
2

More the dist. b/w verticle line more is the dist b/w clusters

Advantage

1. No need to decide no of cluster

2. Amazing for EDA



Disadvantage

1. Sensitive to the choice of linkage
2. Time complexity is higher
3. Only be used for offline

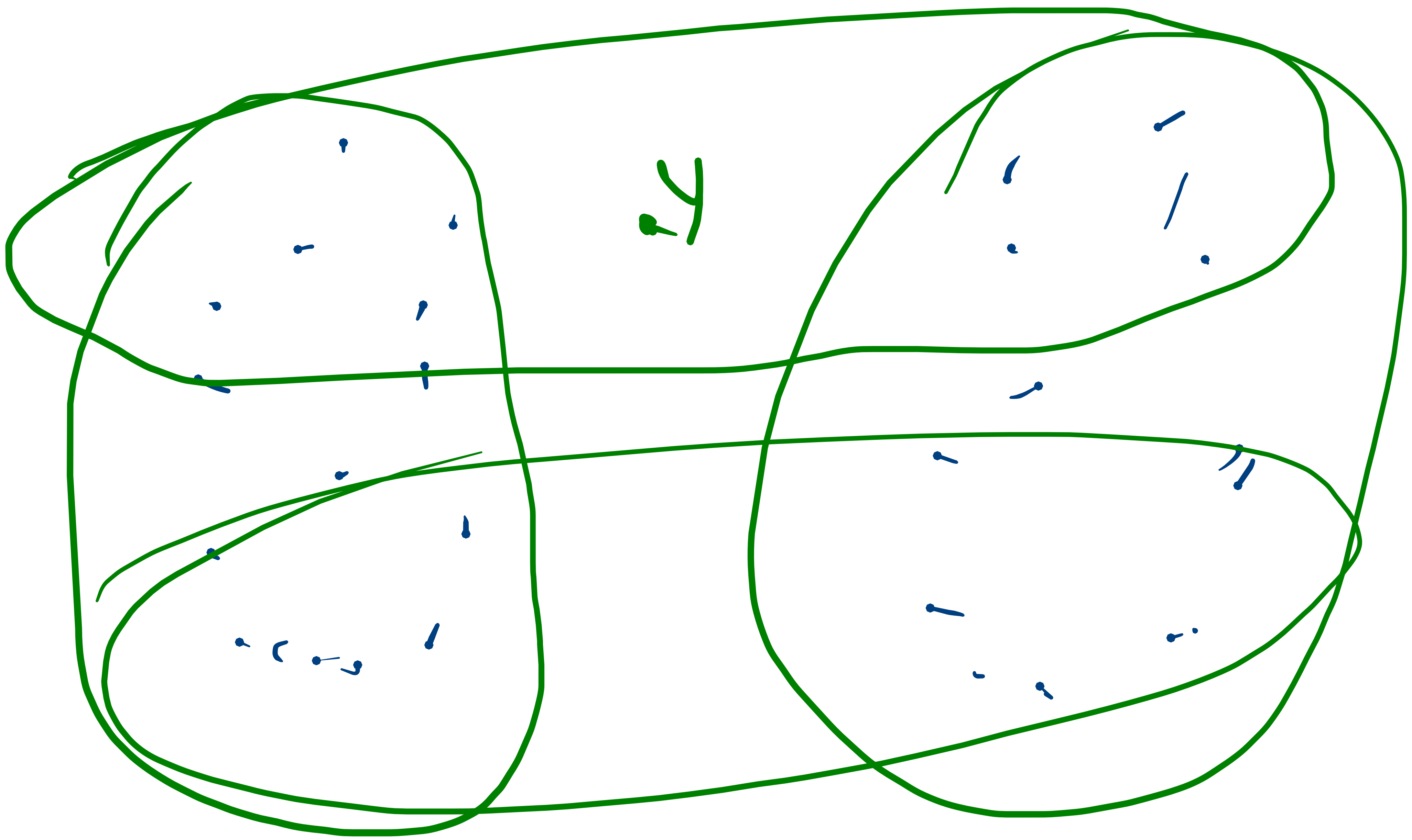
K-means

predict

→ Assign Y to smallest dist b/w C_1 & C_2

Hierarchical

→ No predict method



K-means	Hieanchial clusthing
1) No of cluster need to be defined	-> Need not be defined
-> Predict is available	-> No predict
-> Use mean on centroid	-> Based on linkage
-> Spherical clusters	-> Need not be spherical
-> Faster	-> Slower

$O(n^3)$

$\to O(n^2 \log n)$

Space

$\therefore O(n^2)$