

Data Warehouse and Data Mining

Dhruv Gupta

dhruv.gupta@ntnu.no

07-February-2022



NTNU

Norwegian University of
Science and Technology

1 Announcements and References

- Administrative
- References for Today's Lecture

2 Clustering

- K-Means Clustering
- Hierarchical Clustering
- Density based Clustering
- Evaluating Clustering and Clusters

1 Announcements and References

- Administrative
- References for Today's Lecture

2 Clustering

- K-Means Clustering
- Hierarchical Clustering
- Density based Clustering
- Evaluating Clustering and Clusters

Administrative

1 Second Assignment

- Due by 19.March.2021.

2 Classes on Tuesdays and Fridays for next two weeks

- 19.March.2021 (**Friday**) 0815 — 1000 HRS.
- 23.March.2021 (**Tuesday**) 0815 — 1000 HRS.
- 26.March.2021 (**Friday**) 0815 — 1000 HRS.

3 Volunteers for feedback regarding course

- Interested? Please contact me by email!

1 Announcements and References

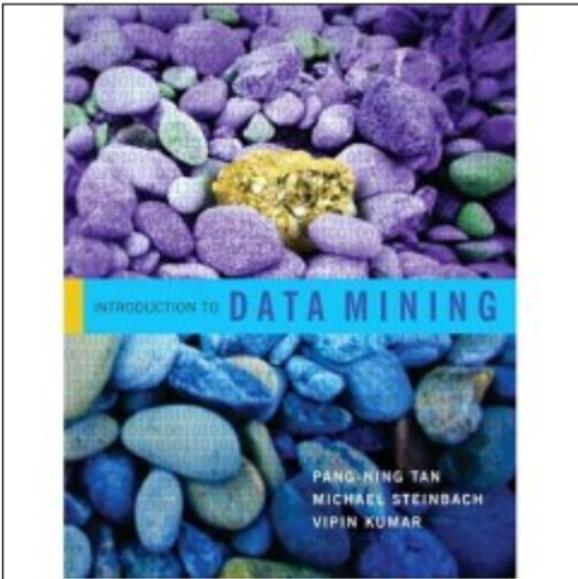
- Administrative
- References for Today's Lecture

2 Clustering

- K-Means Clustering
- Hierarchical Clustering
- Density based Clustering
- Evaluating Clustering and Clusters

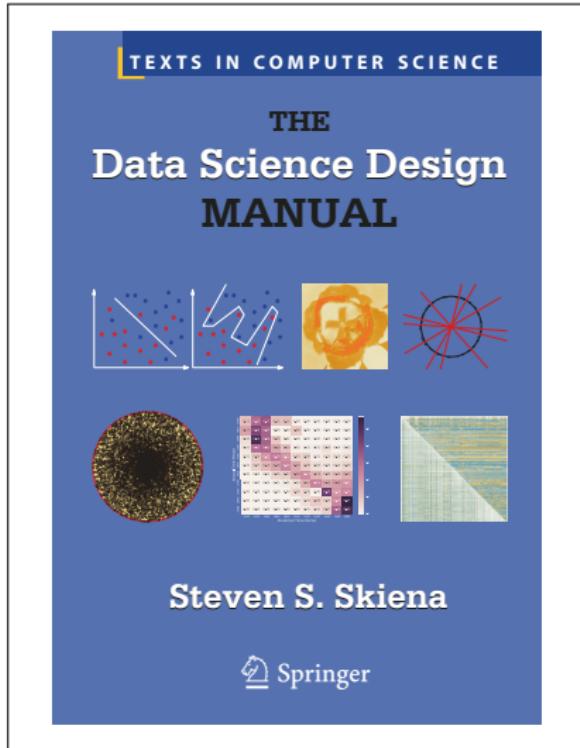
References for "Clustering"

- 1 Book: Tan et al. "*Introduction to Data Mining*", 1st Edition, 2006, Pearson Education Inc.
- 2 Text and images for majority of slides in "Clustering" are based on the book by Tan et al.



References for "Clustering"

- 1 Book: Steven S. Skiena, "*The Data Science Design Manual*", 2017, Springer.
- 2 All text and images for some slides in "Clustering" are based on the book by Steven S. Skiena.



1 Announcements and References

- Administrative
- References for Today's Lecture

2 Clustering

- K-Means Clustering
- Hierarchical Clustering
- Density based Clustering
- Evaluating Clustering and Clusters

FP-Growth Algorithm — Mining Frequent Itemsets

- Let, $minsup = 3$.
- $P = D$:
 $\{DB(4), DE(3), DA(3), DBE(3), DBA(3), DEA(3), DBEA(3)\}$
(after removing C as it is infrequent — support = 2).
- $P = C$:
 $\{CB(4), CE(3), CBE(3)\}$.
- $P = A$:
 $\{AE(4), AB(4), AEB(4)\}$.
- $P = E$: $\{EB(5)\}$.

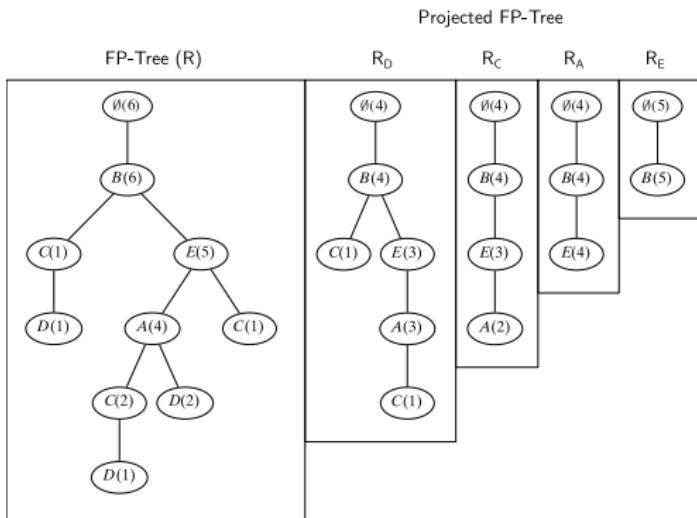


Figure 8.9. FPGrowth algorithm: frequent pattern tree projection.

1 Announcements and References

- Administrative
- References for Today's Lecture

2 Clustering

- K-Means Clustering
- Hierarchical Clustering
- Density based Clustering
- Evaluating Clustering and Clusters

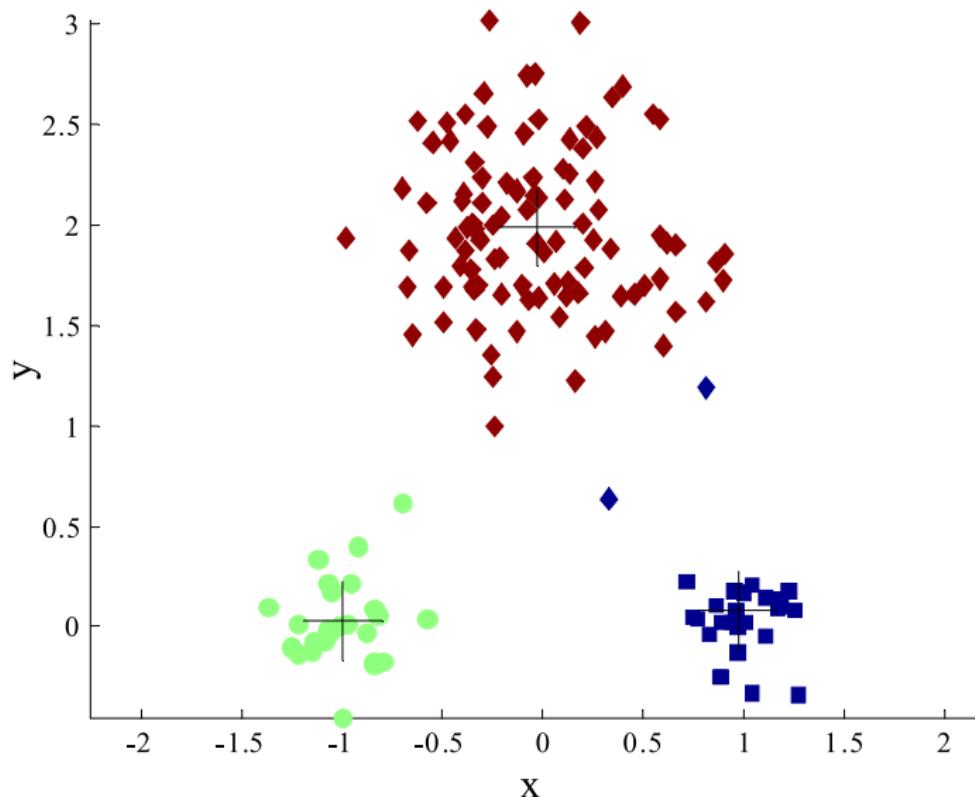
K-Means Clustering

- Partitional clustering approach.
- Number of clusters, K , must be specified.
- Each cluster is associated with a centroid (center point).
- Each point is assigned to the cluster with the closest centroid.
- The basic algorithm is very simple.

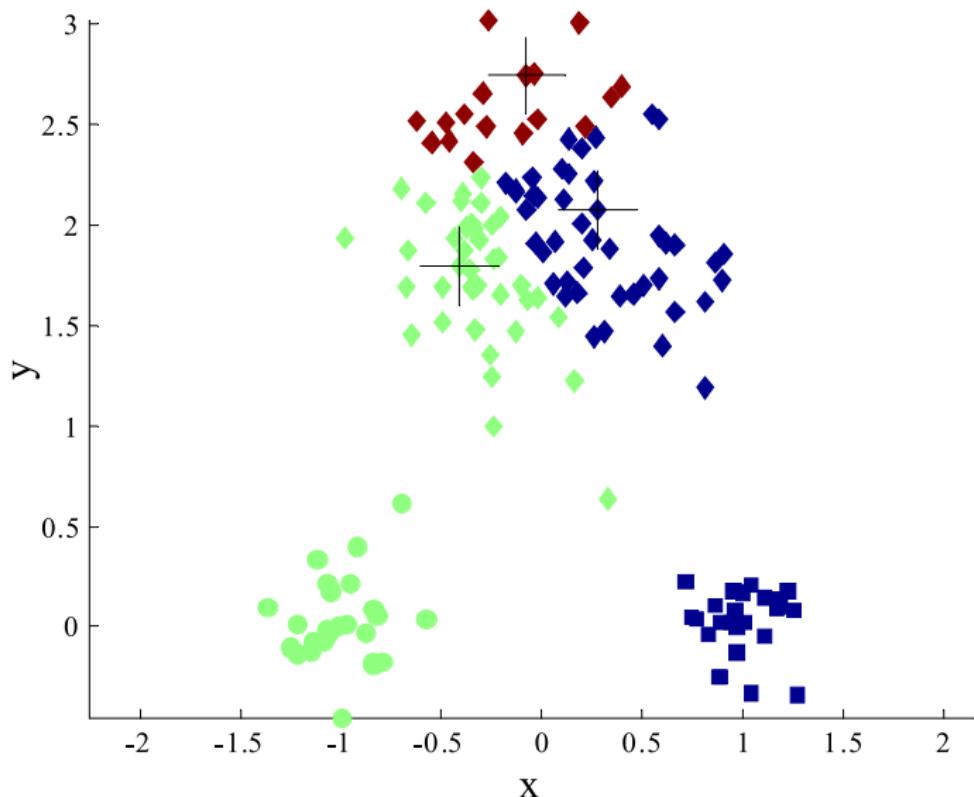
Algorithm 7.1 Basic K-means algorithm.

