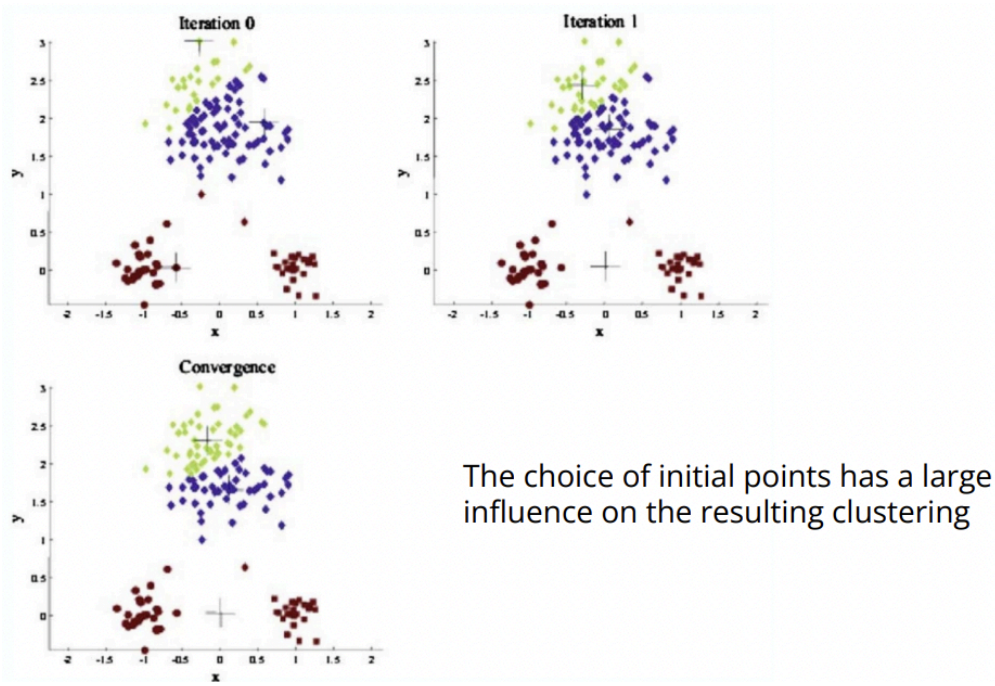
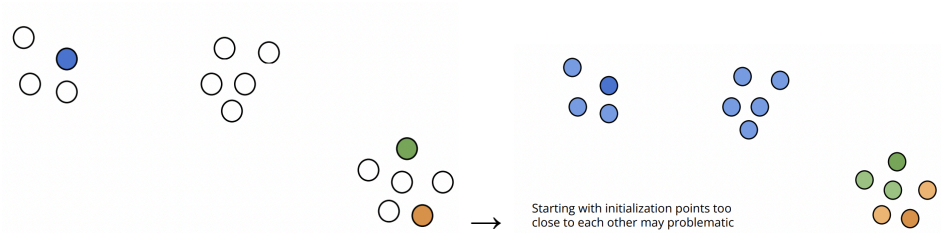
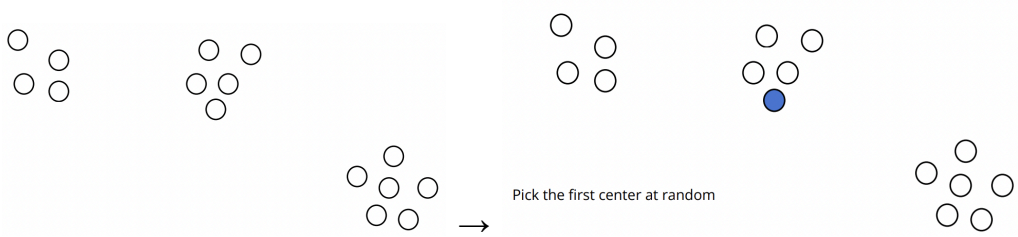
CAS CS 506
Lec 06

