Hierarchical Clustering II

AW

Lecture Overview

1. Recap
2. Max (Complete Graph)
3. Group Average
4. Centroid Distance
5. DBSCAN

Strengths of Hierarchical Clustering

- Do not have to assume any particular number of clusters
  - Any desired number of clusters can be obtained by ‘cutting’ the dendogram at the proper level
- They may correspond to meaningful taxonomies
  - Example in biological sciences (e.g., animal kingdom, phylogeny reconstruction, …)

Hierarchical Clustering

- Two main types of hierarchical clustering:
  1. 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
  2. Divisive:
     - Start with one, all-inclusive cluster
     - At each step, split a cluster until each cluster contains a point (or there are k clusters)
- Traditional hierarchical algorithms use a similarity or distance matrix
  - Merge or split one cluster at a time

Agglomerative Clustering Algorithm

- More popular hierarchical clustering technique
- Basic algorithm is straightforward:

1. Compute the proximity matrix

   Let each data point be a cluster

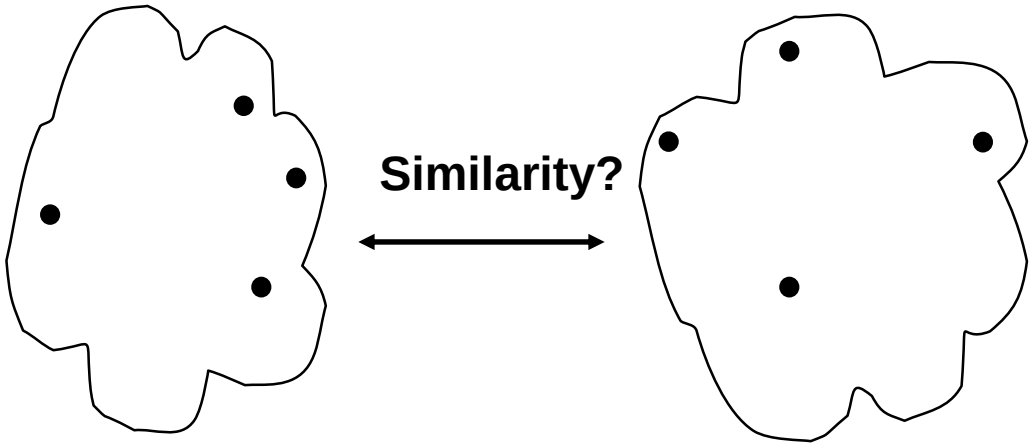
2. **Repeat**

   i. Merge the two closest clusters

   ii. Update the proximity matrix

   Until only a single cluster remains

How to Define Inter-Cluster Similarity?

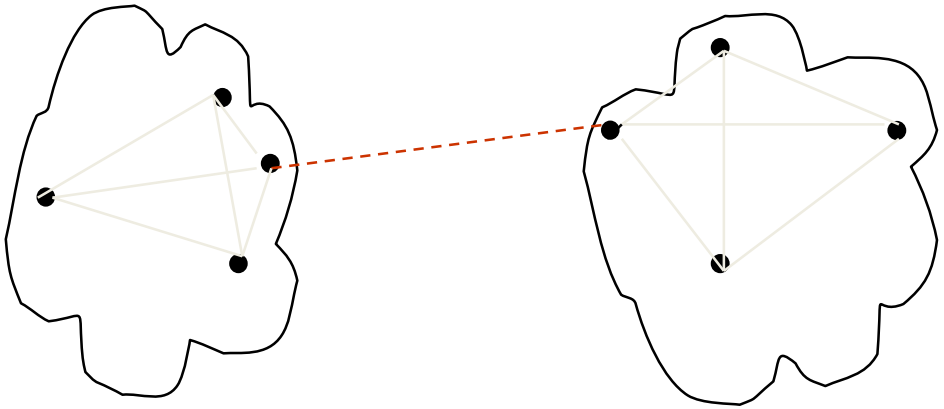


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						

Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

Cluster Similarity – Min



Find shortest edge (line) between members of different clusters

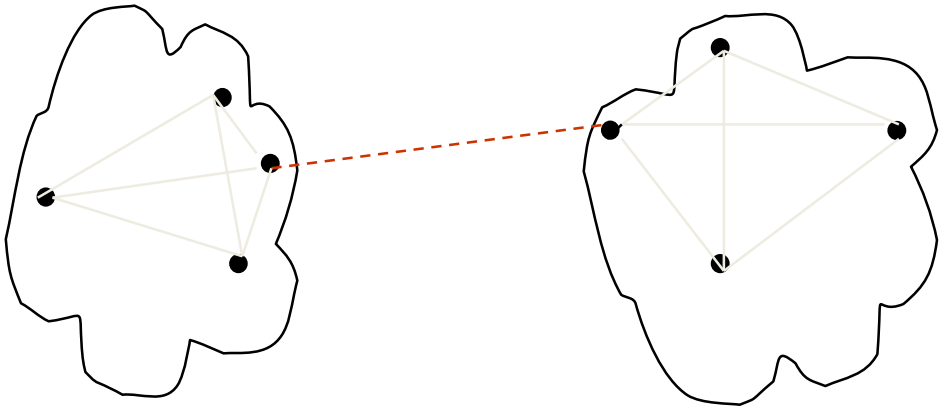
	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						

Proximity Matrix

Lecture Overview

1. Recap
2. Max (Complete Graph)
3. Group Average
4. Centroid Distance
5. DBSCAN

Cluster Similarity – Max (Complete Graph)

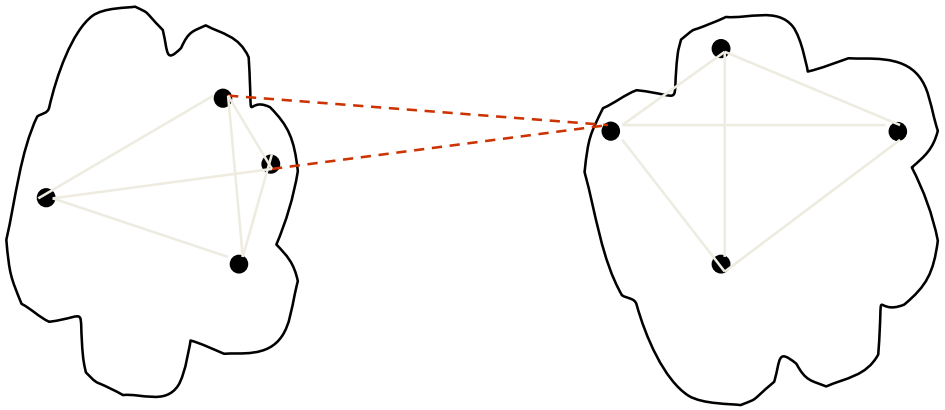


- MAX – Graph Clique
  - fill in edges according to length

	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						

Proximity Matrix

Cluster Similarity – Max (Complete Graph)

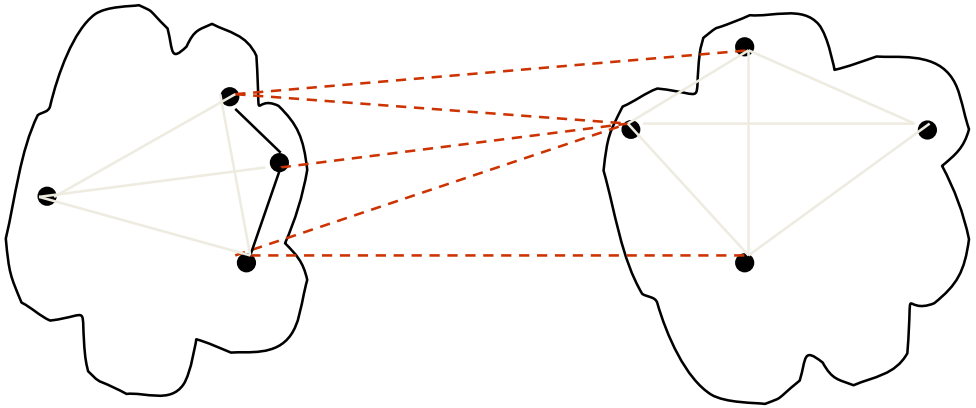


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						

Proximity Matrix

- MAX – Graph Clique
  - fill in edges according to length

Cluster Similarity – Max (Complete Graph)

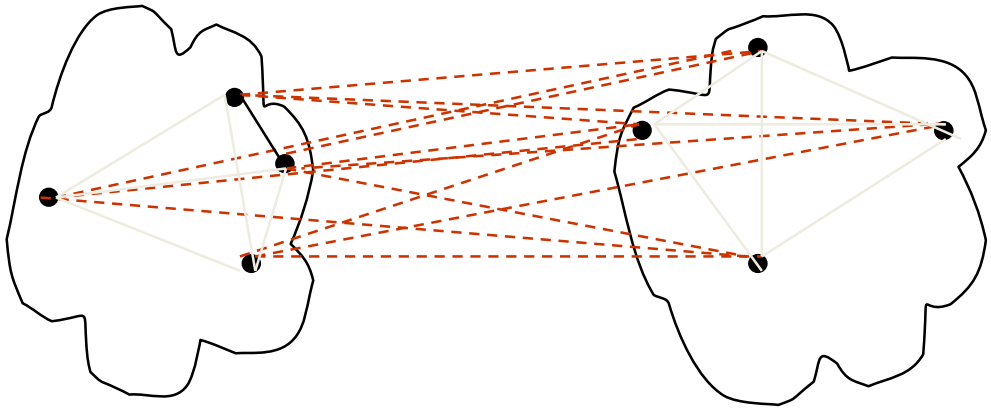


- MAX – Graph Clique
  - fill in edges according to length

	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						

Proximity Matrix

Cluster Similarity – Max (Complete Graph)

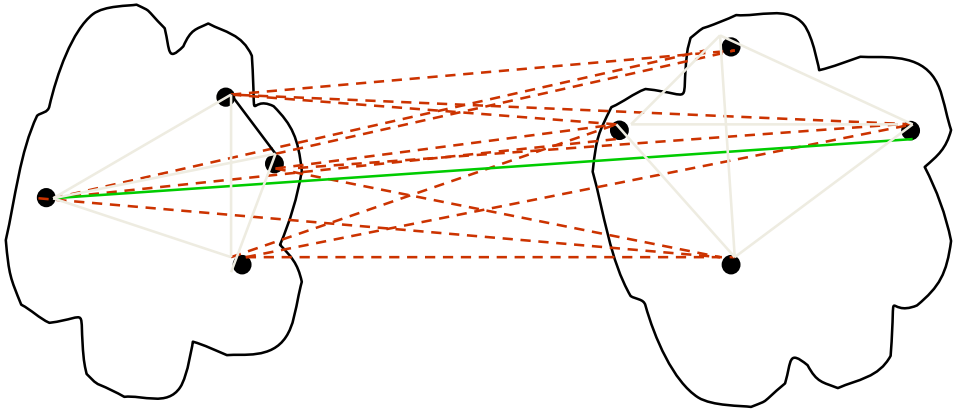


- MAX – Graph Clique
  - fill in edges according to length

	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						

Proximity Matrix

Cluster Similarity – Max (Complete Graph)