- 1: Select K points as initial centroids.
 - 2: **repeat**
 - 3: Form K clusters by assigning each point to its closest centroid.
 - 4: Recompute the centroid of each cluster.
 - 5: **until** Centroids do not change.
-

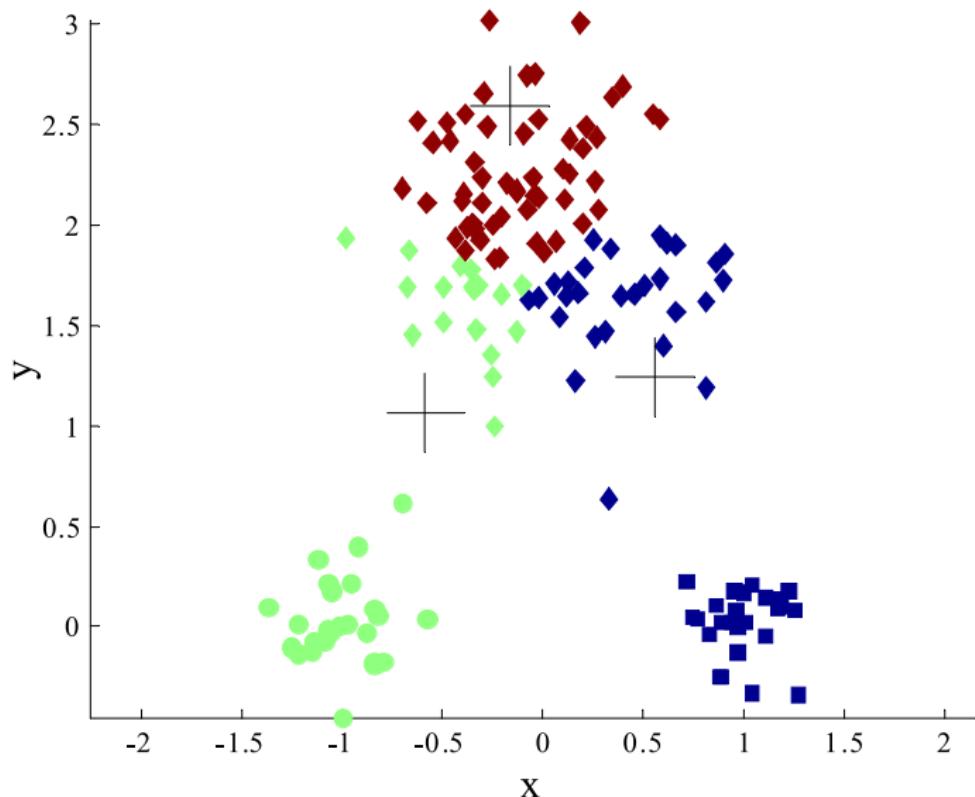
K-Means Clustering — Example



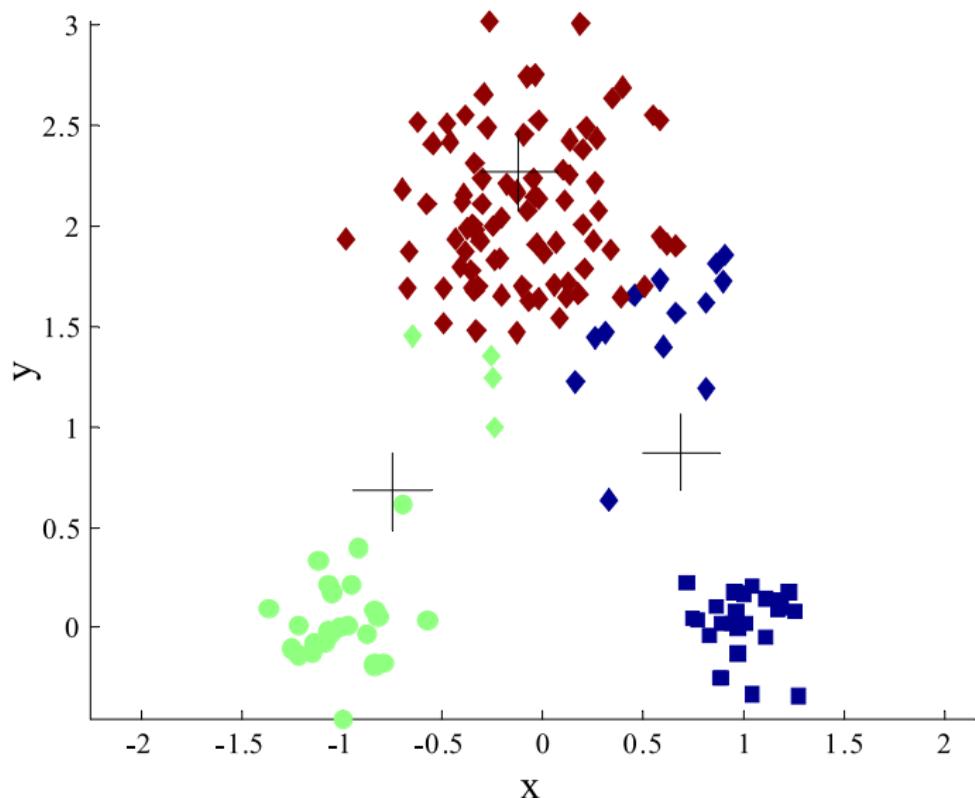
K-Means Clustering — Example Iterations 1



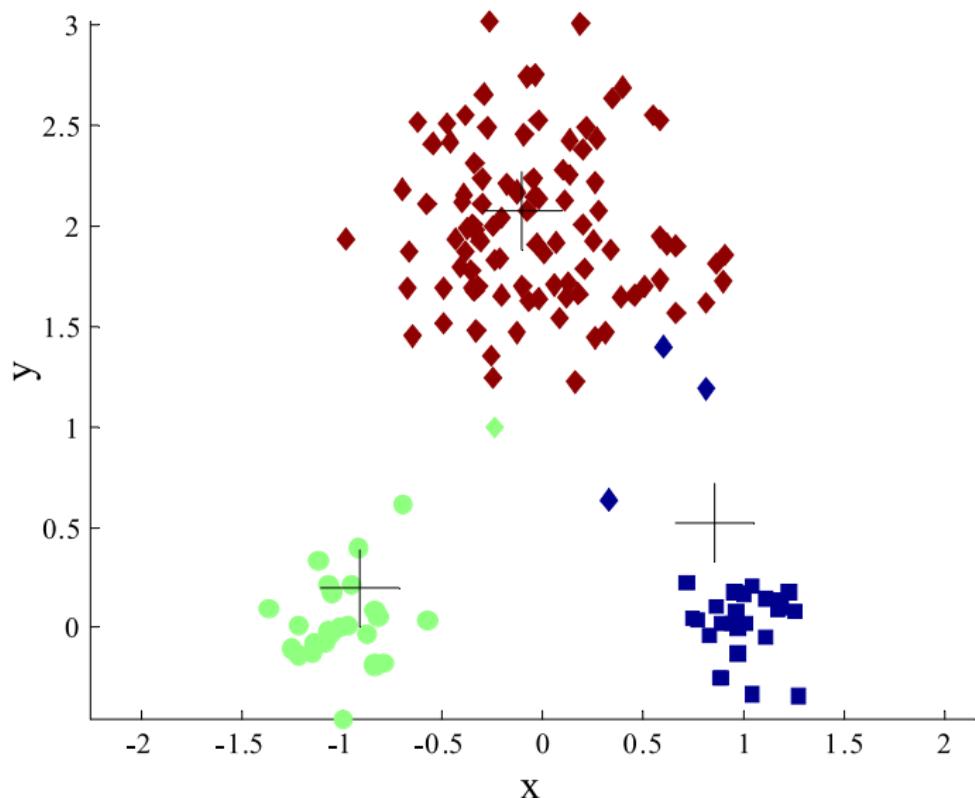
K-Means Clustering — Example Iterations 2



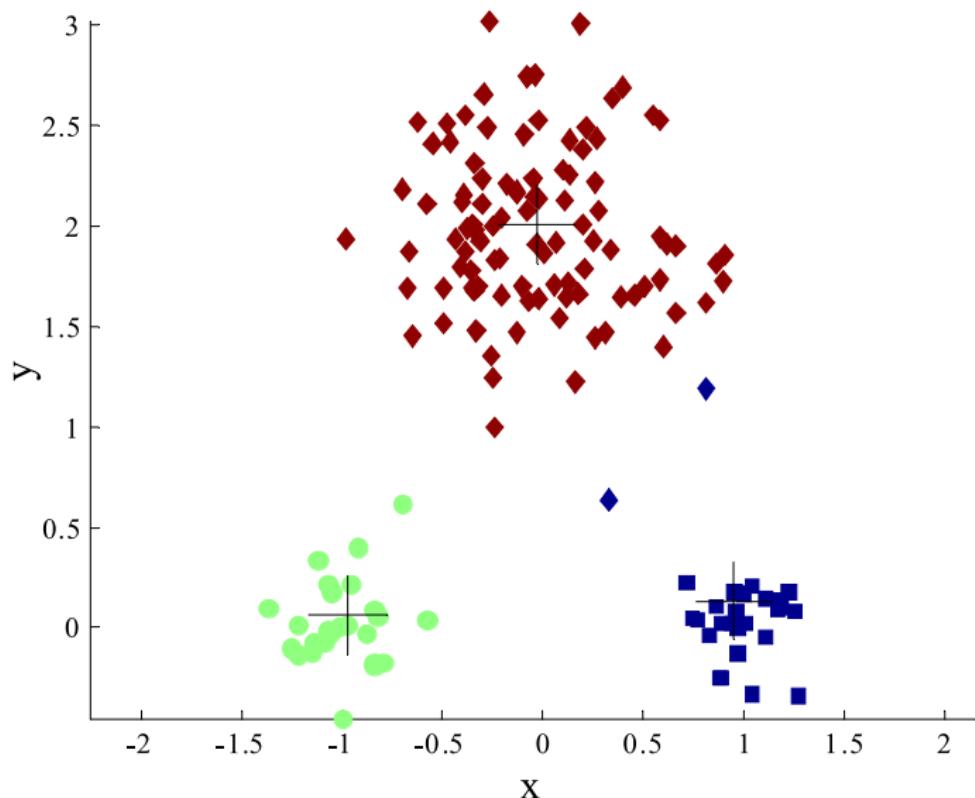
K-Means Clustering — Example Iterations 3



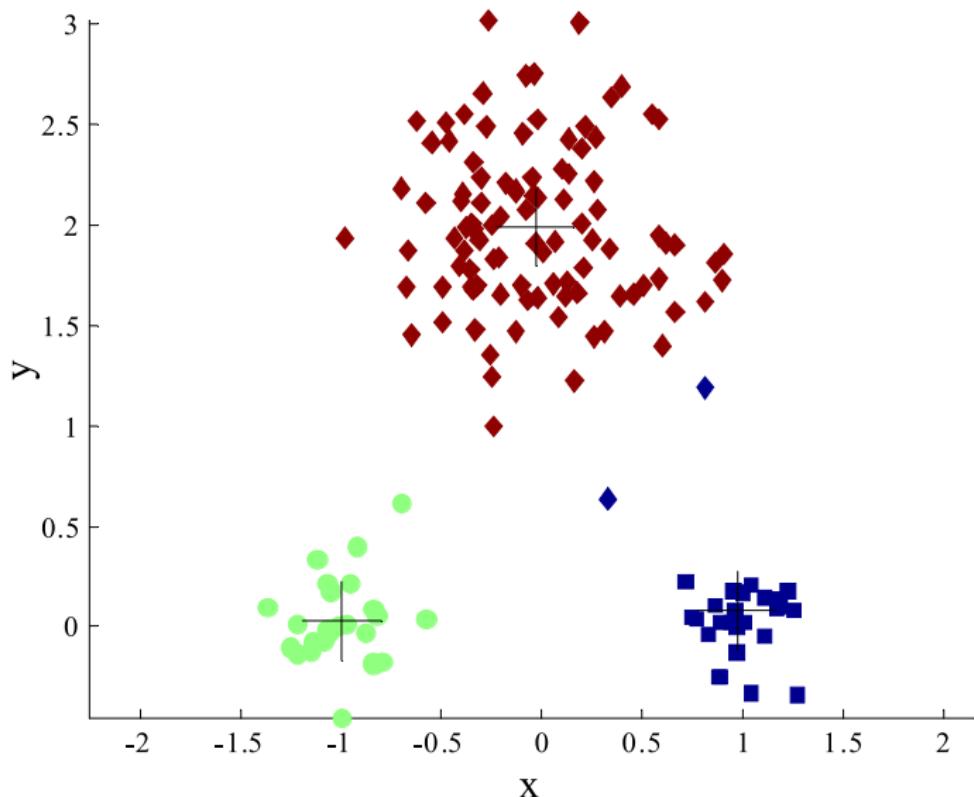
K-Means Clustering — Example Iterations 4



K-Means Clustering — Example Iterations 5



K-Means Clustering — Example Iterations 6



K-Means Clustering — Details

- Simple iterative algorithm.
- Initial centroids are often chosen randomly. Therefore, clusters produced can vary from one run to another.
- The centroid is (typically) the mean of the points in the cluster, but other definitions are possible (based on different similarity functions).
- K-means will converge for common proximity measures with appropriately defined centroid (based on different similarity functions).
- Most of the convergence happens in the first few iterations.
 - Often the stopping condition is changed to: until relatively few points change clusters.
- Time complexity is: $\mathcal{O}(n \cdot K \cdot l \cdot d)$.
- Space complexity is: $\mathcal{O}((n + K) \cdot d)$.
 - n = number of points.
 - K = number of clusters.
 - l = number of iterations.
 - d = number of attributes.

Evaluating K-Means Clusters

- Most common measure is **Sum of Squared Error (SSE)**.
 - For each point, the **error** is the distance to the nearest cluster center.
 - To get SSE, we square these errors and sum them.

$$\text{SSE} = \sum_{i=1}^K \sum_{x \in C_i} (\text{dist}(m_i, x))^2. \quad (1)$$

- x is a data point in cluster C_i and m_i is the representative point for cluster C_i .
- We can show that m_i corresponds to the center (mean) of the cluster:

$$c_i = \frac{1}{n_i} \sum_{x \in C_i} x. \quad (2)$$

- where, n_i is the number of objects in the i^{th} cluster.
- One easy way to **reduce SSE is to increase K**, the number of clusters.
 - A good clustering with smaller K can have a lower SSE than a poor clustering with higher K .