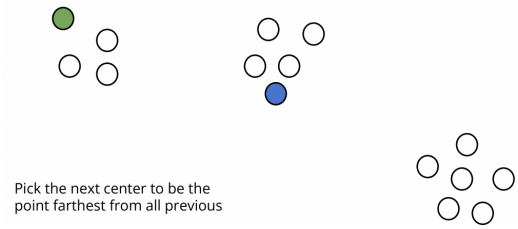
Kmeans++ & Hierarchical Clustering

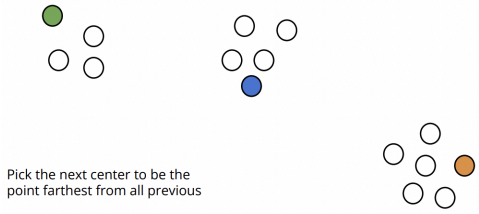
1. K-means - Lloyd's Algorithm
    a. Will this algorithm always converge?
    b. Proof (by contradiction): Suppose it does not converge. Then, either
        i. The minimum of the cost function is only reached in the limit (i.e. after an infinite number of iterations)
        Impossible because we are iterating over a finite set of partitions
        ii. The algorithm gets stuck in a cycle/loop
        Impossible since this would require having a clustering that has a lower cost than itself and we know
            1. If old != new clustering then the cost has improved
            2. If old != new clustering then the cost is unchanged
        iii. Conclusion: It always converges
    c. Will this always converge to the optimal solution?
    d. 

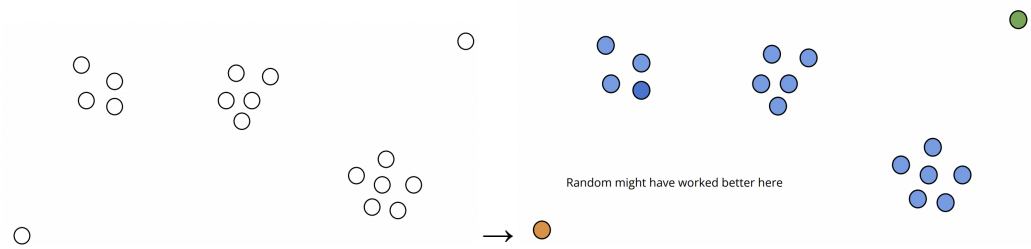
2. K-means - Initialization
    a. One solution: Run Lloyd's algorithm multiple times and choose the result with the lowest cost
    b. This can still lead to bad results because of randomness
    c. Another solution: Try different initialization methods

3. K-means - Random
    a. 

    b. Starting with initialization points too close to each other may be problematic
4. K-means - Farthest First Traversal
    a. 

    b. Pick the first center at random
    c. 

    d. Pick the next center to be the point farthest from all previous
    e. 

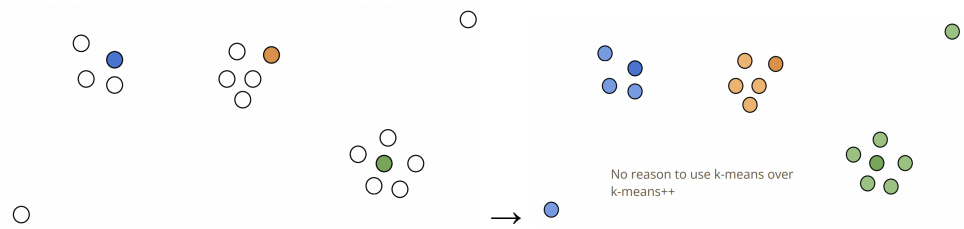
5. K-means - FFT and outliers
    a. 

    b. Random might have worked better here
6. K-means++
    a. Initialize with a combination of the two methods:
        i. Start with a random center

ii. Let D(x) be the distance between x and the centers selected so far. Choose the next center with probability proportional to D(x)^a

iii. When

1. a = 0: randomness initialization (all points have equal probability)
2. a = infinity: farthest first traversal
3. a = 2: k-means++

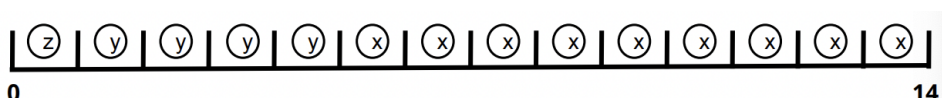
b. 

c. No reason to use k-means over k-means++

d. Suppose we are given a black box that will generate a uniform random number between 0 and any N. How can we use this black box to select points with probability proportional to D(x)^a?