- MAX – Graph Clique Cluster:
  - **The first set of point forming clique**

	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						

Proximity Matrix

MAX Dendrogram Computation

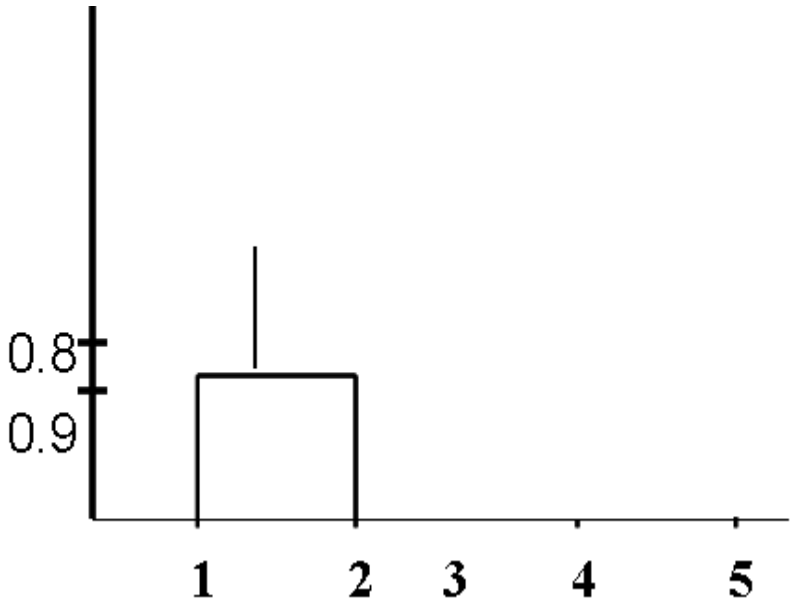
- Similarity of two clusters is based on the two least similar (most distant) points in the different clusters (mini max)
  - Determined by one pair of points, i.e., by one link in the proximity graph.

	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00

MAX Dendrogram Computation

- Similarity of two clusters is based on the two least similar (most distant) points in the different clusters (mini max)
  - Determined by one pair of points, i.e., by one link in the proximity graph.

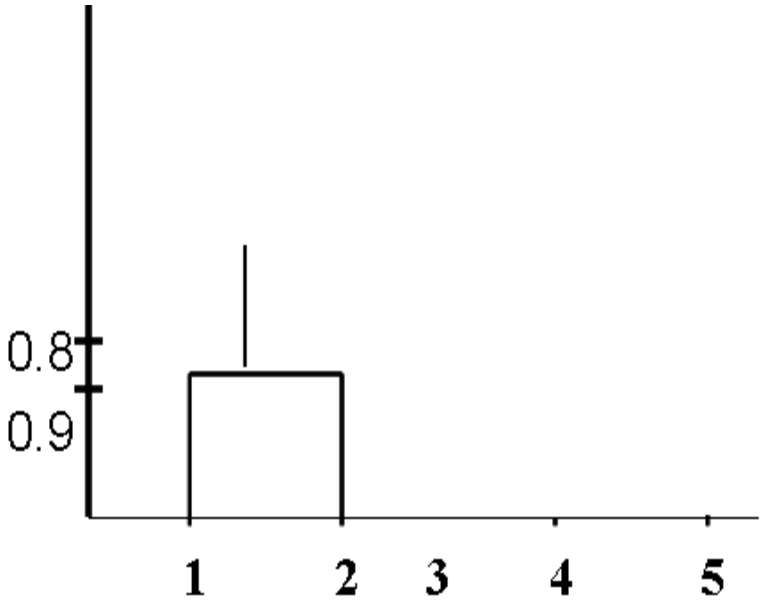
	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



MAX Dendrogram Computation

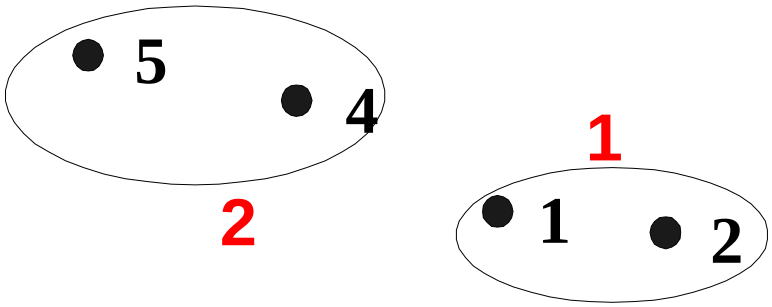
- Similarity of two clusters is based on the two least similar (most distant) points in the different clusters (mini max)
  - Determined by one pair of points, i.e., by one link in the proximity graph.

	I1	I2	I3	I4	I5
I1	1.00	1.00	0.10	0.60	0.20
I2	1.00	1.00	0.10	0.60	0.20
I3	0.10	0.10	1.00	0.40	0.30
I4	0.60	0.60	0.40	1.00	0.80
I5	0.20	0.20	0.30	0.80	1.00

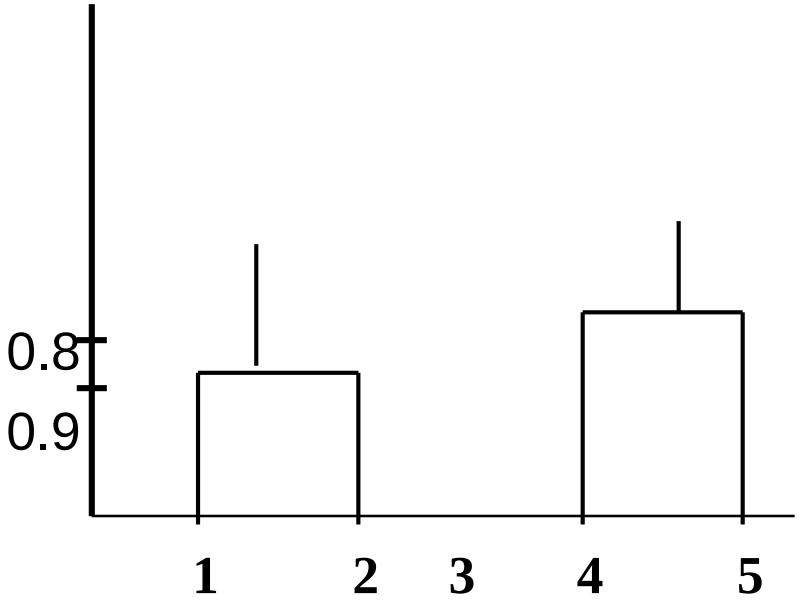


MAX Dendrogram Computation

- Similarity of two clusters is based on the two least similar (most distant) points in the different clusters (mini max)
  - Determined by one pair of points, i.e., by one link in the proximity graph.

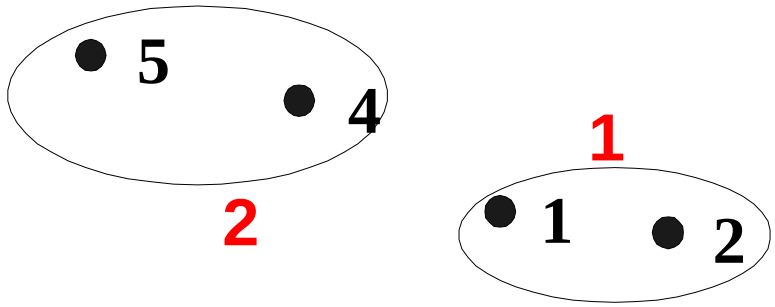


	I1	I2	I3	I4	I5
I1	1.00	1.00	0.10	0.60	0.20
I2	1.00	1.00	0.10	0.60	0.20
I3	0.10	0.10	1.00	0.40	0.30
I4	0.60	0.60	0.40	1.00	0.80
I5	0.20	0.20	0.30	0.80	1.00

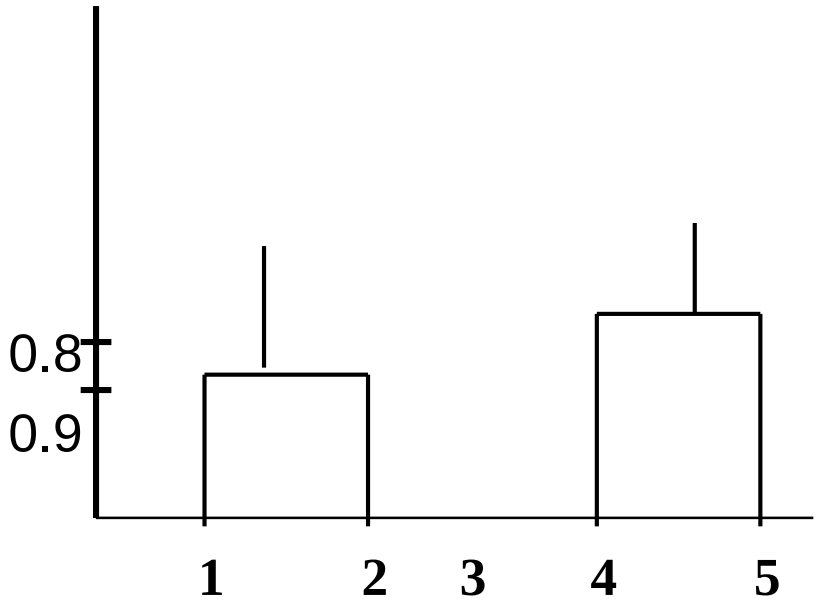


MAX Dendrogram Computation

- Similarity of two clusters is based on the two least similar (most distant) points in the different clusters (mini max)
  - Determined by one pair of points, i.e., by one link in the proximity graph.

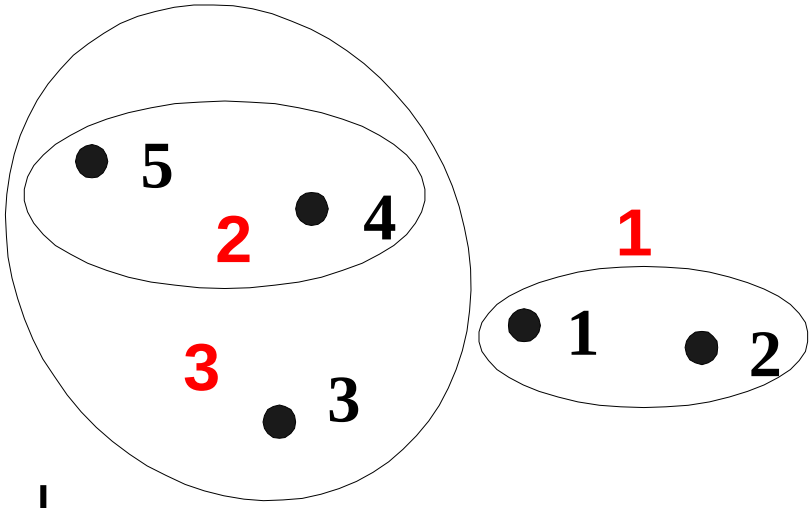


	I1	I2	I3	I4	I5
I1	1.00	1.00	0.10	0.20	0.20
I2	1.00	1.00	0.10	0.20	0.20
I3	0.10	0.10	1.00	0.30	0.30
I4	0.20	0.20	0.30	1.00	1.00
I5	0.20	0.20	0.30	1.00	1.00