Parameters — How to Select K in K-Means

- We seek from error curve: value K where the rate of decline decreases.
- Error curve should look something like an arm in typing position: it slopes down rapidly from shoulder to elbow, and then slower from the elbow to the wrist.
- Want K to be located exactly at the elbow.
- Easier to identify when compared to a similar MSE error plot for random centers, since the relative rate of error reduction for random centers should be analogous to what we see past the elbow. The slow downward drift is telling us the extra clusters are not doing anything special for us.

Parameters — How to Select k in K-Means

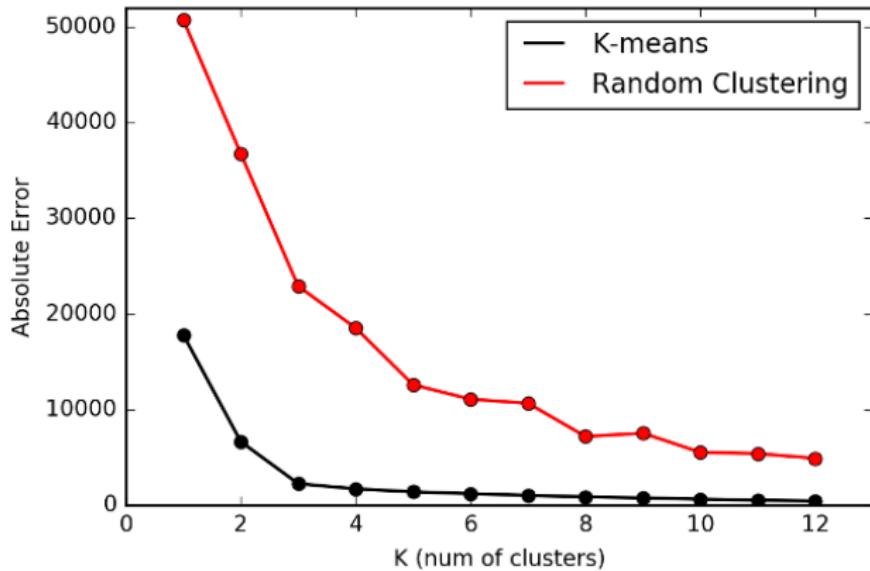
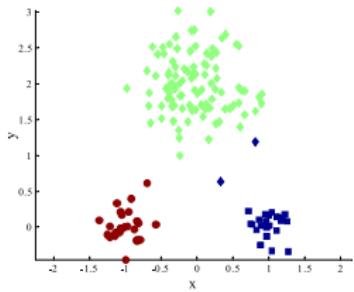
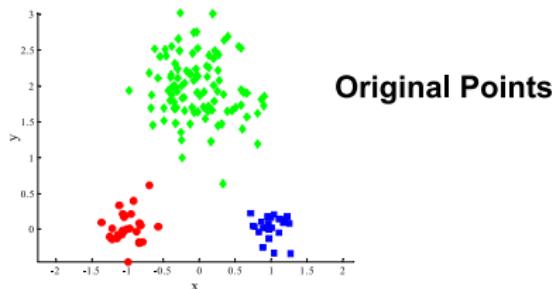
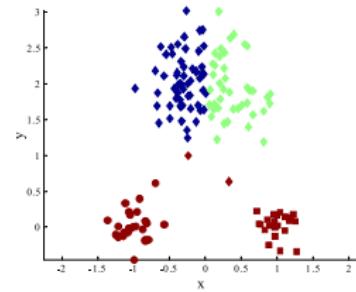


Figure 10.16: The error curve for k -means clustering on the point set of Figure 10.12, showing a bend in the elbow reflecting the three major clusters in the data. The error curve for random cluster centers is shown for comparison.

Parameters: Selecting Initial Centroids

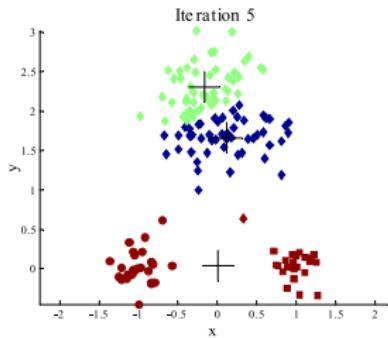
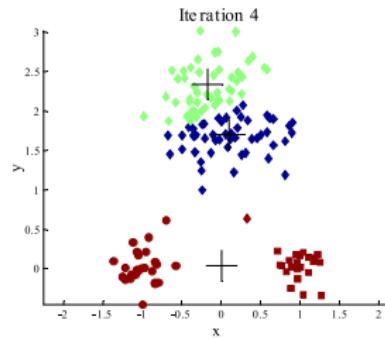
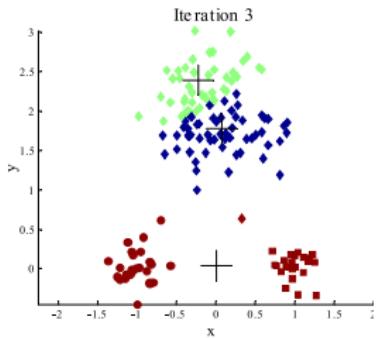
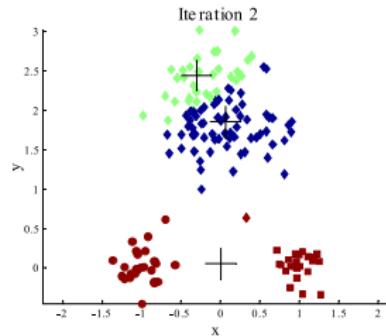
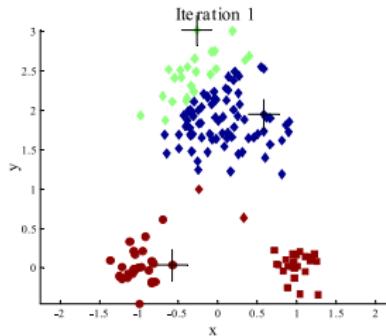


Optimal Clustering



Sub-optimal Clustering

Two Different K-Means Clusterings



Solutions to Initial Centroids Problem

- 1 Multiple runs: helps, but probability is not on your side.
- 2 Sample and use hierarchical clustering to determine initial centroids.
- 3 Select more than K initial centroids and then select among these initial centroids. Select most widely separated.
- 4 K-Means++.
- 5 Bisection K-means.
 - Not as susceptible to initialization issues.
- 6 Post-processing.

K-Means++: An Approach to Selecting Initial Centroids

- This approach can be **slower than random initialization**, but very consistently produces better results in terms of SSE.
- **Algorithm to select initial centroids C :**
 - 1 Select an initial point at random to be the first centroid.
 - 2 For $k - 1$ steps:
 - 1 For each of the N points x_i , $1 \leq i \leq N$, find the minimum squared distance to the currently selected centroids, C_1, \dots, C_j , $1 \leq j < k$, i.e., $\min_j(d(C_j, x_i))^2$.
 - 2 Randomly select a new centroid by choosing a point with probability proportional to $\frac{\min_j(d(C_j, x_i))^2}{(\sum_i d(C_j, x_i))^2}$.
 - 3 End For.
- Afterwards, **continue K-Means as usual**.

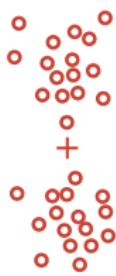
Bisecting K-Means

- Bisecting K-means algorithm: variant of K-means that can **produce a partitional or a hierarchical clustering**.
- Straightforward extension of the basic K-means algorithm:
 - 1 To obtain K clusters, **split the set of all points into two clusters**.
 - 2 **Select one of these clusters to split, and so on,** until K clusters have been produced.

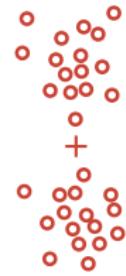
Algorithm 7.3 Bisecting K-means algorithm.

- 1: Initialize the list of clusters to contain the cluster consisting of all points.
 - 2: **repeat**
 - 3: Remove a cluster from the list of clusters.
 - 4: {Perform several “trial” bisections of the chosen cluster.}
 - 5: **for** $i = 1$ to *number of trials* **do**
 - 6: Bisect the selected cluster using basic K-means.
 - 7: **end for**
 - 8: Select the two clusters from the bisection with the lowest total SSE.
 - 9: Add these two clusters to the list of clusters.
 - 10: **until** The list of clusters contains K clusters.
-

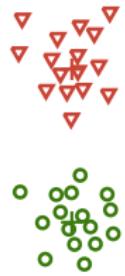
Bisecting K-Means



(a) Iteration 1.



(b) Iteration 2.



(c) Iteration 3.

Figure 7.8. Bisecting K-means on the four clusters example.

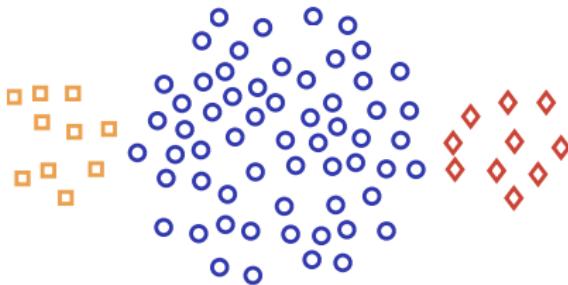
Pre-processing and Post-processing

- Pre-processing:
 - Normalize the data.
 - Eliminate outliers.
- Post-processing:
 - Eliminate small clusters that may represent outliers.
 - Split 'loose' clusters, i.e., clusters with relatively high SSE.
 - Merge clusters that are 'close' and that have relatively low SSE.
 - Can use these steps during the clustering process.

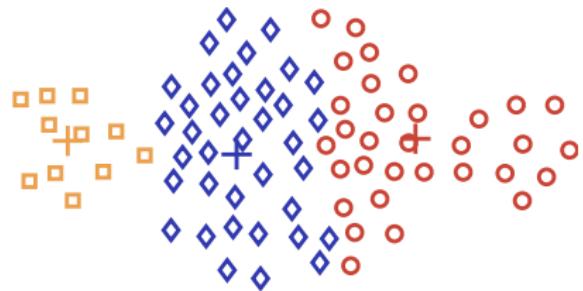
Limitations of K-Means

- K-means has problems when clusters are of differing:
 - Sizes
 - Densities
 - Non-globular shapes
- K-means has problems when the data contains outliers.

Limitations of K-Means: Differing Sizes



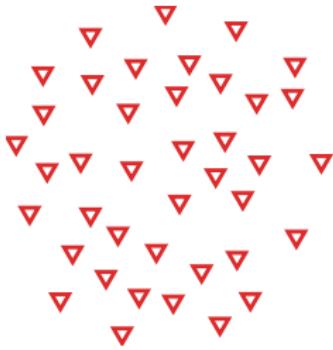
(a) Original points.



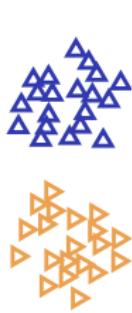
(b) Three K-means clusters.

Figure 7.9. K-means with clusters of different size.

Limitations of K-Means: Differing Densities



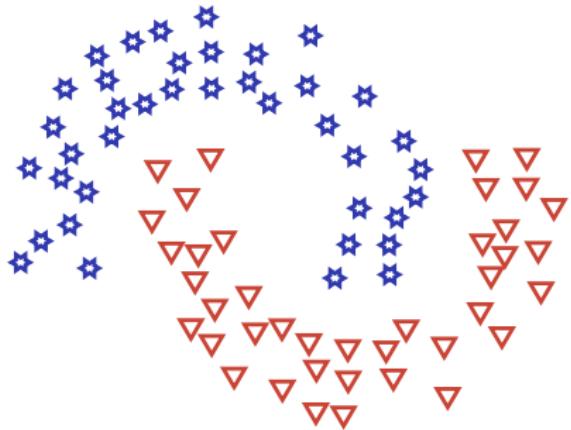
(a) Original points.



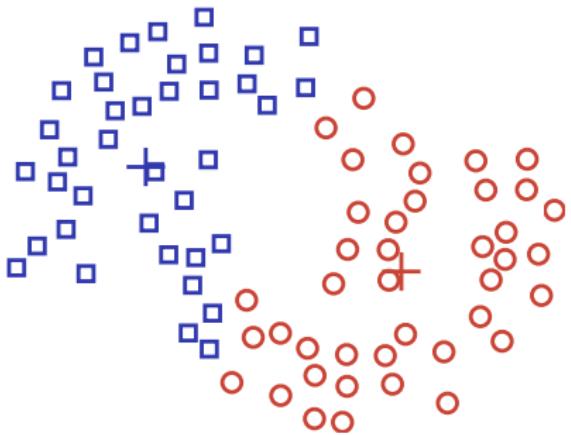
(b) Three K-means clusters.

Figure 7.10. K-means with clusters of different density.

Limitations of K-Means: Non-Globular Clusters



(a) Original points.



(b) Two K-means clusters.

Figure 7.11. K-means with non-globular clusters.