i. Set a = 2

ii. 

iii. 

e. Q: the black box returns "12" as the random number generated. Which point do we choose for the next center (x, y, or z)?

i. A: x

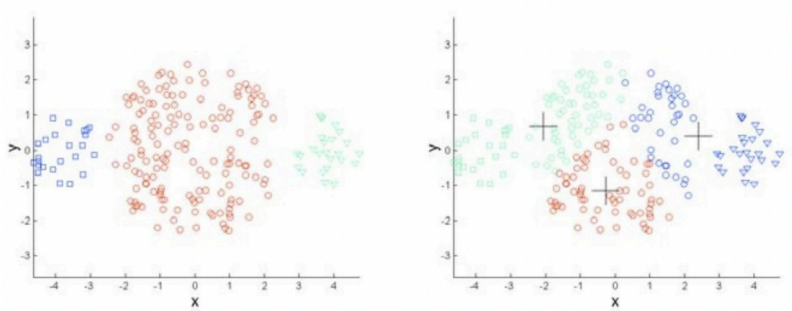
f. Q: the black box returns "4" as the random number generated. Which point do we choose for the next center (x, y, or z)?

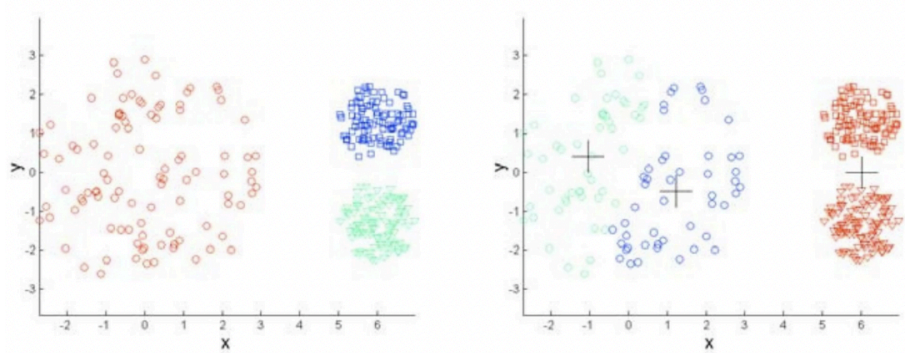
i. A: y

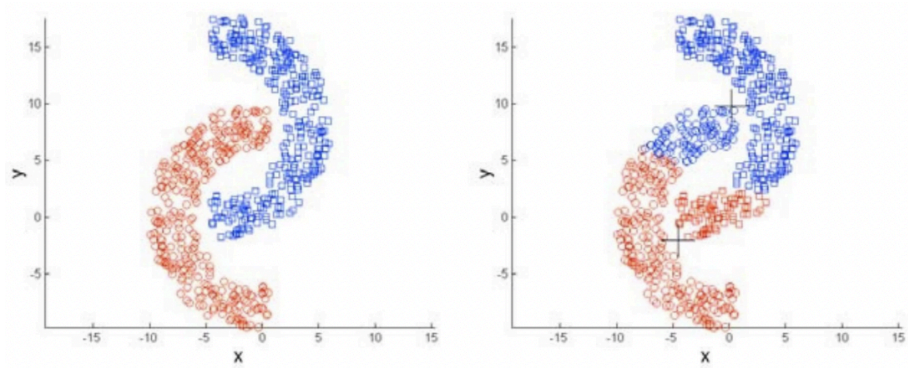
g. What happens if the black box can only generate numbers between 0 and 1?

i. A: z

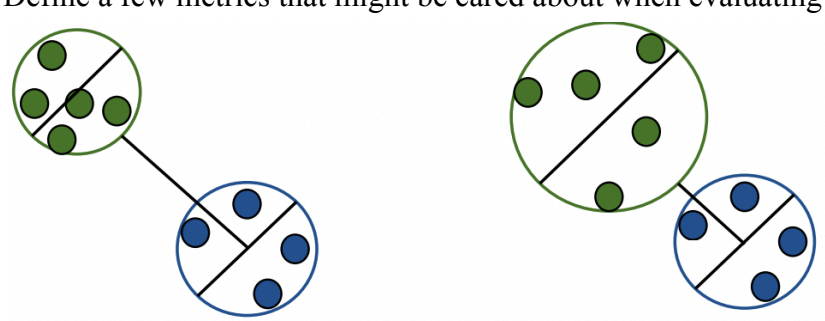
7. K-means / K-means++ Limitations

a. 