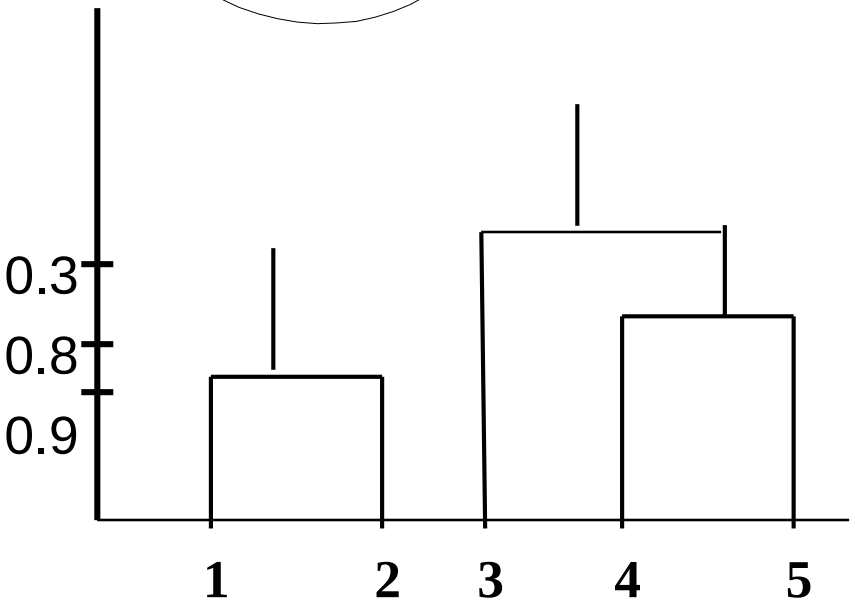


MAX Dendrogram Computation

- Similarity of two clusters is based on the two least similar (most distant) points in the different clusters (mini max)
  - Determined by one pair of points, i.e., by one link in the proximity graph.

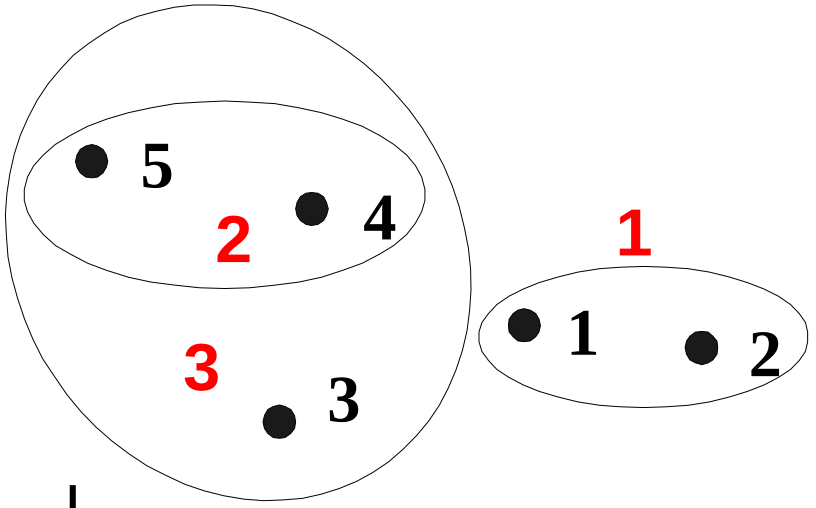


	I1	I2	I3	I4	I5
I1	1.00	1.00	0.10	0.20	0.20
I2	1.00	1.00	0.10	0.20	0.20
I3	0.10	0.10	1.00	0.30	0.30
I4	0.20	0.20	0.30	1.00	1.00
I5	0.20	0.20	0.30	1.00	1.00

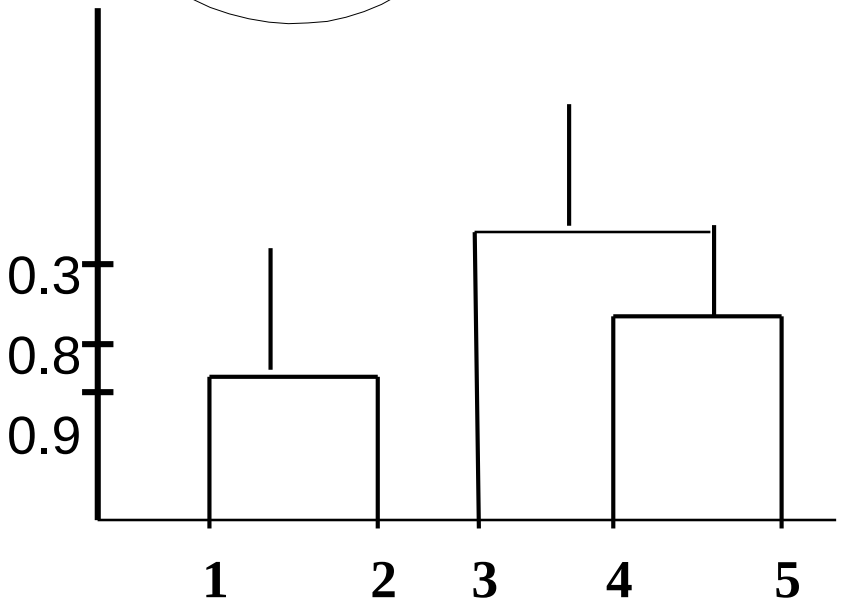


MAX Dendrogram Computation

- Similarity of two clusters is based on the two least similar (most distant) points in the different clusters (mini max)
  - Determined by one pair of points, i.e., by one link in the proximity graph.

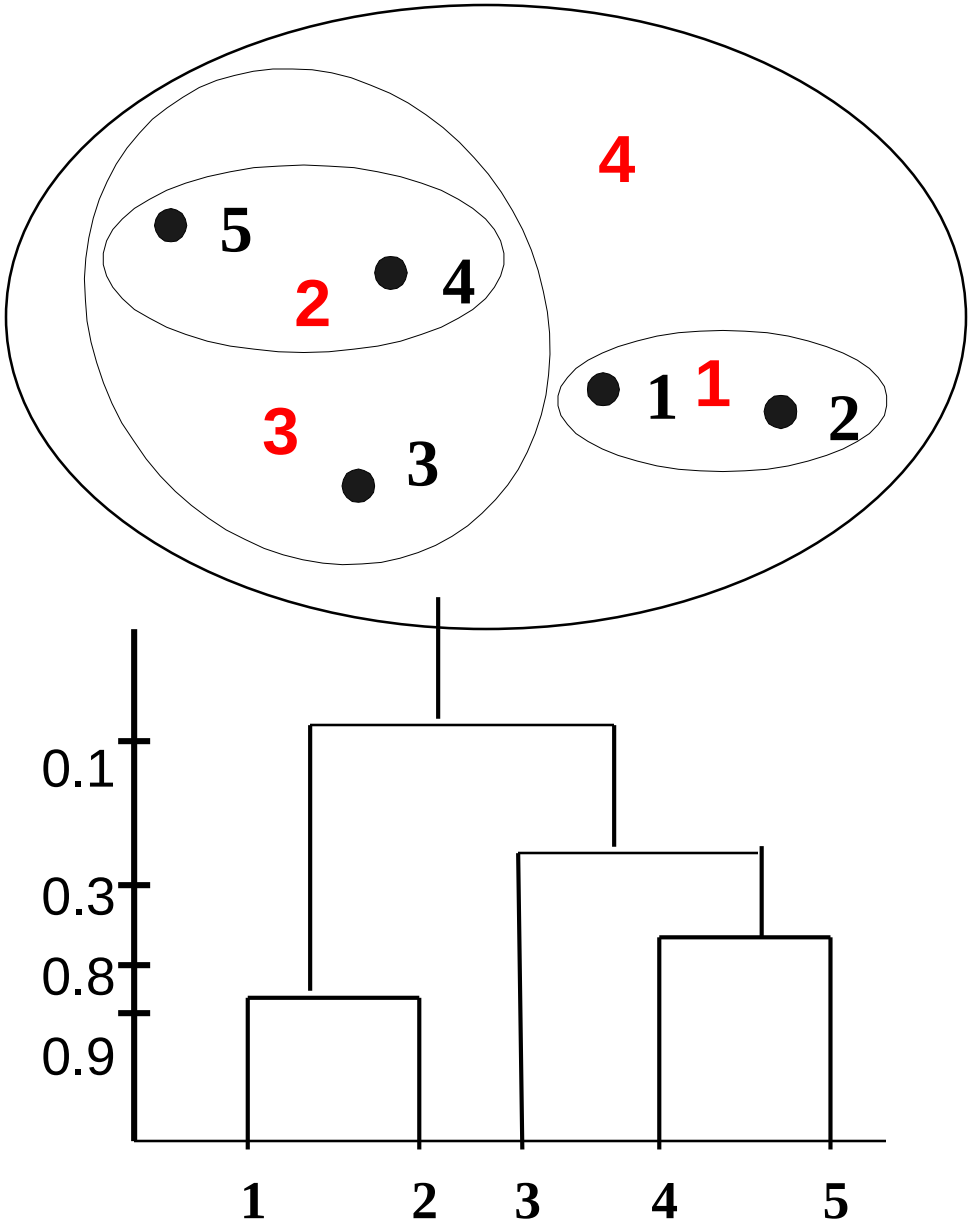


	I1	I2	I3	I4	I5
I1	1.00	1.00	0.10	0.10	0.10
I2	1.00	1.00	0.10	0.10	0.10
I3	0.10	0.10	1.00	1.00	1.00
I4	0.10	0.10	1.00	1.00	1.00
I5	0.10	0.10	1.00	1.00	1.00



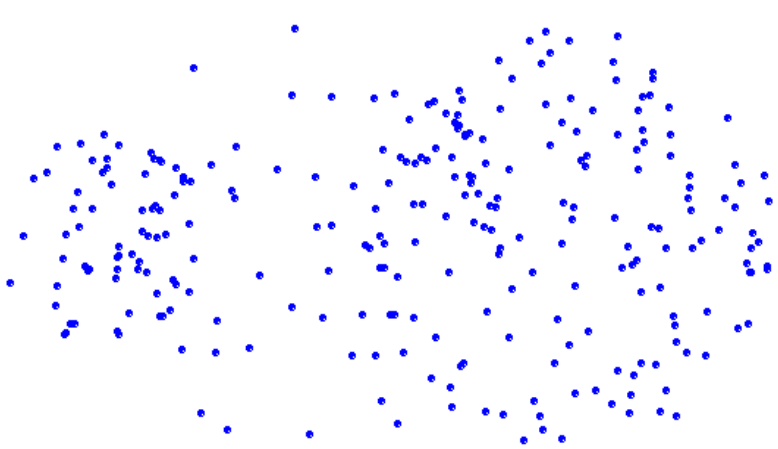
MAX Dendrogram Computation

- Similarity of two clusters is based on the two least similar (most distant) points in the different clusters (mini max)
  - Determined by one pair of points, i.e., by one link in the proximity graph.