1 Announcements and References

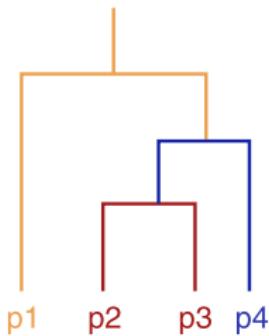
- Administrative
- References for Today's Lecture

2 Clustering

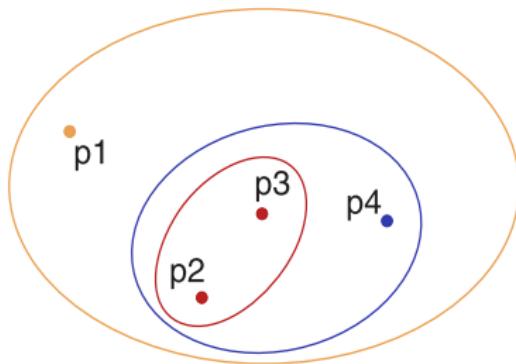
- K-Means Clustering
- Hierarchical Clustering**
- Density based Clustering
- Evaluating Clustering and Clusters

Hierarchical Clustering

- Produces a set of nested clusters organized as a hierarchical tree.
- Can be visualized as a **dendrogram**: a tree like diagram that records the sequences of merges or splits.



(a) Dendrogram.



(b) Nested cluster diagram.

Figure 7.13. A hierarchical clustering of four points shown as a dendrogram and as nested clusters.

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 **dendrogram** at the proper level.
- They may correspond to **meaningful taxonomies**.
- For 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).

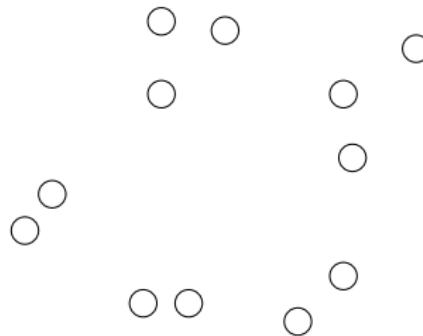
- Traditional hierarchical algorithms use a **similarity or distance matrix**.
- Merge or split one cluster at a time.

Agglomerative Clustering Algorithm

- Most popular hierarchical clustering technique.
- Basic algorithm is straightforward:
 - 1 Compute the proximity matrix.
 - 2 Let each data point be a cluster.
 - 3 Repeat
 - 1 Merge the two closest clusters.
 - 2 Update the proximity matrix.
 - 4 Until only a single cluster remains.
- Key operation is the computation of the proximity of two clusters.
- Different approaches to defining the distance between clusters distinguish the different algorithms.

Starting Situation

- Start with clusters of individual points and a proximity matrix.



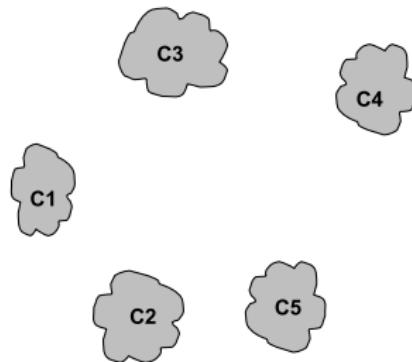
	p1	p2	p3	p4	p5	...
p1						
p2						
p3						
p4						
p5						
.						

Proximity Matrix



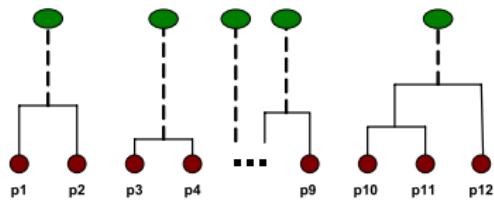
Intermediate Situation

- After some **merging steps**, we have some clusters.



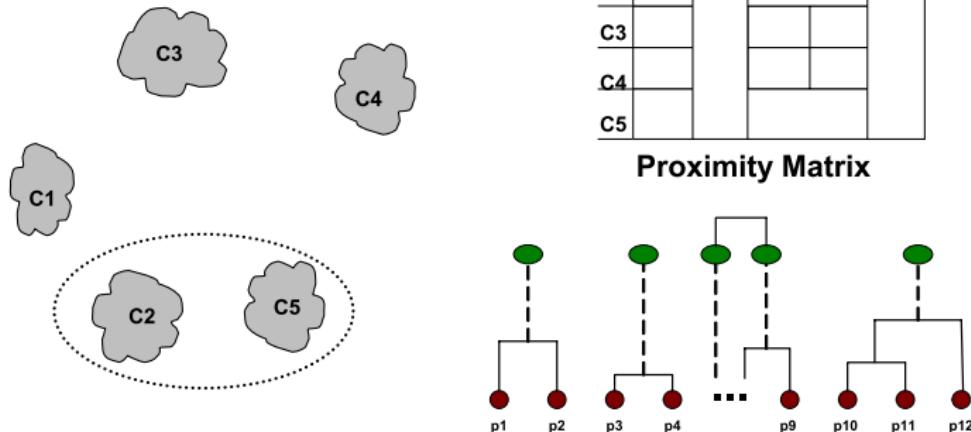
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Proximity Matrix



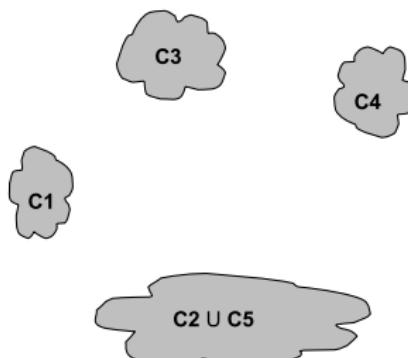
Intermediate Situation

- We want to merge the two closest clusters (C_2 and C_5) and update the proximity matrix.



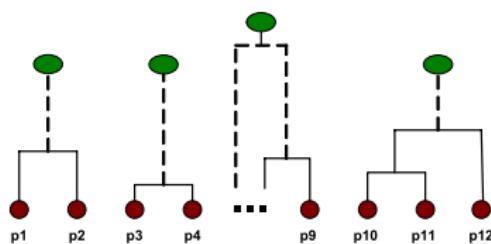
After Merging

- The question is: "How do we update the proximity matrix?".

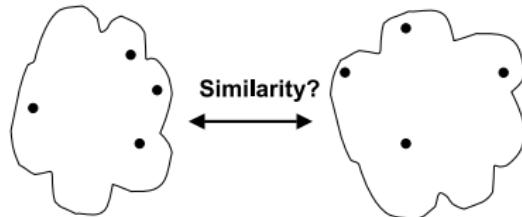


		C2 U			
		C1	C5	C3	C4
C1	C1	?			
	C2 U C5	?	?	?	?
C3		?			
C4		?			

Proximity Matrix



How to Define Inter-Cluster Distance

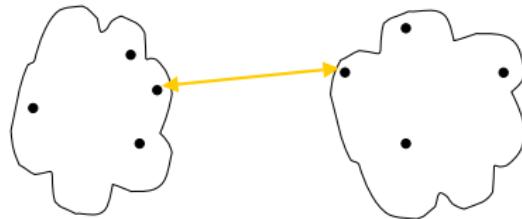


- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
 - Ward's Method uses squared error

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

Proximity Matrix

How to Define Inter-Cluster Distance

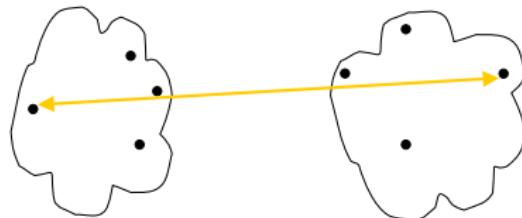


- **MIN**
- **MAX**
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
 - Ward's Method uses squared error

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

Proximity Matrix

How to Define Inter-Cluster Distance

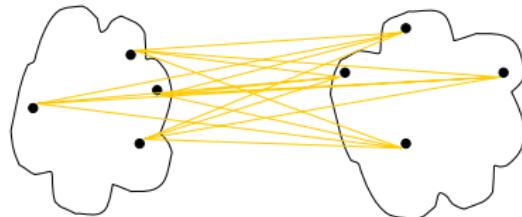


- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
 - Ward's Method uses squared error

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

Proximity Matrix

How to Define Inter-Cluster Distance

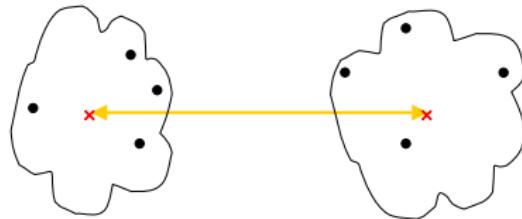


	p1	p2	p3	p4	p5	...
p1						

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

Proximity Matrix

How to Define Inter-Cluster Distance



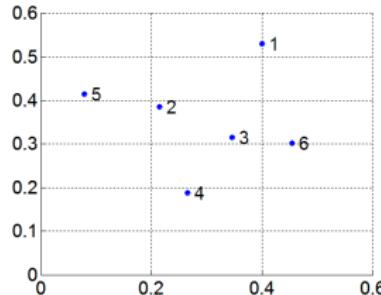
- MIN
- MAX
- Group Average
- **Distance Between Centroids**
- Other methods driven by an objective function
 - Ward's Method uses squared error

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

Proximity Matrix

Hierarchical Clustering: MIN or Single Link

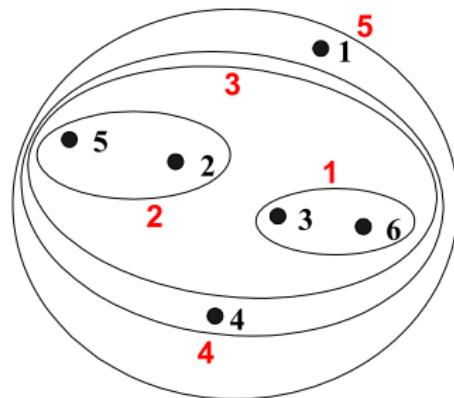
- Proximity of two clusters is based on the two closest points in the different clusters.
- Determined by one pair of points, i.e., by one link in the proximity graph.
- Example:



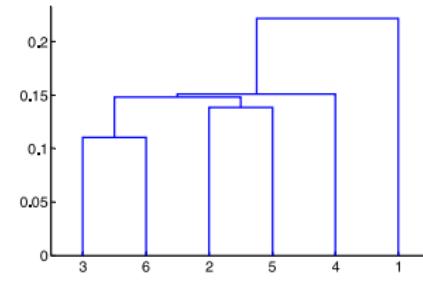
Distance Matrix:

	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

Hierarchical Clustering: MIN



Nested Clusters

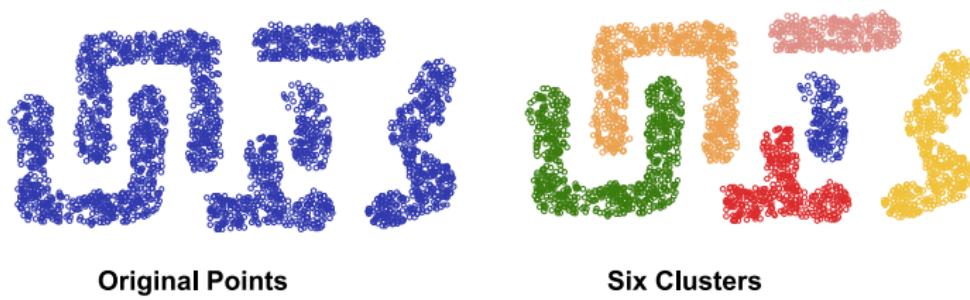


Dendrogram

$$\begin{aligned} \text{dist}(\{3, 6\}, \{2, 5\}) &= \min(\text{dist}(3, 2), \text{dist}(6, 2), \text{dist}(3, 5), \text{dist}(6, 5)) \\ &= \min(0.15, 0.25, 0.28, 0.39) \\ &= 0.15. \end{aligned}$$

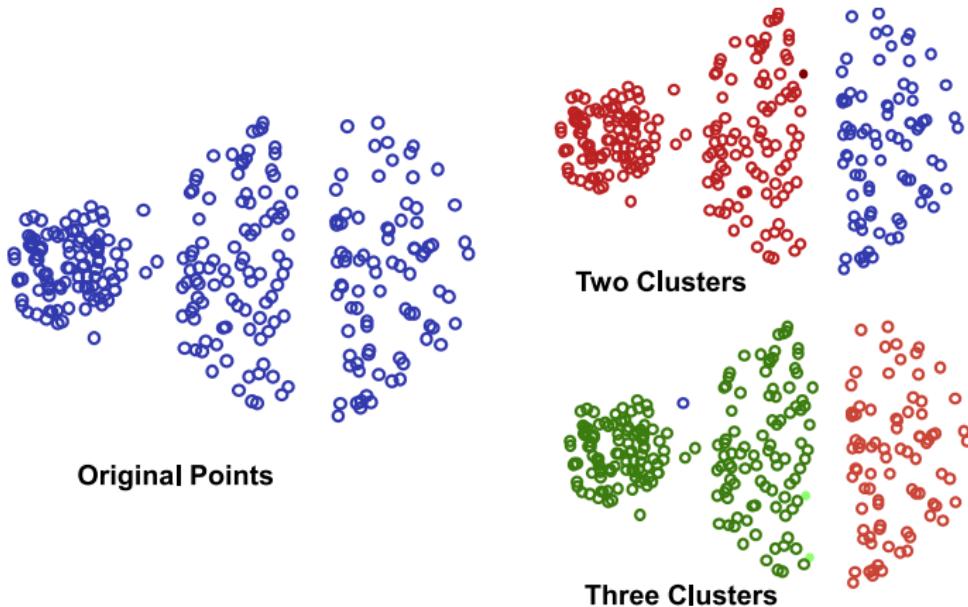
Hierarchical Clustering: Strengths of MIN

- Can handle non-elliptical shapes.