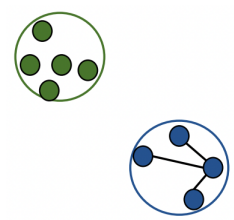
b. 

c. 

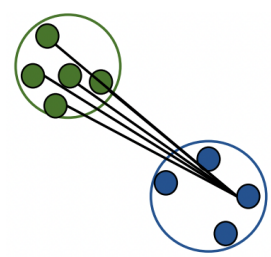
8. How to choose the right k?
    a. Iterate through different values of k (elbow method)
    b. Use empirical / domain-specific knowledge
        i. Example: Is there a known approximate distribution of the data? (K-means is good for spherical Gaussians)
    c. Metric for evaluating a clustering output
9. Evaluation
    a. Recall the goal: Find a clustering such that
        i. Similar data points are in the same cluster
        ii. Dissimilar data points are in the different clusters
    b. K-means cost function tells the within-cluster distances between points will be small overall
    c. But what about the intra-cluster distance? Are the clusters we created far? How far? Relative to what?
    d. Define a few metrics that might be cared about when evaluating a clustering



        i. a: average within-cluster distance
        ii. b: average intra-cluster distance
    e. What does it mean for (b-a) to be 0?
        i. A: The silhouette score in this case is 0, indicating that the sample could equally belong to one cluster or another, suggesting overlapping clusters or a sample that is not clearly defined within its cluster.
    f. What does it mean for (b-a) to be large?
        i. A: The silhouette score approaches +1, which is indicative of a well-clustered sample.
    g. What does it mean for (b-a) to be negative?
        i. A: The silhouette score would be negative, suggesting that the clustering configuration may need improvement.
    h. (b-a) / max(a,b)
10. Silhouette Scores
    a. For each data point i:
        i. a_i: mean distance from point i to every other point in its cluster

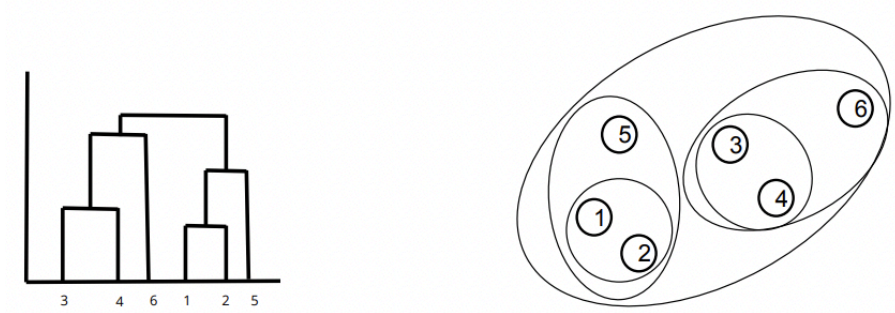


        ii. b_i: smallest mean distance from point i to every point in another cluster

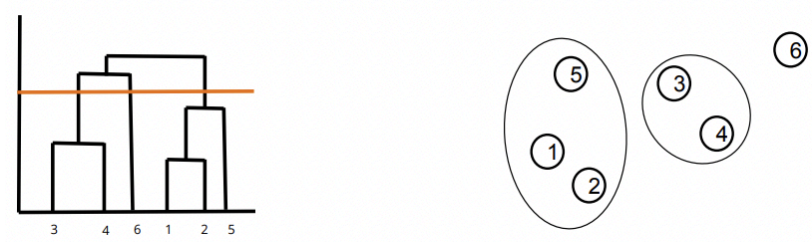


        iii. $s_i = (b_i - a_i) / \max(a_i, b_i)$ → Silhouette score plot
        OR
        iv. return the mean s_i over the entire dataset as a measure of goodness of fit
        v. What is a good silhouette score value?
            1. A: close to +1
        vi. What is a bad silhouette score value?
            1. A: close to -1
11. K-means Variations
    a. K-medians (uses the L_1 norm / manhattan distance)
    b. K-medoids (any distance function + the centers must be in the dataset)
    c. Weighted K-means (each point has a different weight when computing the mean)