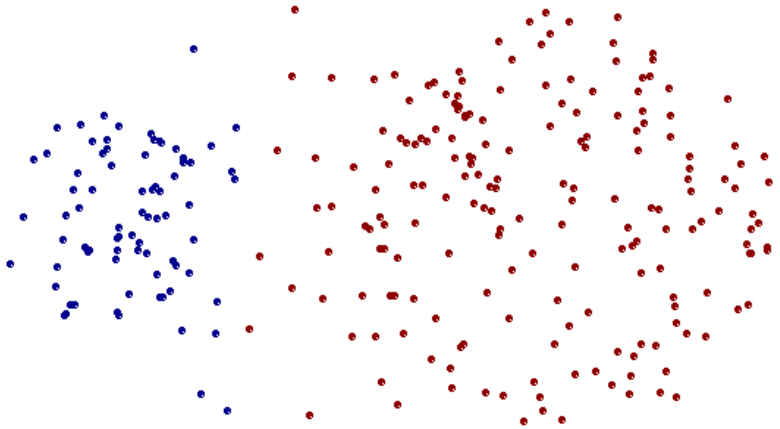


	I1	I2	I3	I4	I5
I1	1.00	1.00	0.10	0.10	0.10
I2	1.00	1.00	0.10	0.10	0.10
I3	0.10	0.10	1.00	1.00	1.00
I4	0.10	0.10	1.00	1.00	1.00
I5	0.10	0.10	1.00	1.00	1.00

Strength of MAX



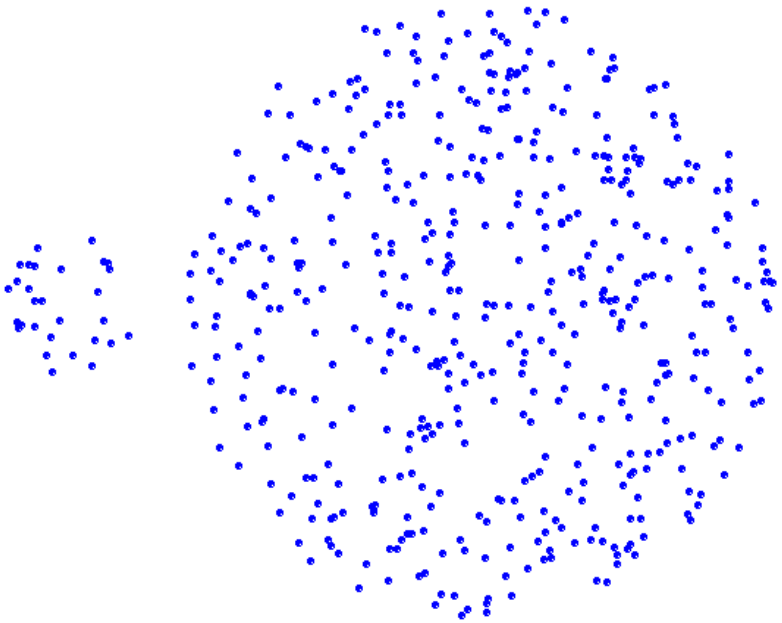
Original Points



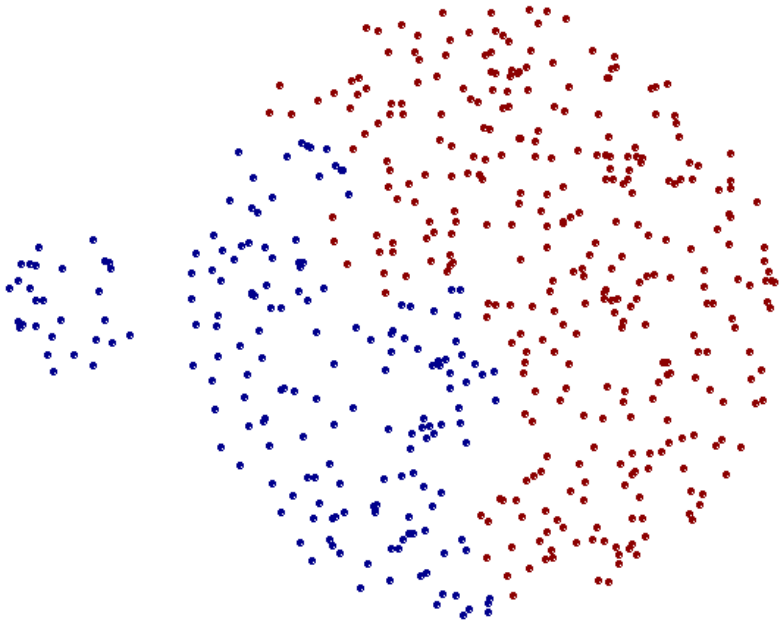
Two Clusters

- Less susceptible to noise and outliers than MIN

Limitations of MAX



Original Points



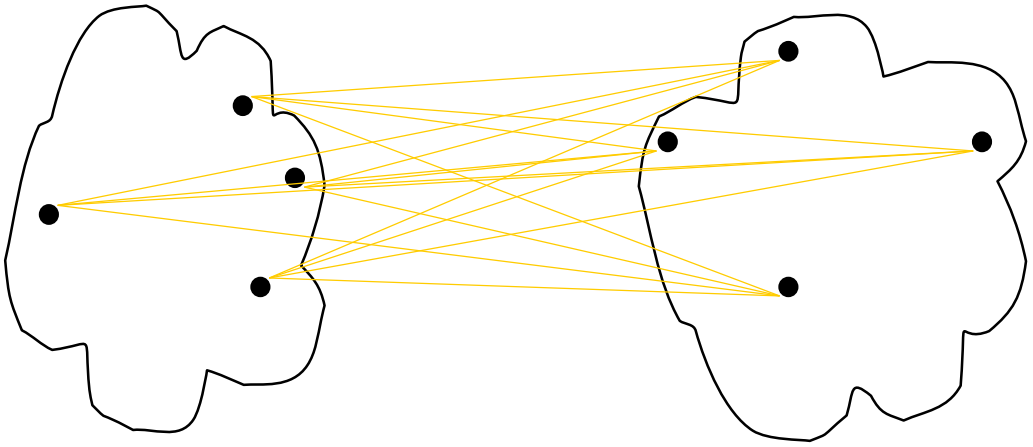
Two Clusters

- Tends to break large clusters
- Biased towards globular clusters

Lecture Overview

1. Recap
2. Max (Complete Graph)
3. Group Average
4. Centroid Distance
5. DBSCAN

Cluster Similarity – Group Average



	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						

Proximity Matrix

- Group Average:
  - Proximity of two clusters C_i and C_j is the average of pairwise proximity between points in the two clusters:

$$prox(C_i, C_j) = \frac{\sum_{\substack{x \in C_i \\ y \in C_j}} Prox\,(x, y)}{|C_i| \times |C_j|}$$

Average Dendrogram Computation

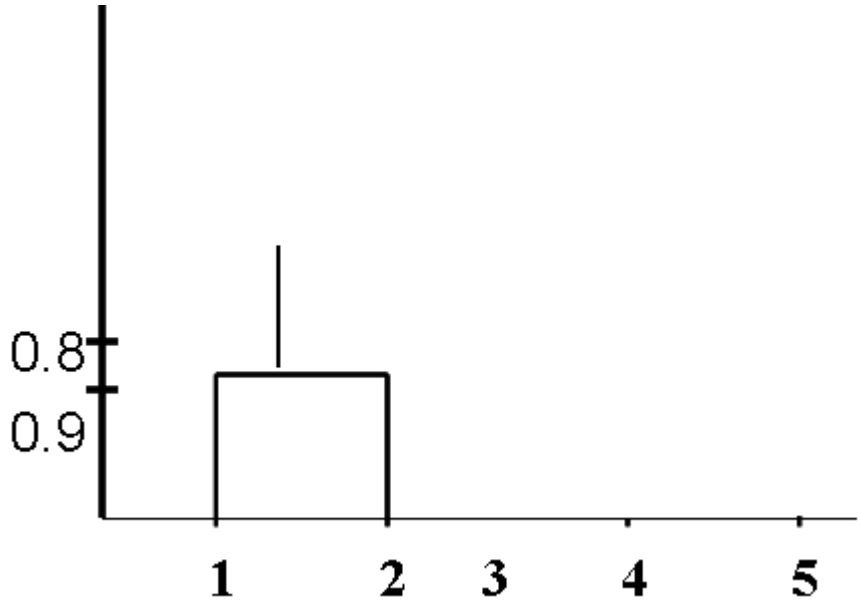
- Proximity of two clusters needs to use average connectivity for scalability since total proximity favors large clusters

	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00

Average Dendrogram Computation

- Proximity of two clusters needs to use average connectivity for scalability since total proximity favors large clusters

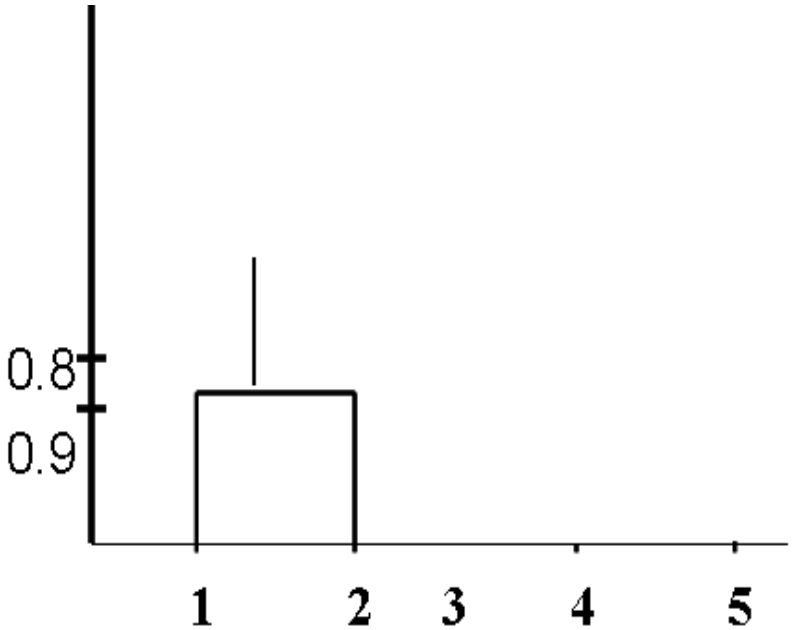
	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



Average Dendrogram Computation

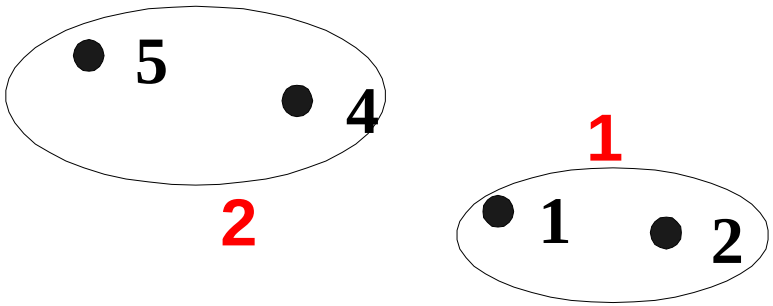
- Proximity of two clusters needs to use average connectivity for scalability since total proximity favors large clusters

	I1	I2	I3	I4	I5
I1	1.000	1.000	0.40	0.625	0.35
I2	1.000	1.000	0.40	0.625	0.35
I3	0.400	0.400	1.00	0.400	0.30
I4	0.625	0.625	0.40	1.000	0.80
I5	0.350	0.350	0.30	0.800	1.00