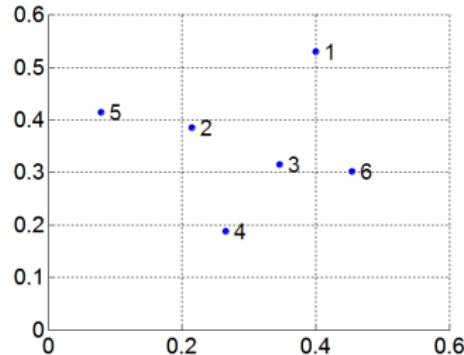
Hierarchical Clustering: Weakness of MIN

- Sensitive to noise and outliers.



Hierarchical Clustering: MAX or Complete Linkage

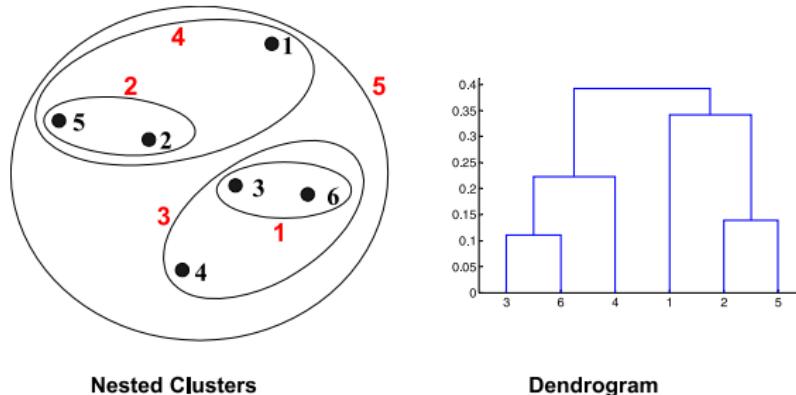
- Proximity of two cluster is based on the two most distant points in the different clusters.
 - Determined by all pairs of points in the two clusters.



Distance Matrix:

	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

Hierarchical Clustering: MAX or Complete Linkage



Nested Clusters

Dendrogram

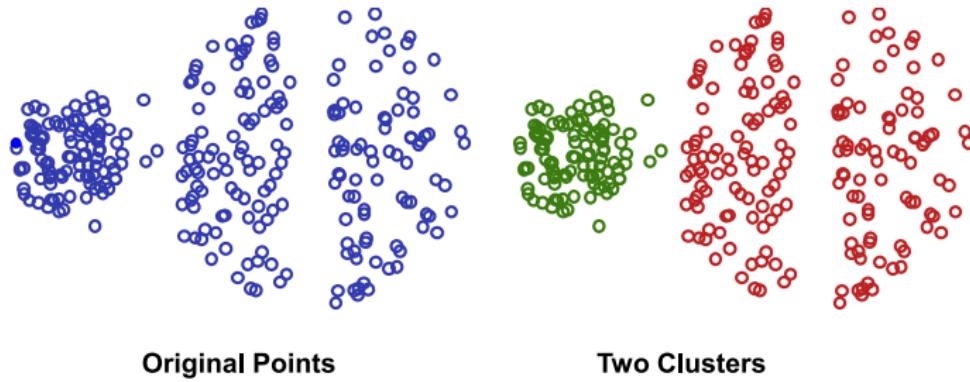
$$\begin{aligned} \text{dist}(\{3, 6\}, \{4\}) &= \max(\text{dist}(3, 4), \text{dist}(6, 4)) \\ &= \max(0.15, 0.22) \\ &= 0.22. \end{aligned}$$

$$\begin{aligned} \text{dist}(\{3, 6\}, \{2, 5\}) &= \max(\text{dist}(3, 2), \text{dist}(6, 2), \text{dist}(3, 5), \text{dist}(6, 5)) \\ &= \max(0.15, 0.25, 0.28, 0.39) \\ &= 0.39. \end{aligned}$$

$$\begin{aligned} \text{dist}(\{3, 6\}, \{1\}) &= \max(\text{dist}(3, 1), \text{dist}(6, 1)) \\ &= \max(0.22, 0.23) \\ &= 0.23. \end{aligned}$$

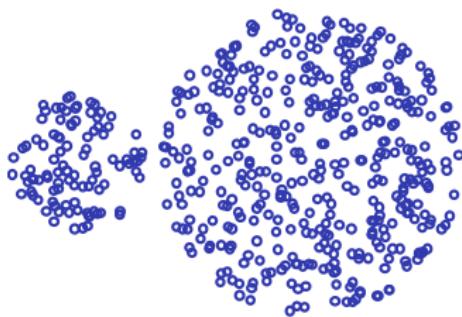
Hierarchical Clustering: Strengths of MAX

- Less susceptible to noise and outliers.

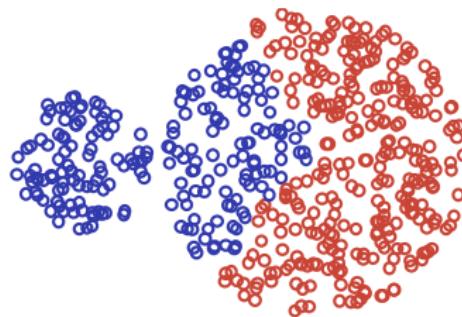


Hierarchical Clustering: Limitations of MAX

- Tends to break large clusters.
- Biased towards globular clusters.



Original Points



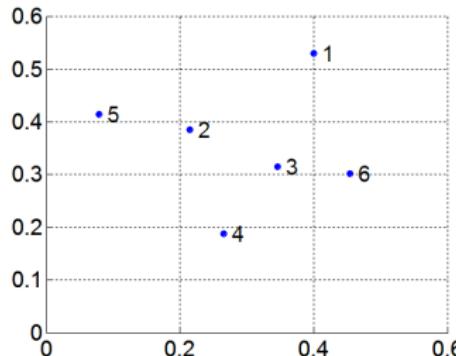
Two Clusters

Hierarchical Clustering: Group Average

- Proximity of two clusters is the average of pairwise proximity between points in the two clusters.

$$\text{proximity}(\text{cluster}_1, \text{cluster}_2) = \frac{\sum_{p_i \in \text{cluster}_i} \sum_{p_j \in \text{cluster}_j} \text{proximity}(p_i, p_j)}{|\text{cluster}_i| \cdot |\text{cluster}_j|}. \quad (3)$$

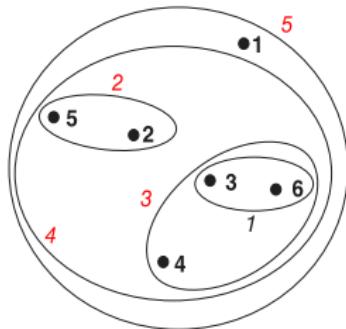
- Need to use average connectivity for scalability since total proximity favors large clusters.



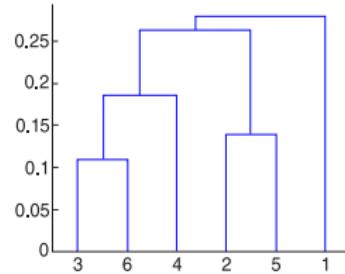
Distance Matrix:

	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

Hierarchical Clustering: Group Average



(a) Group average clustering.



(b) Group average dendrogram.

$$\begin{aligned} \text{dist}(\{3, 6, 4\}, \{1\}) &= (0.22 + 0.37 + 0.23)/(3 \times 1) \\ &= 0.28 \end{aligned}$$

$$\begin{aligned} \text{dist}(\{2, 5\}, \{1\}) &= (0.24 + 0.34)/(2 \times 1) \\ &= 0.29 \end{aligned}$$

$$\begin{aligned} \text{dist}(\{3, 6, 4\}, \{2, 5\}) &= (0.15 + 0.28 + 0.25 + 0.39 + 0.20 + 0.29)/(3 \times 2) \\ &= 0.26 \end{aligned}$$

Hierarchical Clustering: Group Average

- Compromise between single and complete link.
- **Strengths:** less susceptible to noise and outliers.
- **Limitations:** biased towards globular clusters.

Hierarchical Clustering: Time and Space Requirements

- $\mathcal{O}(N^2)$ space since it uses the **proximity matrix**.
 - N is the number of points.
- $\mathcal{O}(N^3)$ time in many cases.
 - There are **N steps** and at each step the size, N^2 , proximity matrix must be updated and searched.
 - Complexity can be reduced to $\mathcal{O}(N^2\log(N))$ time with some cleverness.

Hierarchical Clustering: Problems and Limitations

- Once a **decision** is made to combine two clusters, it **cannot be undone**.
- No **global objective function** is directly minimized.
- Different schemes have problems with one or more of the following:
 - Sensitivity to **noise and outliers**.
 - Difficulty handling **clusters of different sizes and non-globular shapes**.
 - Breaking large clusters**.

1 Announcements and References

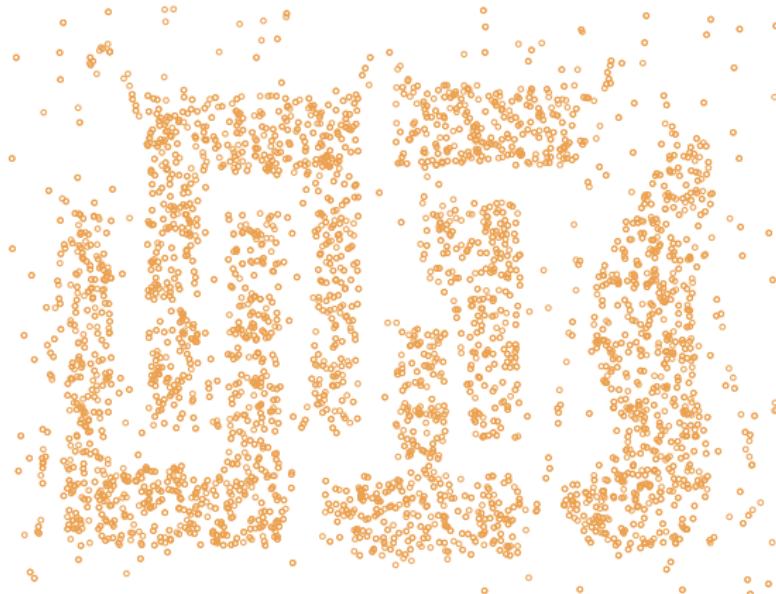
- Administrative
- References for Today's Lecture

2 Clustering

- K-Means Clustering
- Hierarchical Clustering
- **Density based Clustering**
- Evaluating Clustering and Clusters

Density based Clustering

- Clusters are regions of high density that are separated from one another by regions on low density.



Density based Clustering — DBSCAN Algorithm

- 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 a **specified number of points (MinPts) within Eps**.
 - These are points that are at the interior of a cluster.
 - Counts the point itself.
 - A **border point** is not a core point, 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

MinPts = 7

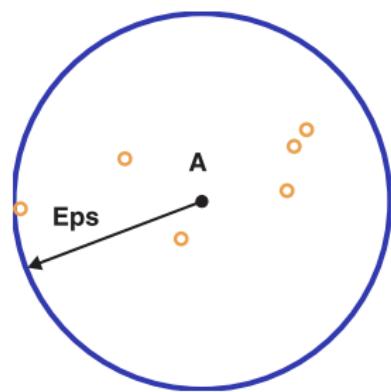


Figure 7.20. Center-based density.