12. Hierarchical Clustering
    a. At every step, we record which clusters were merged in order to produce a dendrogram



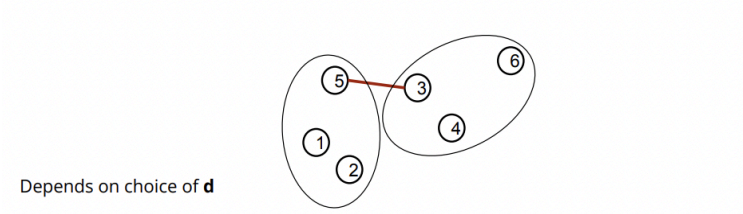
    b. We can "cut" the dendrogram at any threshold to produce any number of clusters



    c. Two types of hierarchical clustering
        i. Agglomerative
            1. Start with every point in its own cluster
            2. At each step, merge the two closest clusters
            3. Stop when every point is in the same cluster
        ii. Divisive
            1. Start with every point in the same cluster
            2. At each step, split until every point is in its own cluster
13. Agglomerative Clustering Algorithm
    a. Let each point in the dataset be in its own cluster
    b. Compute the distance between all pairs of clusters
    c. Merge the two closest clusters
    d. Repeat 2 & 3 until all points are in the same cluster
14. Hierarchical Clustering - Distance Functions
    a. Distance between points: $d(p_1, p_2)$
    b. Distance between clusters: $D(C_1, C_2)$

15. Single-Link Distance
    a. Minimum of all pairwise distances between a point from one cluster and a point from the other cluster

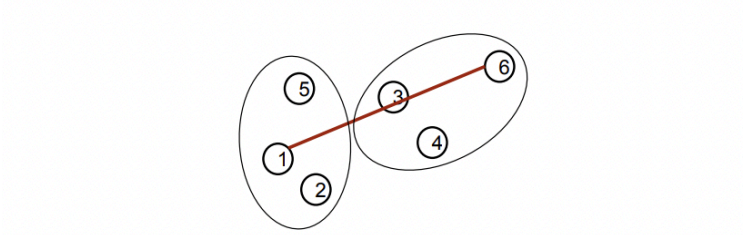
$$D_{SL}(C_1, C_2) = \min \{d(p_1, p_2) \mid p_1 \in C_1, p_2 \in C_2\}$$

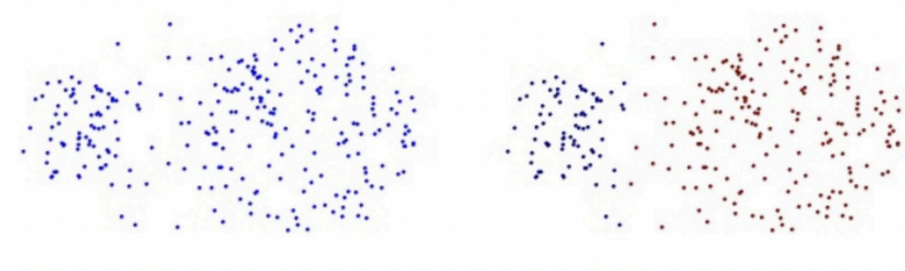


    b. 
    c. Can handle clusters of different sizes
    d. But… Sensitive to noise points and tends to create elongated clusters

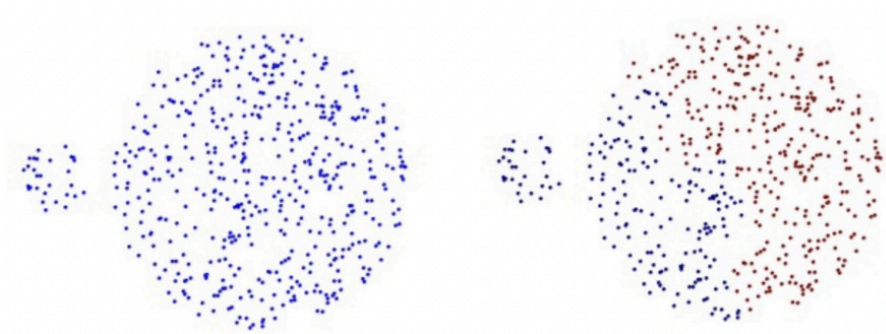
16. Complete-Link Distance
    a. Maximum of all pairwise distances between a point from one cluster and a point from the other cluster