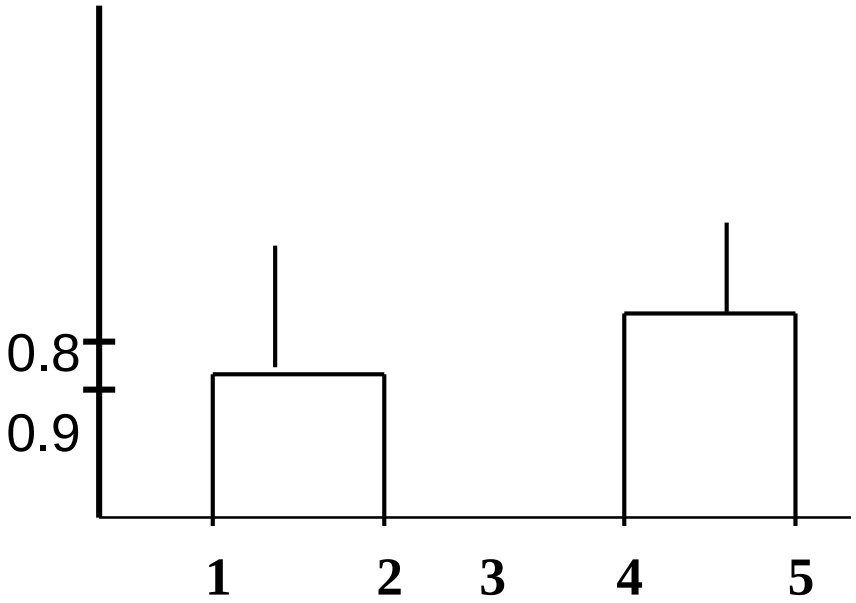


Average Dendrogram Computation

- Proximity of two clusters needs to use average connectivity for scalability since total proximity favors large clusters

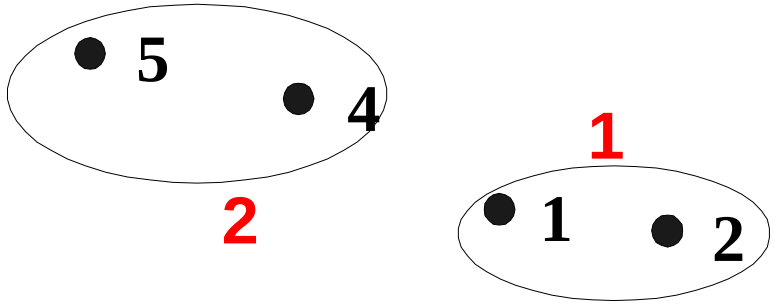


	I1	I2	I3	I4	I5
I1	1.000	1.000	0.40	0.625	0.35
I2	1.000	1.000	0.40	0.625	0.35
I3	0.400	0.400	1.00	0.400	0.30
I4	0.625	0.625	0.40	1.000	0.80
I5	0.350	0.350	0.30	0.800	1.00

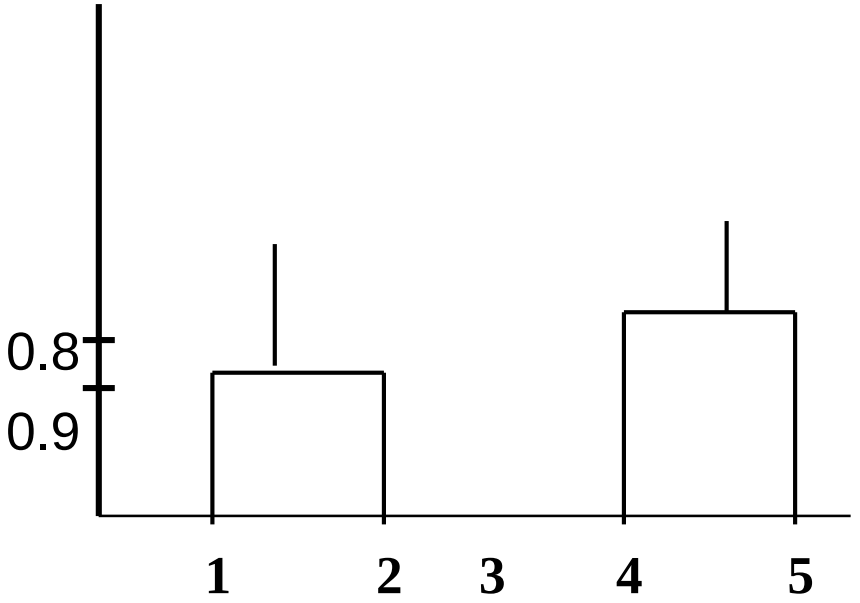


Average Dendrogram Computation

- Proximity of two clusters needs to use average connectivity for scalability since total proximity favors large clusters

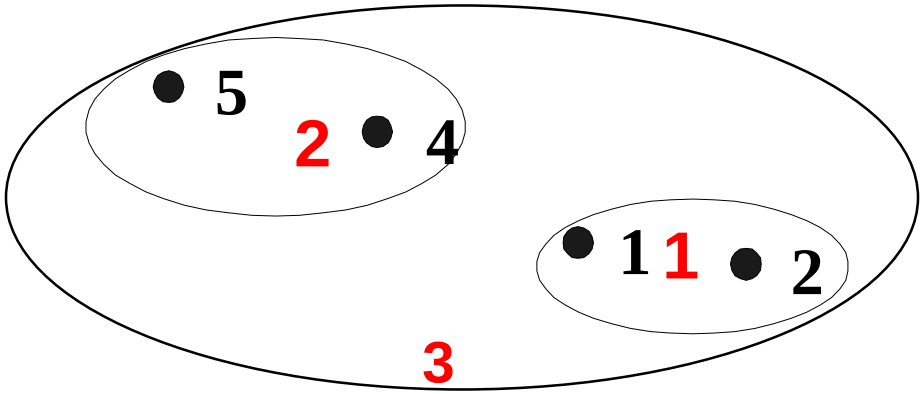


	I1	I2	I3	I4	I5
I1	1.000	1.000	0.400	0.488	0.488
I2	1.000	1.000	0.400	0.488	0.488
I3	0.400	0.400	1.000	0.350	0.350
I4	0.488	0.488	0.350	1.000	1.000
I5	0.488	0.488	0.350	1.000	1.000

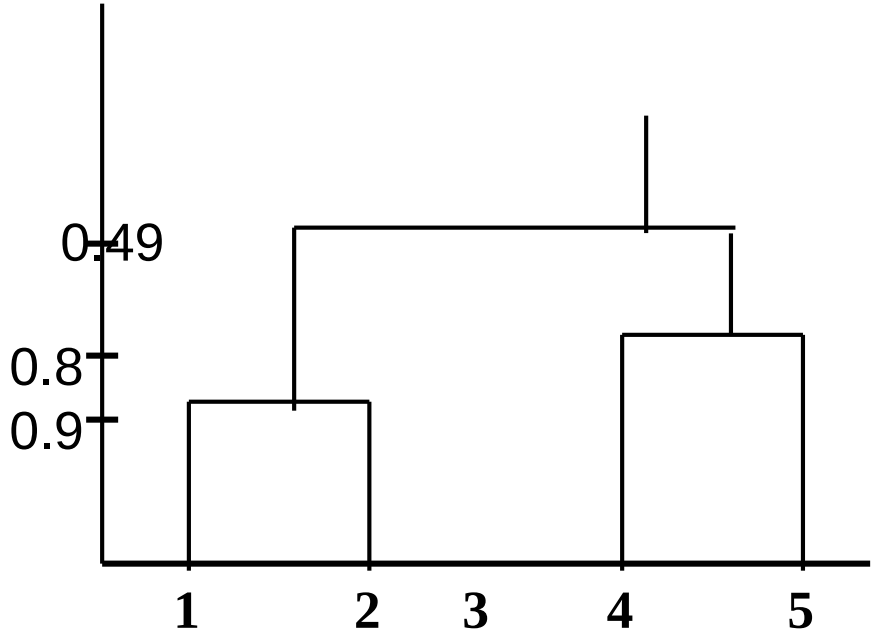


Average Dendrogram Computation

- Proximity of two clusters needs to use average connectivity for scalability since total proximity favors large clusters

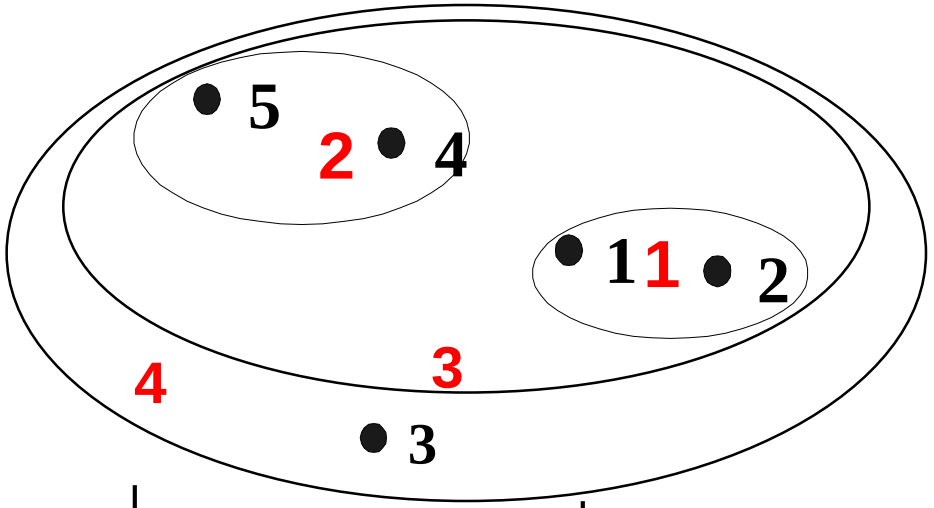


	I1	I2	I3	I4	I5
I1	1.000	1.000	0.375	1.000	1.000
I2	1.000	1.000	0.375	1.000	1.000
I3	0.375	0.375	1.000	0.375	0.375
I4	1.000	1.000	0.375	1.000	1.000
I5	1.000	1.000	0.375	1.000	1.000

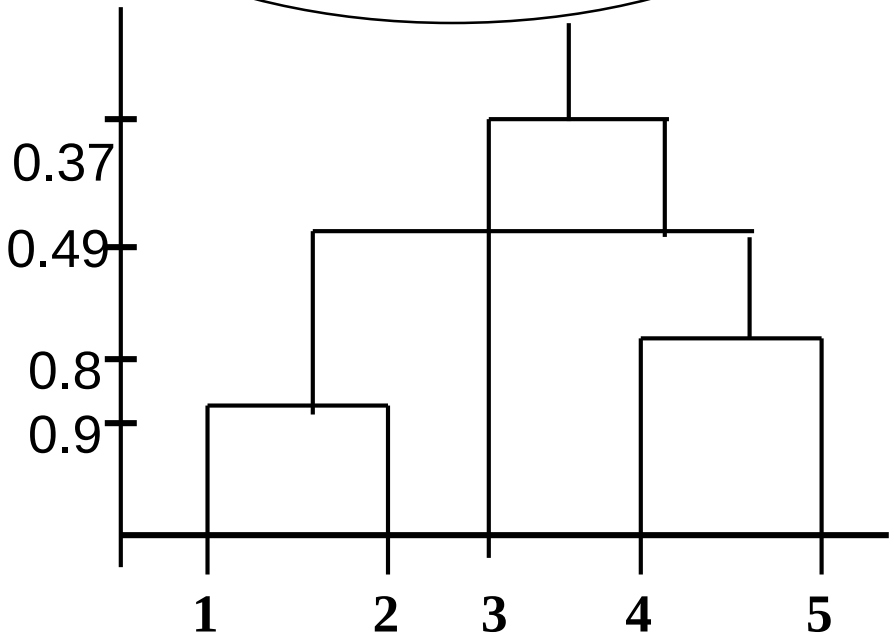


Average Dendrogram Computation

- Proximity of two clusters needs to use average connectivity for scalability since total proximity favors large clusters