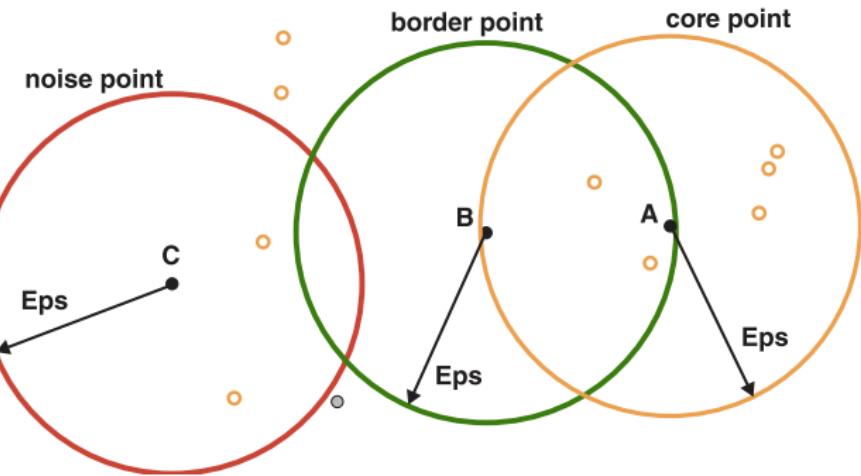
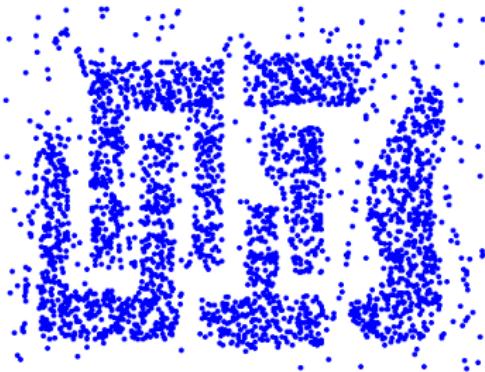
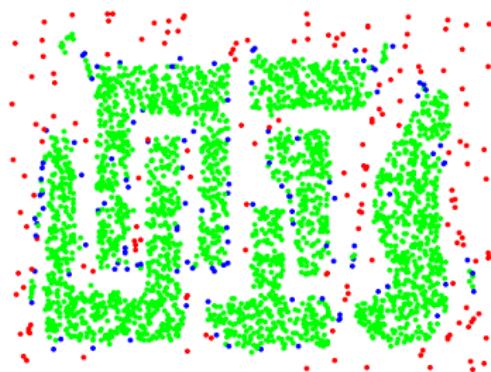


Figure 7.21. Core, border, and noise points.

DBSCAN — Core, Border, and Noise Points



Original Points

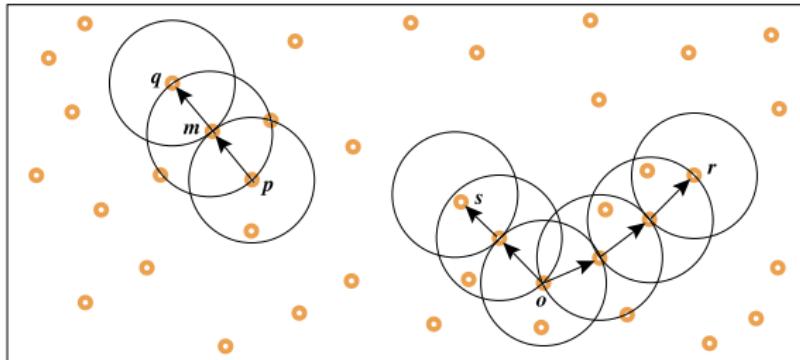


Point types: **core**,
border and **noise**

Eps = 10, MinPts = 4

DBSCAN Algorithm

- Form clusters using core points, and assign border points to one of its neighboring clusters.
- DBSCAN Algorithm:
 - Label all points as core, border, or noise points.
 - Eliminate noise points.
 - Put an edge between all core points within a distance Eps of each other.
 - Make each group of connected core points into a separate cluster.
 - Assign each border point to one of the clusters of its associated core points.

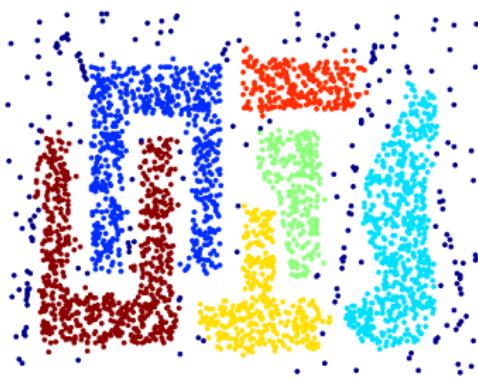


DBSCAN— Strengths

- Resistant to noise.
- Can handle clusters of different shapes and sizes.



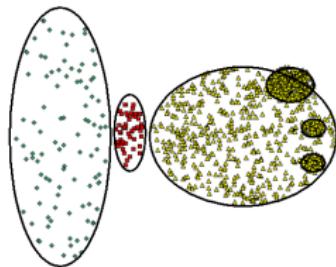
Original Points



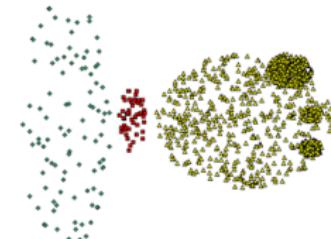
Clusters

DBSCAN — Weakness

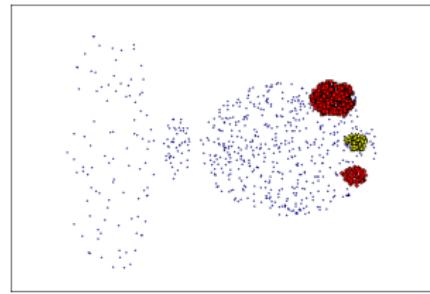
- Varying densities.
- High-dimensional data.



Original Points



(MinPts=4, Eps=9.92).



(MinPts=4, Eps=9.75)

DBSCAN — Determining Parameters (EPS and MinPts)

- Idea is that for points in a cluster, their k^{th} nearest neighbors are at close 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.

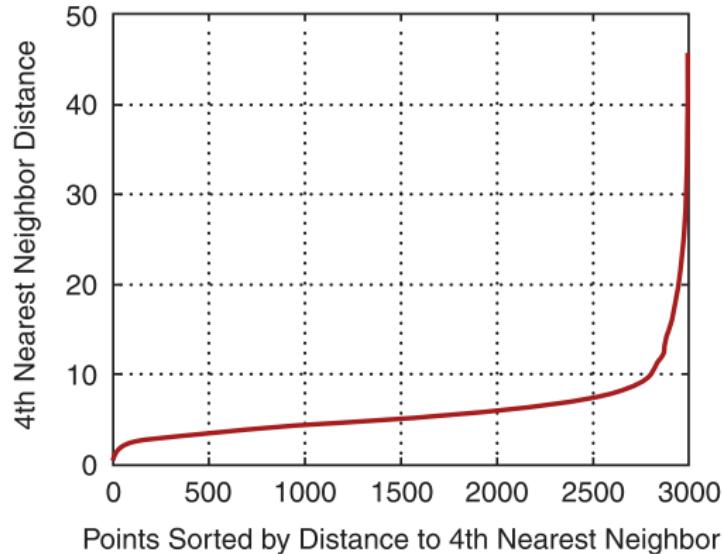


Figure 7.23. K-dist plot for sample data.

DBSCAN — Time and Space Complexity

- **Time complexity:** $\mathcal{O}(m \cdot \text{time to find points in the Eps-neighborhood})$, where m is the number of points.
- Worst case, this complexity is $\mathcal{O}(m^2)$.
- Using indexes time complexity can be as low as $\mathcal{O}(m \cdot \log(m))$.
- **Space requirement** is $\mathcal{O}(m)$ only a small amount of data for each point is needed (i.e., the cluster label and the identification of each point as a core, border, or noise point).

1 Announcements and References

- Administrative
- References for Today's Lecture

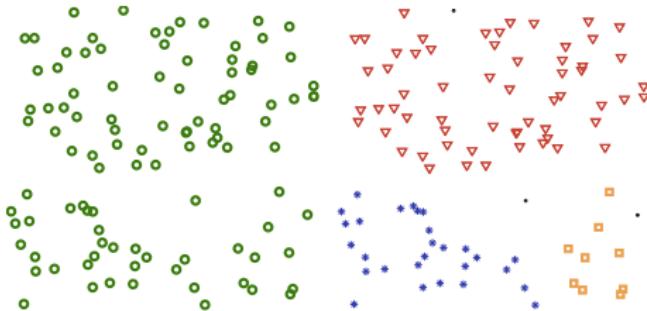
2 Clustering

- K-Means Clustering
- Hierarchical Clustering
- Density based Clustering
- Evaluating Clustering and Clusters

Evaluation — Cluster Validity

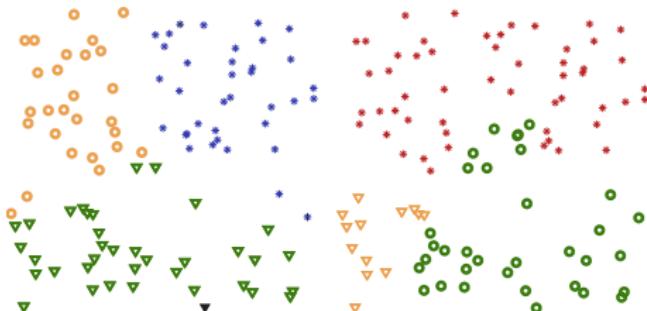
- For supervised classification we have a variety of measures to evaluate how good our model is: accuracy, precision, recall.
- For cluster analysis, the analogous question is how to **evaluate the “goodness” of the resulting clusters?**
- But “clusters are in the eye of the beholder”!
- Then why do we want to evaluate them?
 - To **avoid finding patterns in noise**.
 - To **compare clustering algorithms**.
 - To **compare two sets of clusters**.
 - To **compare two clusters**.

Cluster that can be found in Random Data



(a) Original points.

(b) Three clusters found by DBSCAN.



(c) Three clusters found by K-means.

(d) Three clusters found by complete link.

Figure 7.26. Clustering of 100 uniformly distributed points.

Different Aspects of Cluster Validation

- 1 Determining the clustering tendency of a set of data, i.e., distinguishing whether non-random structure actually exists in the data.
- 2 Comparing the results of a cluster analysis to externally known results, e.g., to externally given class labels.
- 3 Evaluating how well the results of a cluster analysis fit the data without reference to external information.
 - Use only the data.
- 4 Comparing the results of two different sets of cluster analyses to determine which is better.
- 5 Determining the ‘correct’ number of clusters.
 - For 2, 3, and 4, we can further distinguish whether we want to evaluate the entire clustering or just individual clusters.

Measures of Cluster Validity

- Numerical measures to determine cluster validity, are of **three types**.
- **Internal Index (Unsupervised)**: Used to measure the goodness of a clustering structure without respect to external information.
 - Sum of Squared Error (SSE)
- **External Index (Supervised)**: Used to measure the extent to which cluster labels match externally supplied class labels.
 - Entropy
- **Relative Index**: Used to compare two different clusterings or clusters.
 - Often an external or internal index is used for this function, e.g., SSE or entropy.
- Sometimes these are referred to as criteria instead of indices.
 - However, sometimes criterion is the general strategy and index is the numerical measure that implements the criterion.

Unsupervised Cluster Evaluation: Cohesion and Separation

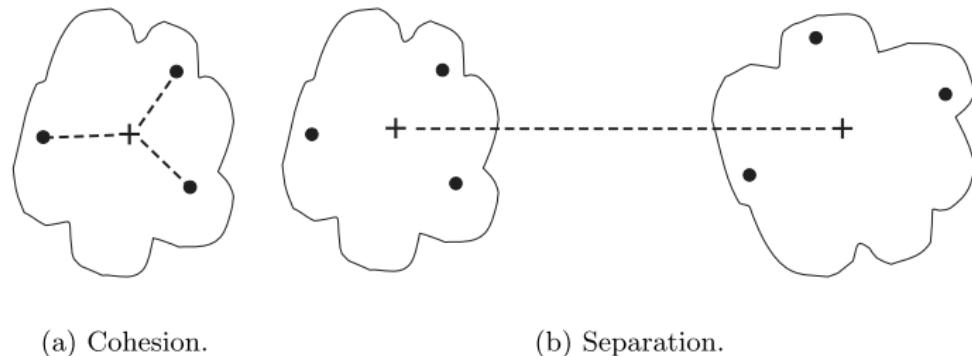


Figure 7.28. Prototype-based view of cluster cohesion and separation.

- **Cluster Cohesion:** Measures how closely related are objects in a cluster.
 - Example: SSE.
- **Cluster Separation:** Measure how distinct or well-separated a cluster is from other clusters.

Unsupervised Cluster Evaluation: Cohesion and Separation

- Example: Squared Error

- Cohesion is measured by the within cluster sum of squares (SSE):

$$\text{SSE} = \sum_i \sum_{x \in C_i} (x - m_i)^2. \quad (4)$$

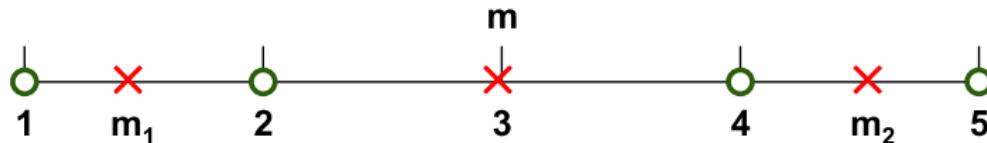
- Separation is measured by the between cluster sum of squares.

$$\text{SSB} = \sum_i |C_i|(m - m_i)^2, \quad (5)$$

- where, $|C_i|$ is the size of cluster i .