$$D_{CL}(C_1, C_2) = \max \{d(p_1, p_2) \mid p_1 \in C_1, p_2 \in C_2\}$$



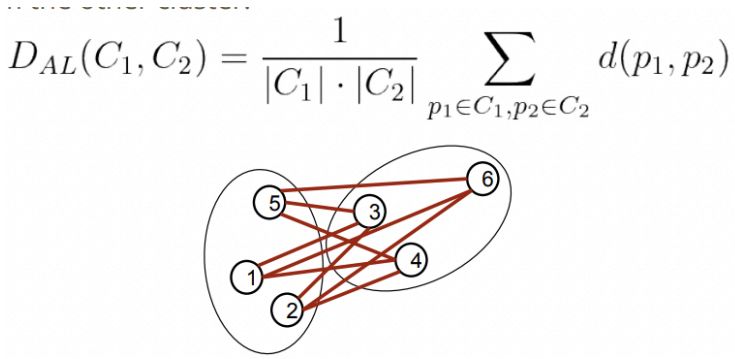
b. 

c. Less susceptible to noise and creates more balanced (equal diameter) clusters
d. But… tends to split up large clusters and all clusters tend to have the same diameter



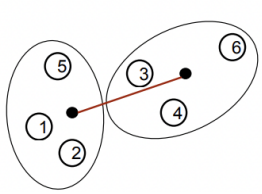
17. Average-Link Distance
    a. Average of all pairwise distances between a point from one cluster and a point from the other cluster

$$D_{AL}(C_1, C_2) = \frac{1}{|C_1| \cdot |C_2|} \sum_{p_1 \in C_1, p_2 \in C_2} d(p_1, p_2)$$



    b. Less susceptible to noise and outliers
    c. But… tends to be biased toward globular clusters
18. Centroid Distance
    a. The distance between the centroids of clusters

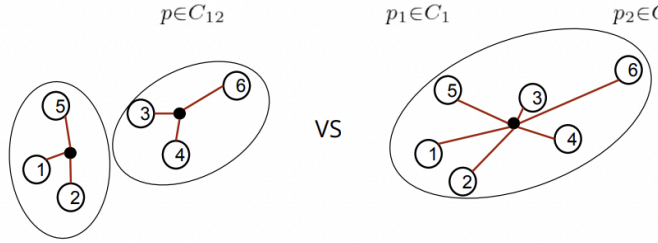
$$D_C(C_1, C_2) = d(\mu_1, \mu_2)$$



19. Ward's Distance

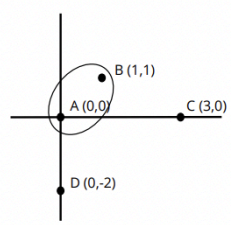
a. Difference between the spread / variance of points in the merged cluster and the unmerged clusters

$$D_{WD}(C_1, C_2) = \sum_{p \in C_{12}} d(p, \mu_{12}) - \sum_{p_1 \in C_1} d(p_1, \mu_1) - \sum_{p_2 \in C_2} d(p_2, \mu_2)$$



20. Agglomerative Clustering Algorithm Example

d = Euclidean
D = Single-Link

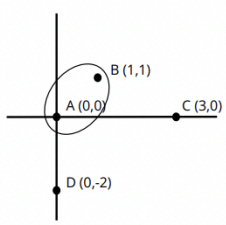


Distance Matrix

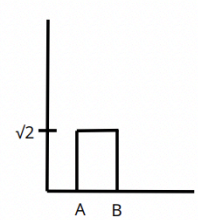
	A	B	C	D
A	0	√2	3	2
B	√2	0	√5	√10
C	3	√5	0	√13
D	2	√10	√13	0

a.

d = Euclidean
D = Single-Link

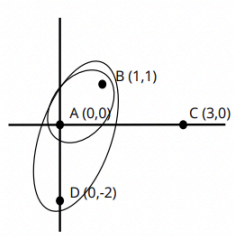


Dendrogram



b.