	I1	I2	I3	I4	I5
I1	1.000	1.000	0.375	1.000	1.000
I2	1.000	1.000	0.375	1.000	1.000
I3	0.375	0.375	1.000	0.375	0.375
I4	1.000	1.000	0.375	1.000	1.000
I5	1.000	1.000	0.375	1.000	1.000



Average: Strength and Limitations

- Compromise between Single and Complete Link
- Strengths
  - Less susceptible to noise and outliers than Min
  - Does not break large clusters as much as MAX
- Limitations
  - Still biased towards globular clusters

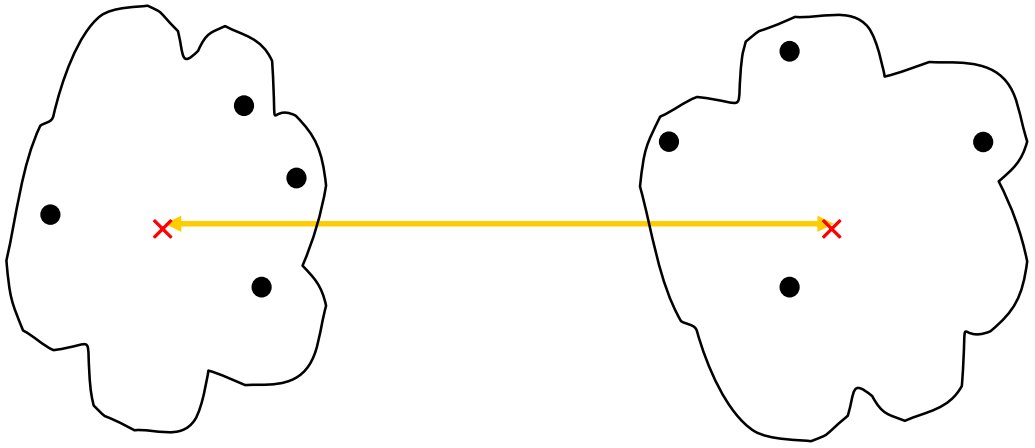
The Clustering w/ min, max, average in **R**

```
library(graphics); attach(iris)
data_iris<-iris[, -5]
cl_sin_s <- hclust(dist(data_iris), method = "single")
#using Euclidean Distance dist
plot(cl_sin_s, hang = -1)
cl_sin_c <- hclust(dist(data_iris), method = "complete")
dev.new()
plot(cl_sin_c, hang = -1)
cl_sin_a <- hclust(dist(data_iris), method = "ave")
dev.new()
plot(cl_sin_a, hang = -1)
dev.off();dev.off();dev.off()
#show classification into 3 classes as needed
plot(cl_sin_s, hang = -1)
rect.hclust(cl_sin_s, 3)
true_cl_s<-cutree(cl_sin_s, 3) #extract 3 level classification
table(true_cl_s,iris$Species) #compare with true classification
```

Lecture Overview

1. Recap
2. Max (Complete Graph)
3. Group Average
4. Centroid Distance
5. DBSCAN

Cluster Similarity – Centroid Distance



	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						

Proximity Matrix

- Distance Between Centroids
  - First need to compute centroids

Distance Between Centroids Dendrogram

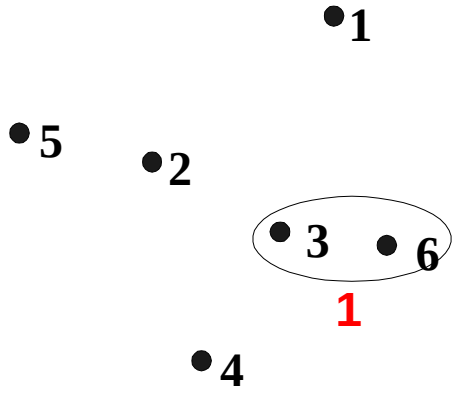
- Centroid distance between clusters C_i and C_j is the distance between the centroid c_i of C_i and the centroid c_j of C_i i.e. $\text{dist}(C_i, C_J) = dist(c_i, c_j)$

- Centroids w.r.t to SSE – means

- Problematic property – non-monotonic:
  - Let cluster $C_{s\ tk}$ at step k be obtained by join of two clusters $\mathrm{C}_{\mathrm{s,t}}^{\mathrm{k}} = C_s^{k-1} \dot{\cup} C_t^{k-1}$ obtained on step $k-1$. The proximity between clusters $C_{s,t}^k$ and $C_j{}^k$ at current step could be bigger that the proximity of either C_s^{k-1} or C_t^{k-1} to $C_j{}^k = C_j^{k-1}$

Distance Between Centroids Dendrogram

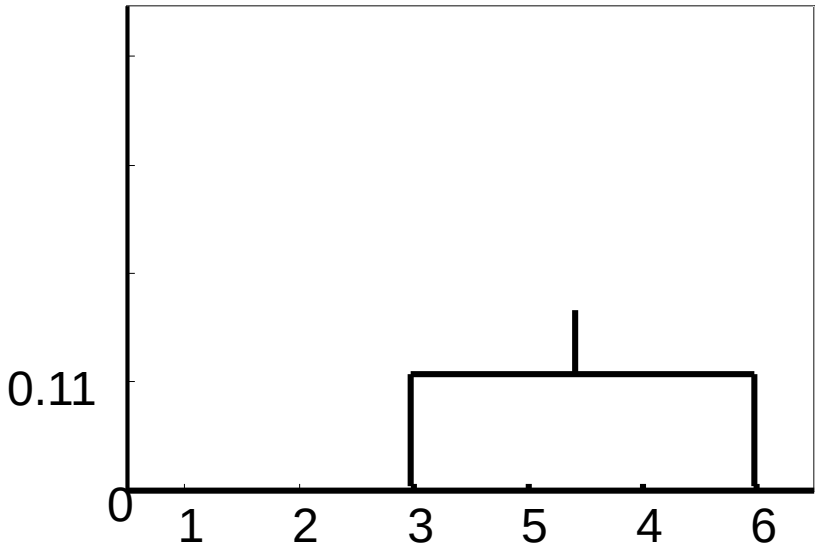
Point	x Coordinate	y Coordinate
p1	0.40	0.53
p2	0.22	0.38
p3	0.35	0.32
p4	0.26	0.19
p5	0.08	0.41
p6	0.45	0.30



Centroid coordinates

	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00

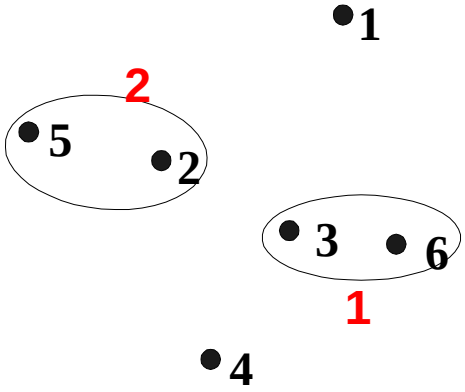
Distance matrix



Distance Between Centroids Dendrogram

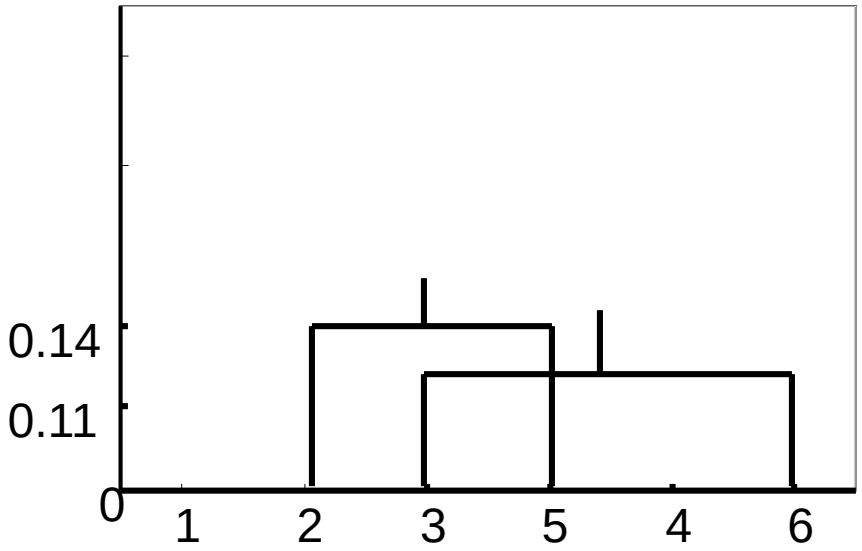
p1	0.4	0.53
p2	0.22	0.38
c1	0.4	0.31
p4	0.26	0.19
p5	0.08	0.41

Centroid coordinates



	p1	p2	c1	p4	p5
p1	0.0	0.24	0.22	0.37	0.34
p2	0.24	0.0	0.19	0.2	0.14
c1	0.22	0.19	0.0	0.18	0.34
p4	0.37	0.2	0.18	0.0	0.29
p5	0.34	0.14	0.34	0.29	0.0

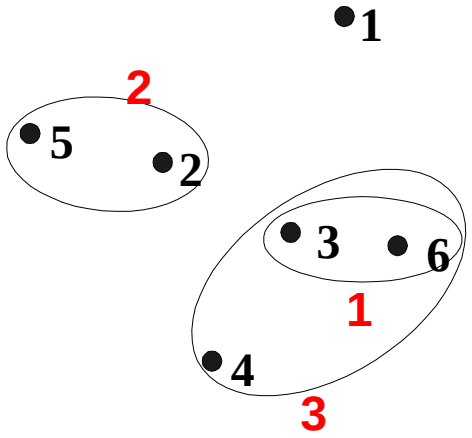
Distance matrix



Distance Between Centroids Dendrogram

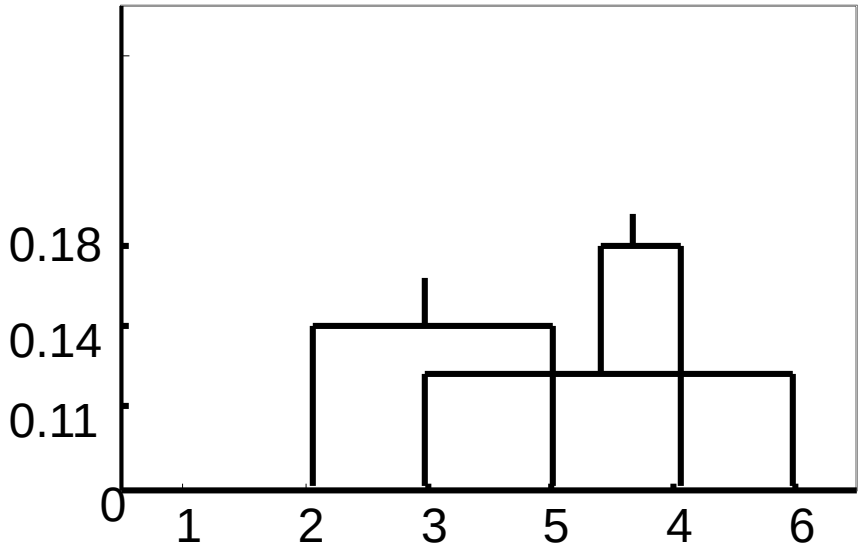
p1	0.4	0.53
c2	0.15	0.395
c1	0.4	0.31
p4	0.26	0.19