Unsupervised Cluster Evaluation: Cohesion and Separation

- Example: SSE
 - SSB + SSE = constant.



K=1 cluster: $SSE = (1 - 3)^2 + (2 - 3)^2 + (4 - 3)^2 + (5 - 3)^2 = 10$

$$SSB = 4 \times (3 - 3)^2 = 0$$

$$Total = 10 + 0 = 10$$

K=2 clusters: $SSE = (1 - 1.5)^2 + (2 - 1.5)^2 + (4 - 4.5)^2 + (5 - 4.5)^2 = 1$

$$SSB = 2 \times (3 - 1.5)^2 + 2 \times (4.5 - 3)^2 = 9$$

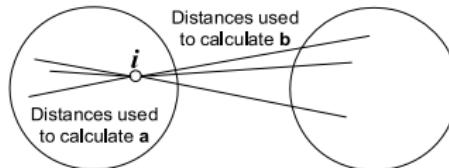
$$Total = 1 + 9 = 10$$

Unsupervised Cluster Evaluation: Silhouette Coefficient

- **Silhouette Coefficient:** combines ideas of both cohesion and separation, but for individual points, as well as clusters and clusterings.
- For an individual point, i
 - Calculate $a = \text{average distance of } i \text{ to the points in its cluster}$.
 - Calculate $b = \min(\text{average distance of } i \text{ to points in another cluster})$.
 - The silhouette coefficient for a point is then given by

$$s = \frac{(b - a)}{\max(a, b)} \quad (6)$$

- Value can vary between -1 and 1
- Typically ranges between 0 and 1.
- The closer to 1 the better.
- Can calculate the average silhouette coefficient for a cluster or a clustering.



Unsupervised Cluster Evaluation: Silhouette Coefficient

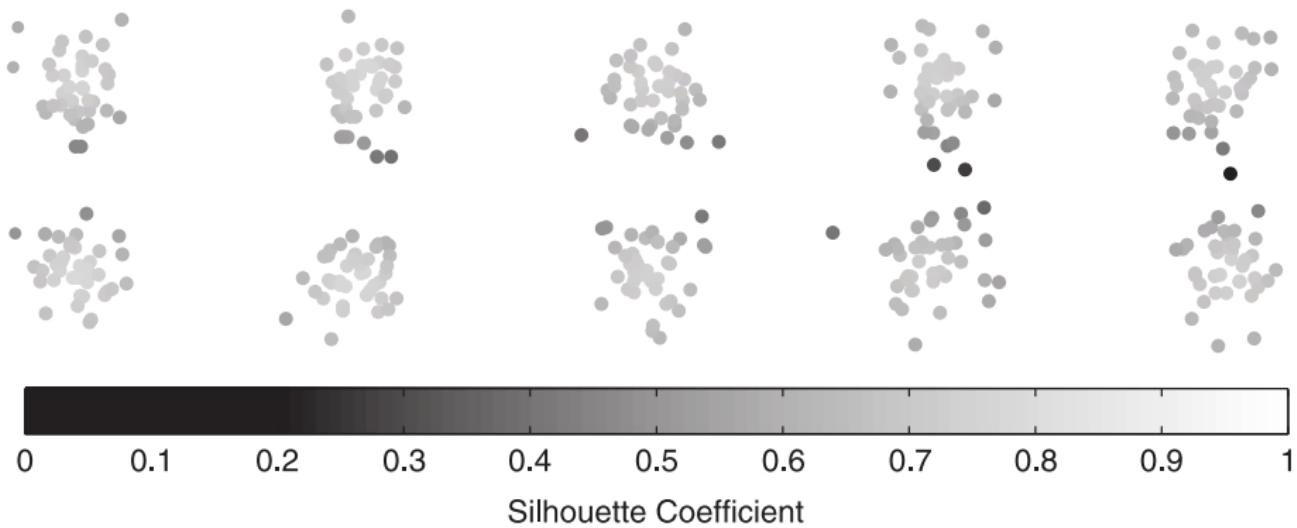


Figure 7.29. Silhouette coefficients for points in ten clusters.

Unsupervised Cluster Evaluation: Correlation

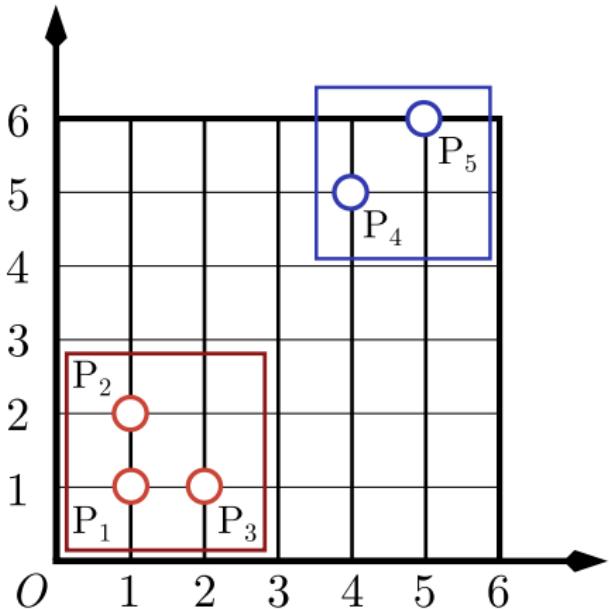
- Two matrices:
 - 1 Proximity Matrix
 - A **similarity matrix** can also be obtained by **transforming / normalizing the distances** using the formula:

$$s = 1 - \frac{d - d_{min}}{d_{max} - d_{min}} \quad (7)$$

2 Ideal Similarity Matrix

- One row and one column for each data point.
- An entry is 1 if the associated pair of points belong to the same cluster.
- An entry is 0 if the associated pair of points belongs to different clusters.

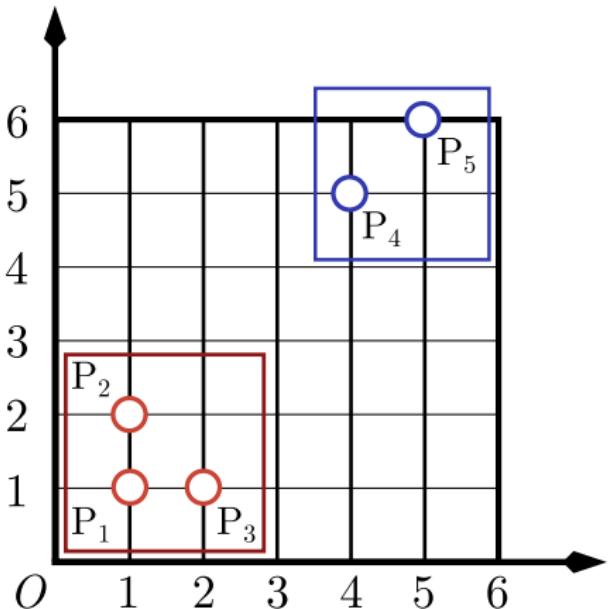
Unsupervised Cluster Evaluation: Correlation



P	X	Y
1	1	1
2	1	2
3	2	1
4	4	5
5	5	6

Ideal Similarity Matrix

Unsupervised Cluster Evaluation: Correlation

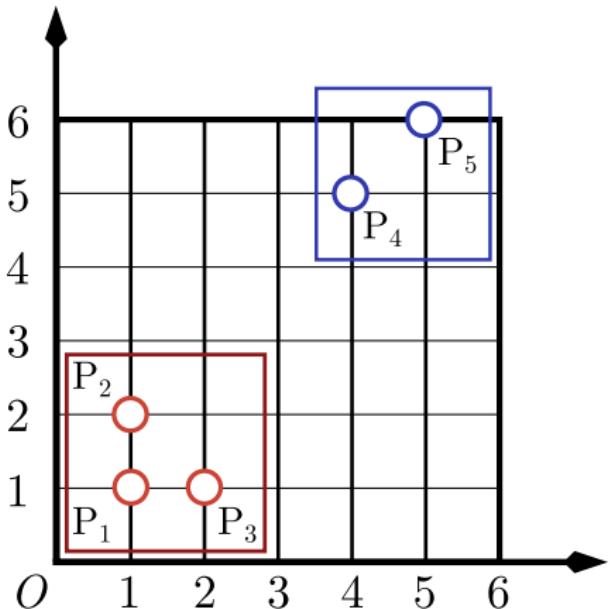


P	X	Y
1	1	1
2	1	2
3	2	1
4	4	5
5	5	6

	1	2	3	4	5
1	0	1	1	5	6.4
2	1	0	1.4	4.2	5.7
3	1	1.4	0	4.5	5.8
4	5	4.2	4.5	0	1.4
5	6.4	5.7	5.8	1.4	0

Proximity Matrix

Unsupervised Cluster Evaluation: Correlation



P	X	Y
1	1	1
2	1	2
3	2	1
4	4	5
5	5	6

	1	2	3	4	5
1	1	0.8	0.8	0.2	0
2	0.8	1	0.8	0.3	0.1
3	0.8	0.8	1	0.3	0.1
4	0.2	0.3	0.3	1	0.8
5	0	0.1	0.1	0.8	1

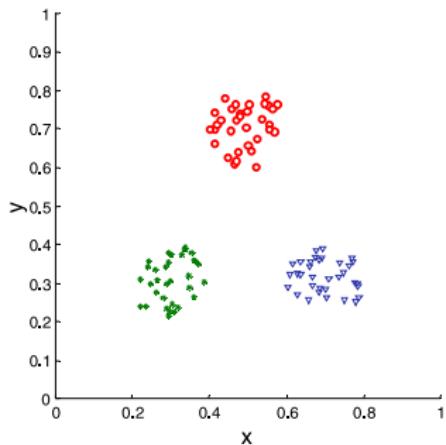
Similarity Matrix

Unsupervised Cluster Evaluation: Correlation

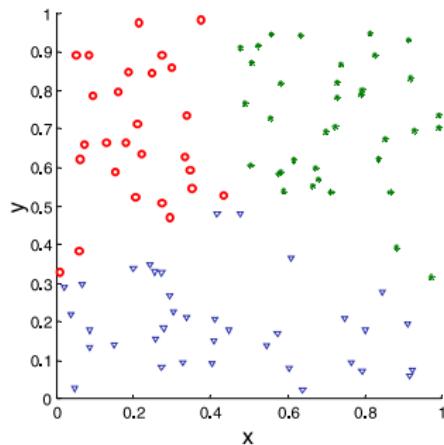
- Compute the correlation between the two matrices.
 - Since the **matrices are symmetric**, only the correlation between $\frac{n \cdot (n-1)}{2}$ entries needs to be calculated.
- **High magnitude of correlation** indicates that points that belong to the same cluster are close to each other.
 - Correlation may be positive or negative depending on whether the similarity matrix is a similarity or dissimilarity matrix.
- Not a good measure for some density or contiguity based clusters.

Unsupervised Cluster Evaluation: Correlation

- Correlation of ideal similarity and proximity matrices for the K-means clusterings of the following two data sets.



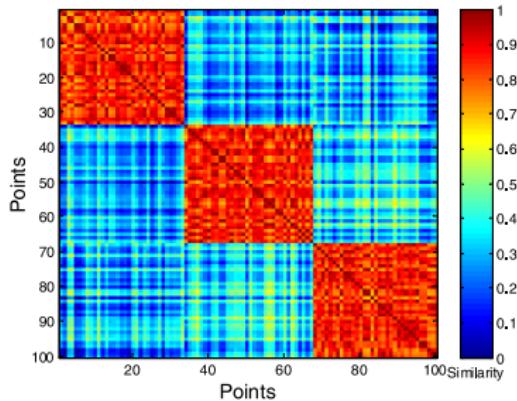
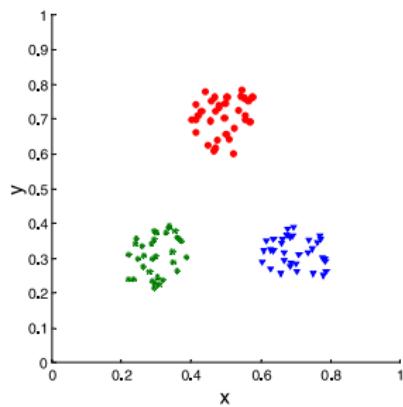
Corr = -0.9235



Corr = -0.5810

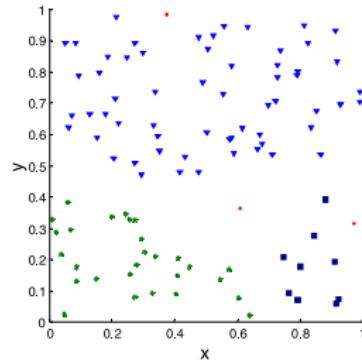
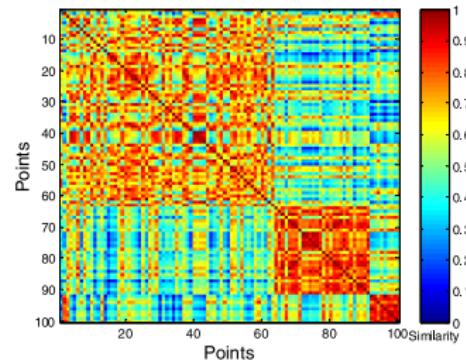
Unsupervised Cluster Evaluation: Similarity Matrix

- Order the similarity matrix with respect to cluster labels and inspect visually.