d = Euclidean
D = Single-Link

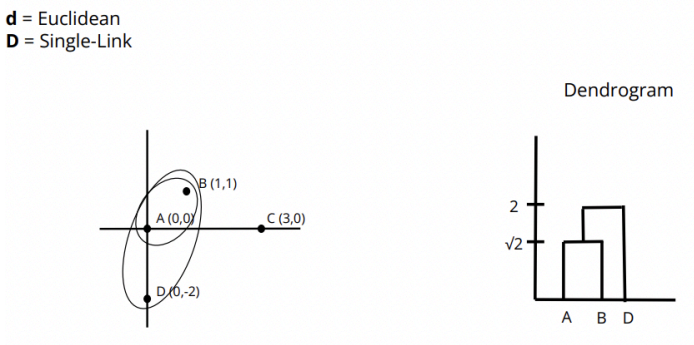


Distance Matrix

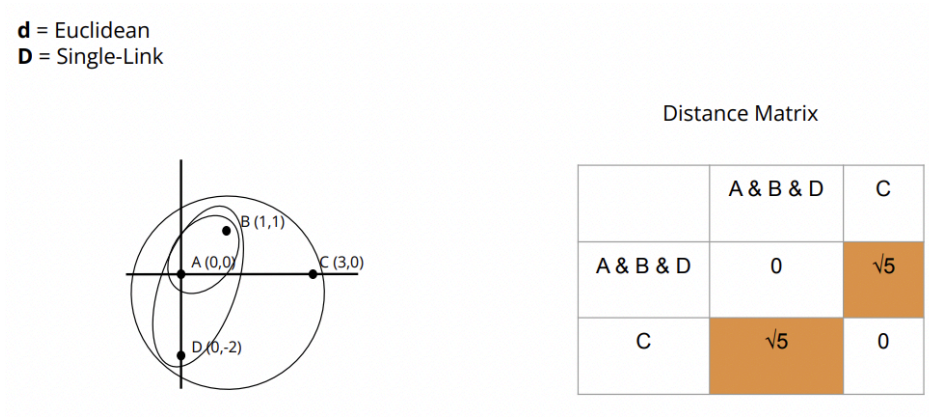
	A & B	C	D
A & B	0	√5	2
C	√5	0	√13
D	2	√13	0

c.

d.

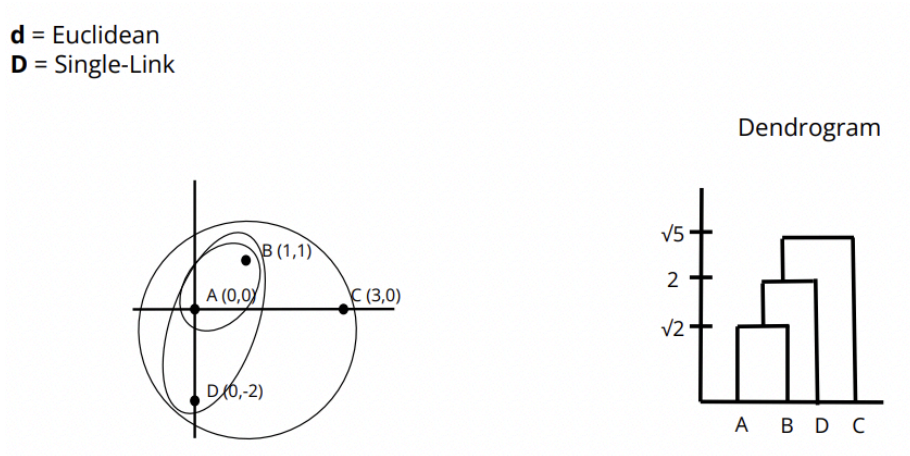


e.



	A & B & D	C
A & B & D	0	√5
C	√5	0

f.



21. Hierarchical Clustering
    a. Finding the threshold with which to cut the dendrogram requires exploration and tuning. But in general hierarchical clustering is used to expose a hierarchy in the data
    b. To capture the difference between clusterings you can use a cost function or methods that will be discussed later when looking at clustering aggregation