Centroid coordinates



	p1	c2	c1	p4
p1	0.0	0.28	0.23	0.37
c2	0.28	0.0	0.26	0.24
c1	0.23	0.26	0.0	0.18
p4	0.37	0.24	0.18	0.0

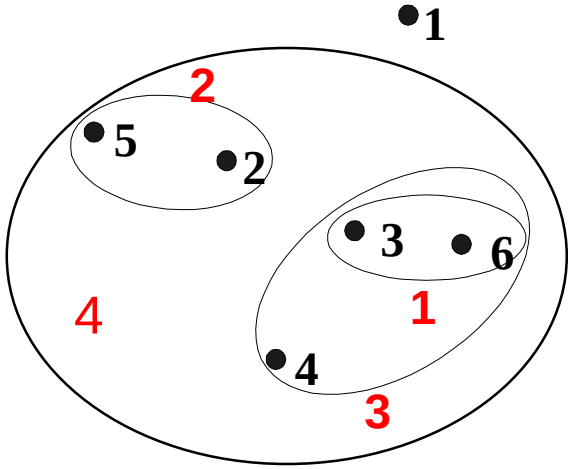
Distance matrix



Distance Between Centroids Dendrogram

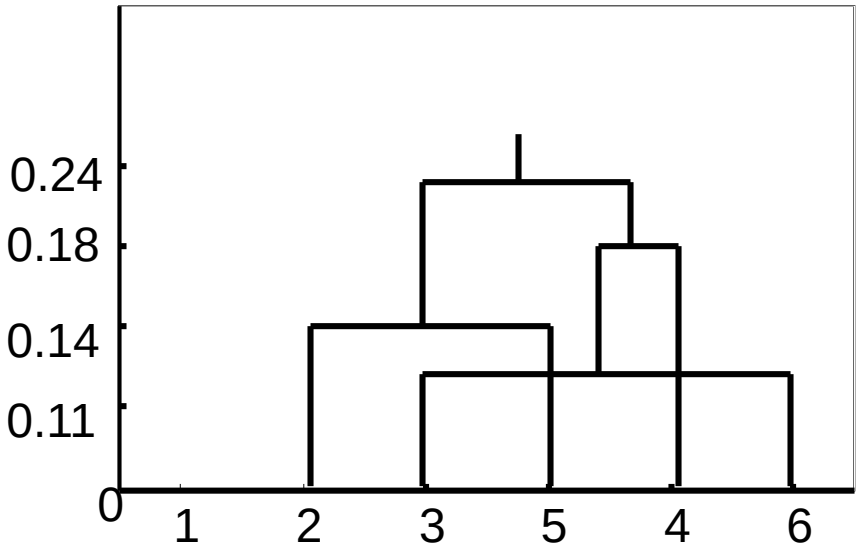
p1	0.4	0.53
c2	0.15	0.395
c3	0.353	0.27

Centroid coordinates



	p1	c2	c3
p1	0.0	0.28	0.27
c2	0.28	0.0	0.24
c3	0.27	0.24	0.0

Distance matrix



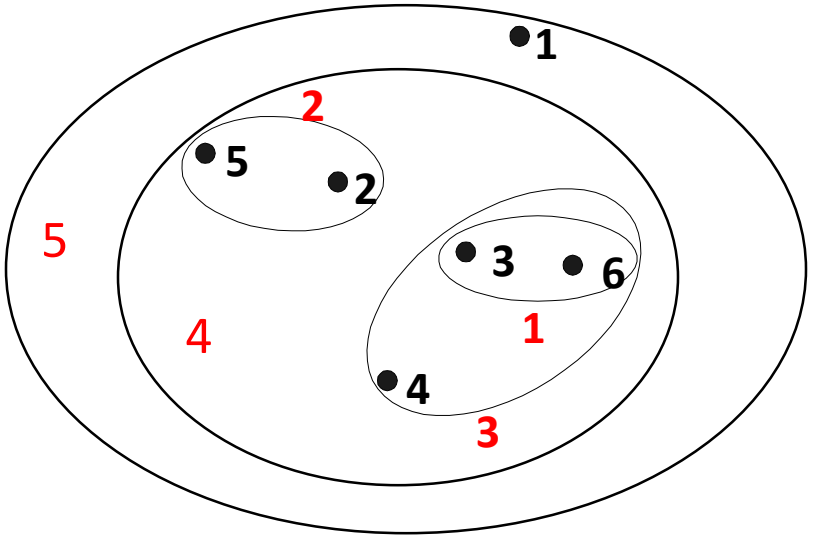
Distance Between Centroids Dendrogram

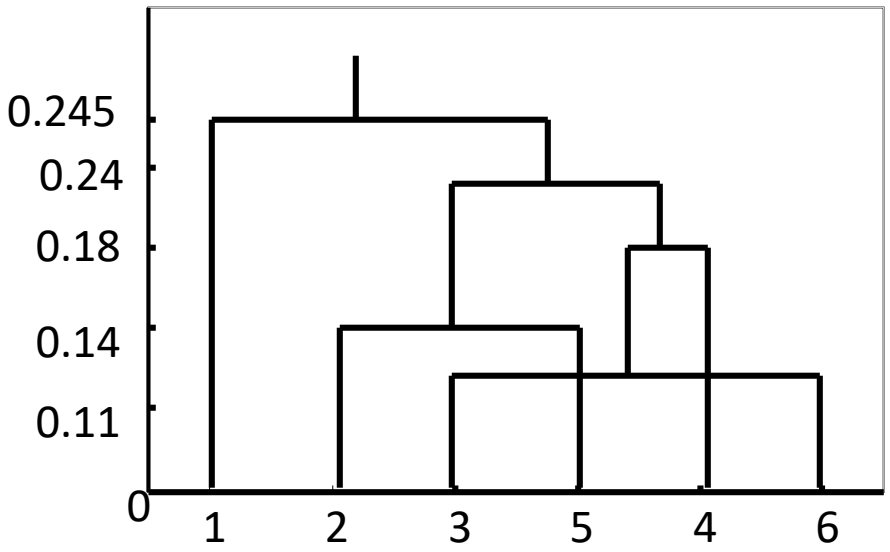
p1	0.4	0.53
c4	0.272	0.32

Centroid coordinates

	p1	c4
p1	0.0	0.245
c4	0.245	0.0

Distance matrix





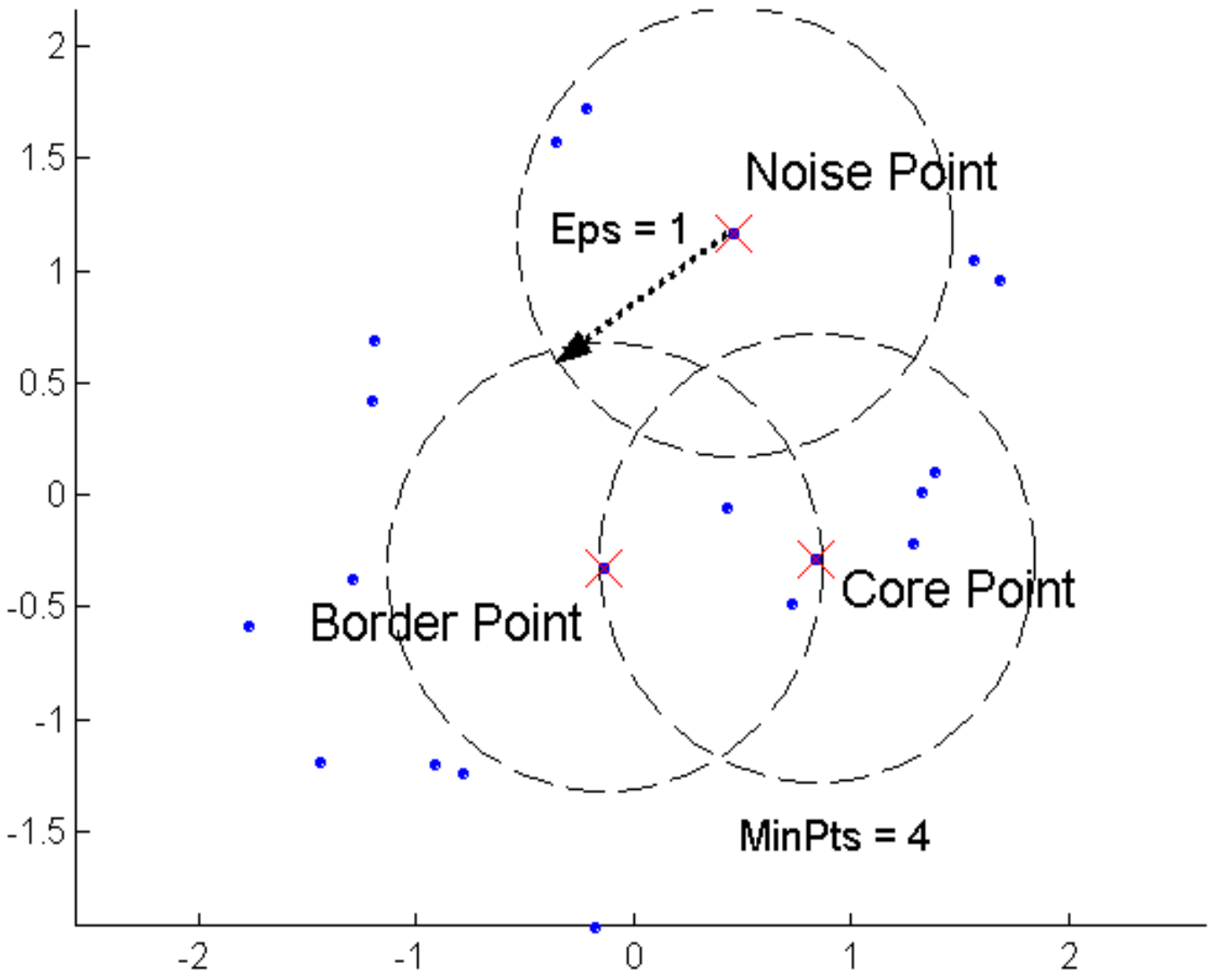
Lecture Overview

1. Recap
2. Max (Complete Graph)
3. Group Average
4. Centroid Distance
5. DBSCAN

DBSCAN

- DBSCAN is a density-based algorithm.
  - Density = number of points within a specified radius (eps)
  - A point is a core point if it has at least the specified number of points (MinPts) within radius ($\leq$ eps), which includes the point itself.
    - These are points that are at the interior of a cluster
  - A border point has fewer than MinPts within Eps, but is in the neighborhood of a core point
  - A noise point is any point that is not a core point or a border point.