Unsupervised Cluster Evaluation: Similarity Matrix

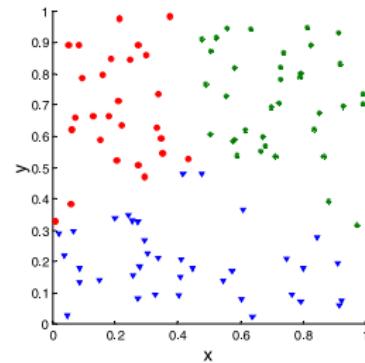
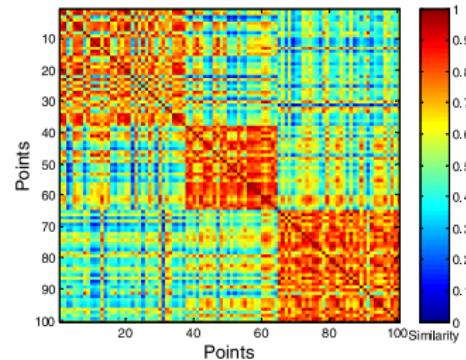
- Clusters in random data are not so crisp.



DBSCAN

Unsupervised Cluster Evaluation: Similarity Matrix

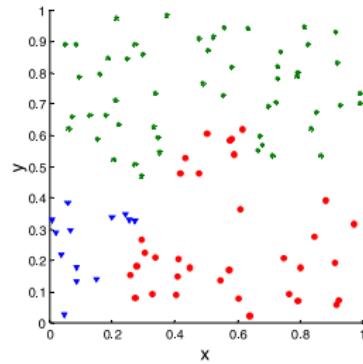
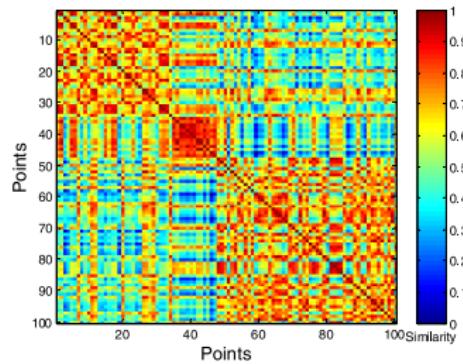
- Clusters in random data are not so crisp.



K-means

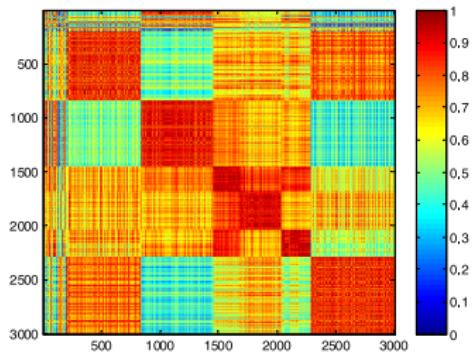
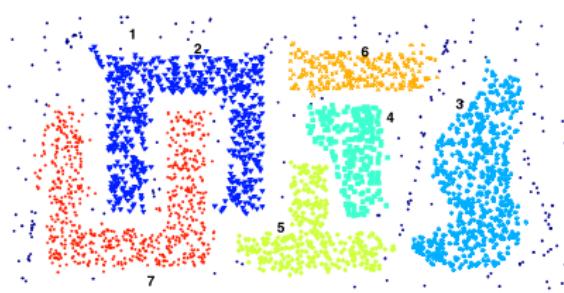
Unsupervised Cluster Evaluation: Similarity Matrix

- Clusters in random data are not so crisp.



Complete Link

Unsupervised Cluster Evaluation: Similarity Matrix

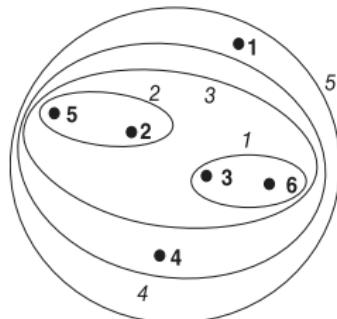


DBSCAN

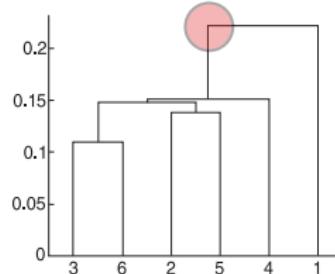
Unsupervised Cluster Evaluation: Cophenetic Correlation

- Cophenetic Distance between two objects is the proximity at which an agglomerative hierarchical clustering technique puts the objects in the same cluster for the first time.
- Cophenetic distance matrix, the entries are the cophenetic distances between each pair of objects.
- Cophenetic Correlation Coefficient (CPCC) is the correlation between the entries of this matrix and the original dissimilarity matrix.
- It is a standard measure of how well a hierarchical clustering fits the data.
- Common use is to evaluate which type of hierarchical clustering is best for a particular type of data.

Unsupervised Cluster Evaluation: Cophenetic Correlation



(a) Single link clustering.



(b) Single link dendrogram.

Figure 7.16. Single link clustering of the six points shown in Figure 7.15.

Table 7.7. Cophenetic distance matrix for single link and data in Table 2.14 on page 90.

Point	P1	P2	P3	P4	P5	P6
P1	0	0.222	0.222	0.222	0.222	0.222
P2	0.222	0	0.148	0.151	0.139	0.148
P3	0.222	0.148	0	0.151	0.148	0.110
P4	0.222	0.151	0.151	0	0.151	0.151
P5	0.222	0.139	0.148	0.151	0	0.148
P6	0.222	0.148	0.110	0.151	0.148	0

Technique	CPCC
Single Link	0.44
Complete Link	0.63
Group Average	0.66
Ward's	0.64

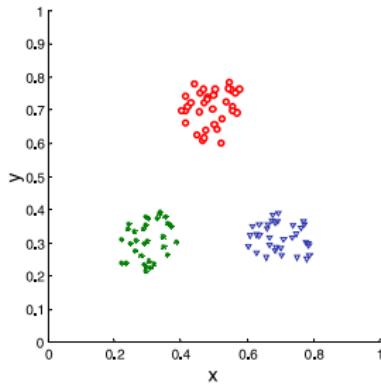
Clustering Tendency — Hopkins Statistic

- 1 Generate p points that are randomly distributed across the data space.
- 2 Sample p actual data points.
- 3 For both sets of points, we find the distance to the nearest neighbor in the original data set.
- 4 Let the u_i be the nearest neighbor distances of the artificially generated points, while the w_i are the nearest neighbor distances of the sample of points from the original data set.
- 5 The Hopkins statistic H is then defined by:

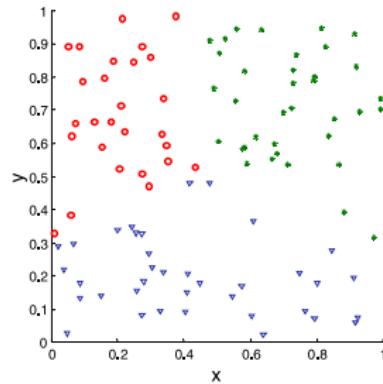
$$H = \frac{\sum_{i=1}^p w_i}{\sum_{i=1}^p u_i + \sum_{i=1}^p w_i} \quad (8)$$

Clustering Tendency — Hopkins Statistic

- $H \approx 0.5$: randomly generated points and the sample of data points have roughly the same nearest neighbor distances.
- $H \approx 1.0$: data that is regularly distributed in the data space.
- $H \approx 0.0$: data that is highly clustered.



Average Value of $H = 0.95$



Average Value of $H = 0.56$

Figure: Statistic accompanying the figures are wrong in the book.

Clustering Tendency — Hopkins Statistic

$$HS_i = \frac{\sum_{y_j \in \mathbf{R}_i} (\delta_{\min}(\mathbf{y}_j)))^d}{\sum_{y_j \in \mathbf{R}_i} (\delta_{\min}(\mathbf{y}_j)))^d + \sum_{x_j \in \mathbf{D}_i} (\delta_{\min}(\mathbf{x}_j)))^d} \quad (9)$$

- This statistic compares the nearest-neighbor distribution of randomly generated points to the same distribution for random subsets of points from \mathbf{D} .
- $\delta_{\min}(x_j) < \delta_{\min}(y_j)$: HS_i tends to 1.0: data is well clustered.
- $\delta_{\min}(x_j) \approx \delta_{\min}(y_j)$: HS_i tends to 0.5: data is essentially random.
- $\delta_{\min}(x_j) > \delta_{\min}(y_j)$: HS_i tends to 0.0: indicates point repulsion, with no clustering.

Clustering Tendency — Hopkins Statistic

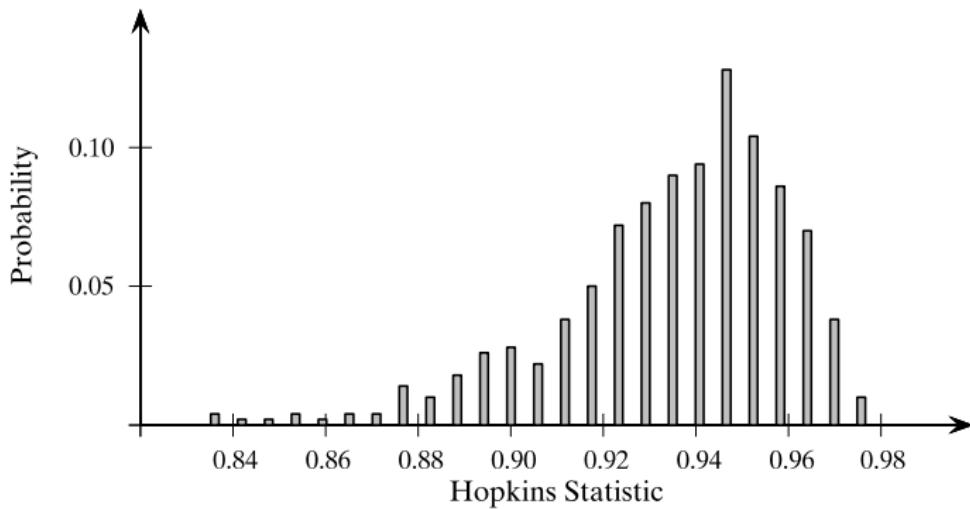


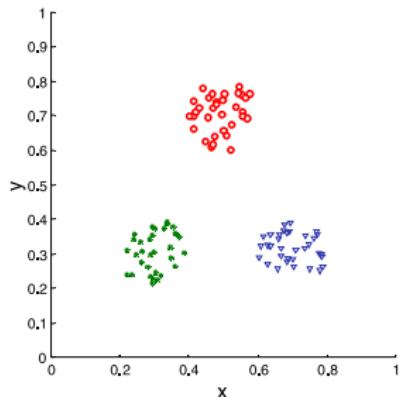
Figure 17.9. Iris dataset: Hopkins statistic distribution.

Framework for Cluster Validity

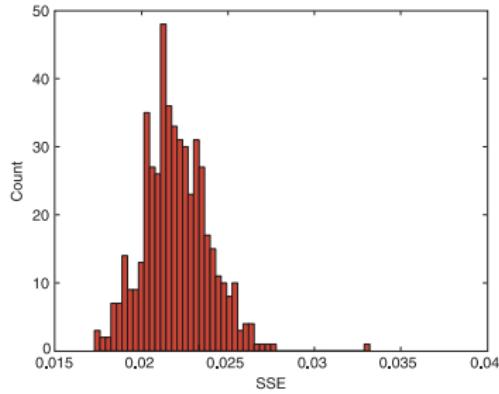
- Need a **framework to interpret any measure.**
 - For example, if our measure of evaluation has the value, 10, is that good, fair, or poor?
- Statistics provide a framework for cluster validity.
 - The **more “atypical” a clustering result is, the more likely it represents valid structure in the data.**
 - Can compare the values of an index that result from random data or clusterings to those of a clustering result.
 - If the **value of the index is unlikely**, then the cluster results are valid.
- For comparing the results of **two different sets of cluster analyses, a framework is less necessary.**
 - However, there is the question of whether the difference between **two index values is significant.**

Statistical Framework for SSE

- Example: Compare SSE of three cohesive clusters against three clusters in random data.



SSE = 0.005



Histogram shows SSE of three clusters in 500 sets of random data points of size 100 distributed over the range 0.2 – 0.8 for x and y values.

Supervised Cluster Evaluation

- We will cover supervised evaluation metrics (e.g., Precision, Recall, Entropy) in classification.