DBSCAN: Core, Border, and Noise Points

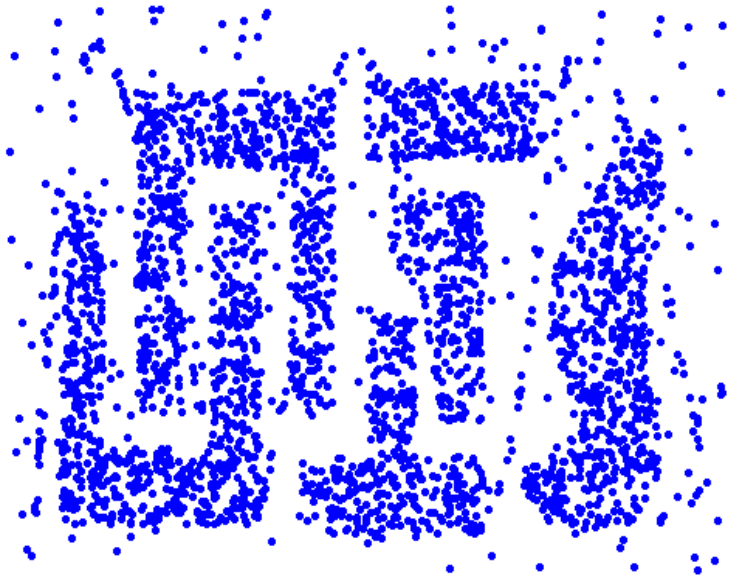


DBSCAN Algorithm

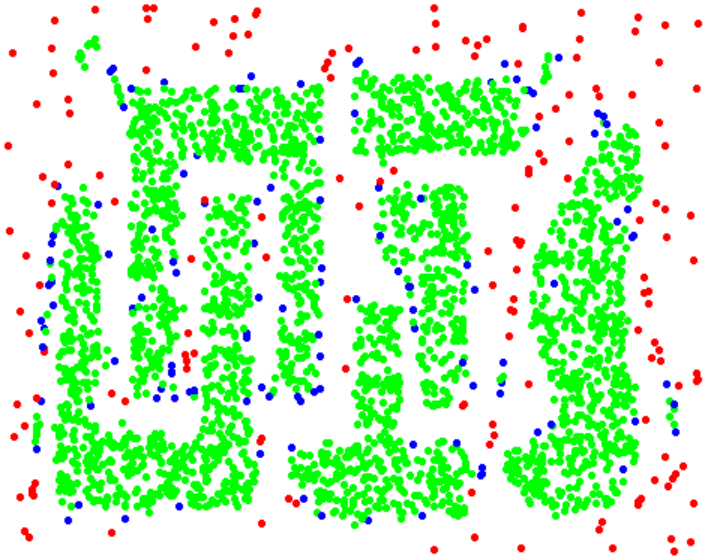
- Eliminate noise points
- Perform clustering on the remaining points

```
current_cluster_label ← 0
for all core points do
  if the core point has no cluster label then
    current_cluster_label ← current_cluster_label + 1
    Label the current core point with cluster label current_cluster_label
  end if
  for all points in the Eps-neighborhood, except the point itself do
    if the point does not have a cluster label then
      Label the point with cluster label of the current core point
    end if
  end for
end for
```

DBSCAN: Core, Border and Noise Points



Original Points

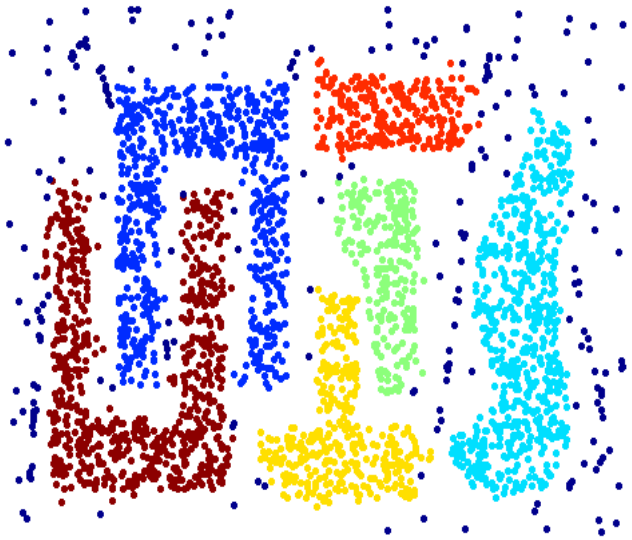


Point types: core, border and noise

Eps = 10, MinPts = 4

When DBSCAN Works Well

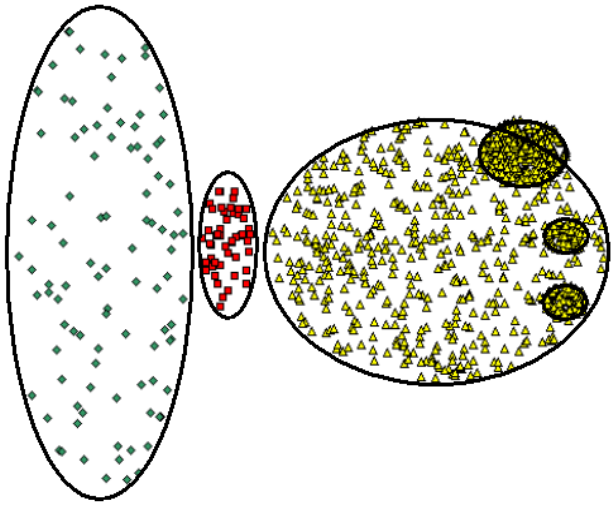
Original Points



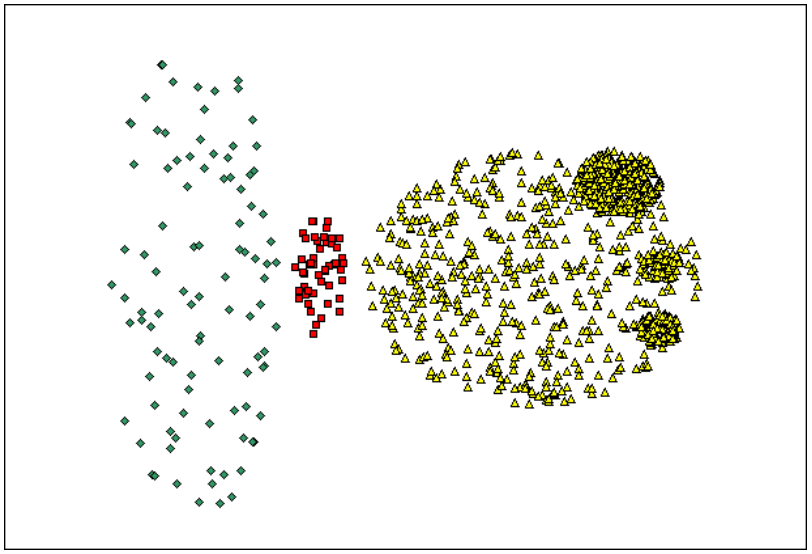
Clusters

- Resistant to Noise
- Can handle clusters of different shapes and sizes

When DBSCAN Does NOT Work Well

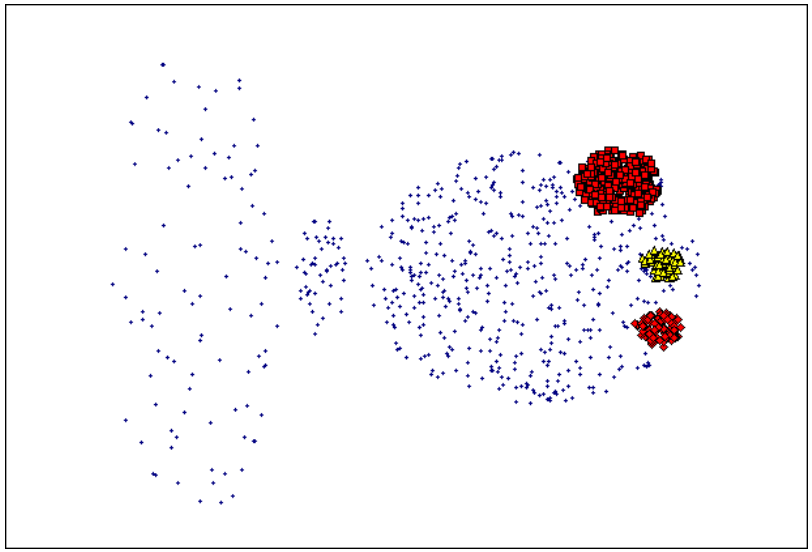


Original Points



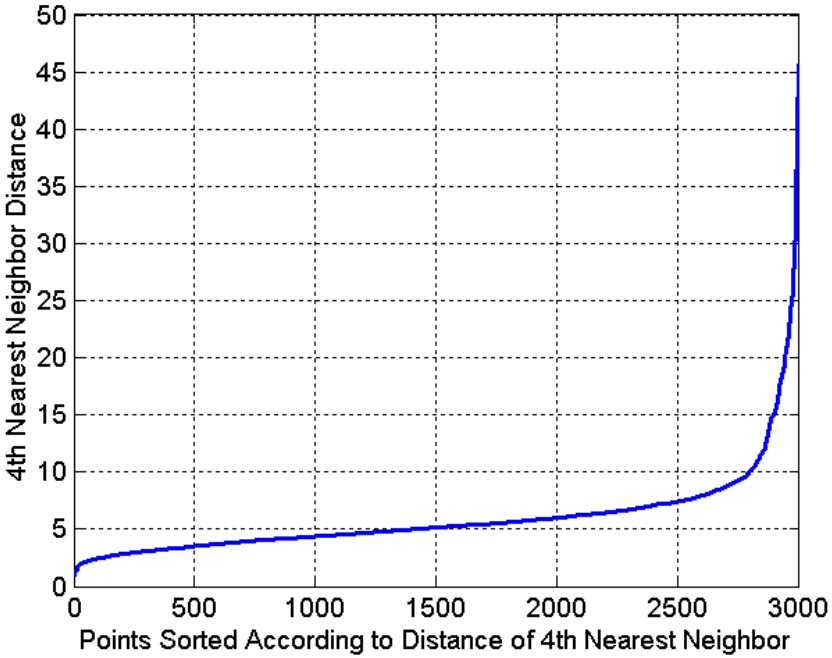
(MinPts=4, Eps=9.75).

- Varying densities
- High-dimensional data



(MinPts=4, Eps=9.92)

DBSCAN: Determining EPS and MinPts



- Idea is that for points in a cluster, their k^{th} nearest neighbors are at roughly the same distance
- Noise points have the k^{th} nearest neighbor at farther distance
- So, plot sorted distance of every point to its k^{th} nearest neighbor and find infliction point

DBSCAN in **R**

```
library(fpc);library(png)
img<-readPNG('example.png')
mimg<-as.matrix(img[,,2])
ind<-which(mimg!= 1, arr.ind=TRUE)
s<-sample(1:dim(ind)[1],2000,replace=FALSE)
sind<-ind[s,c(2,1)]
ds<-dbscan(sind,10.5,MinPts =
    7,method="hybrid",seeds=FALSE,showplot=1)
plot(ds, sind)
```

Reading

- TSKK sec 7.3.2,7.3.3, 7.3.4