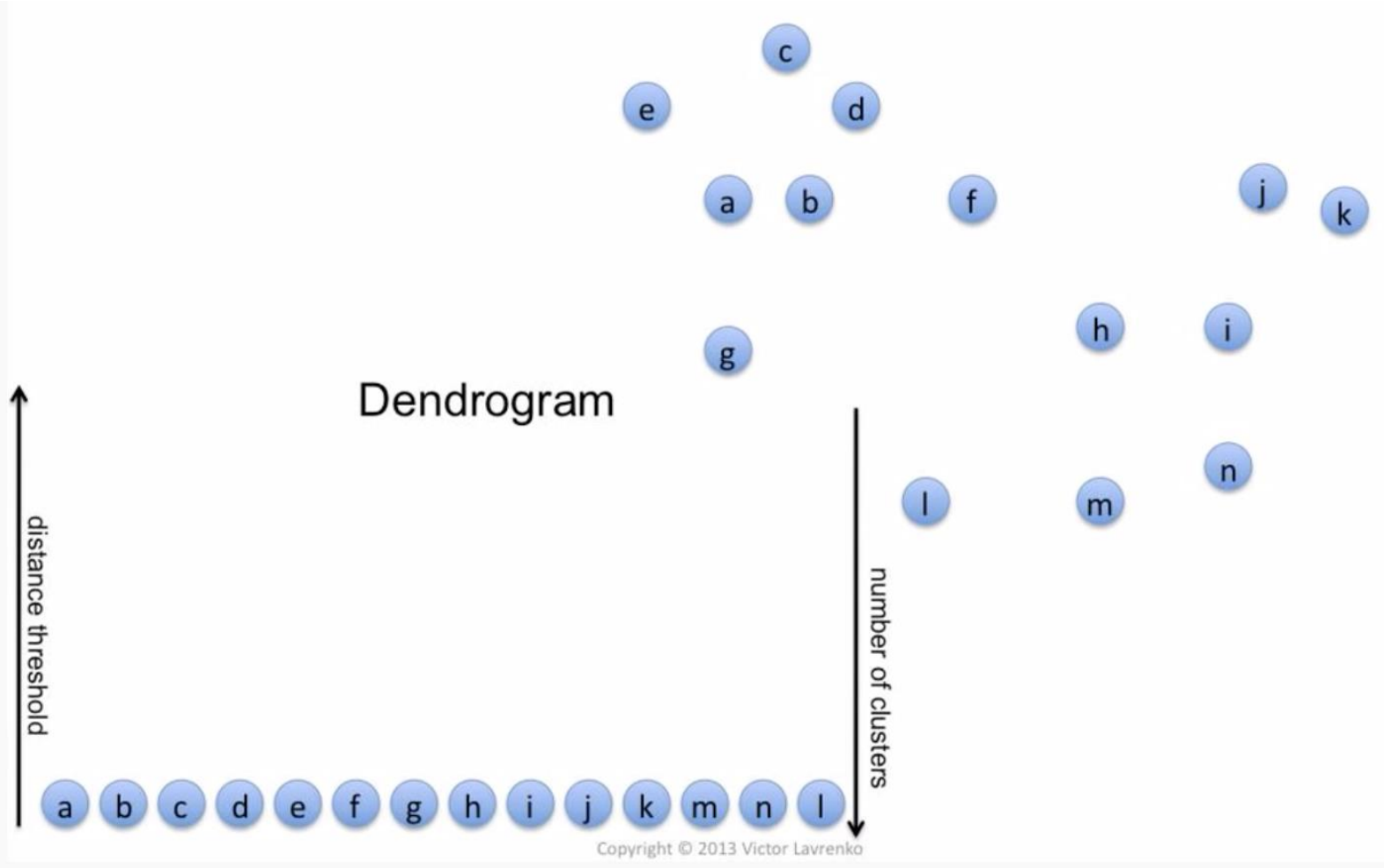
# Hierarchical Clustering

# References


- Emily Fox & Carlos Guestrin
- https://www.coursera.org/learn/ml-clustering-and-retrieval
- What is Hierarchical Clustering? – Kdnuggets
- Hierarchical Clustering | Hierarchical Clustering Python (analyticsvidhya.com)
- https://web.stanford.edu/class/cs276/handouts/lecture12-clustering.ppt
- Dr. Maggie Mashaly,GUC

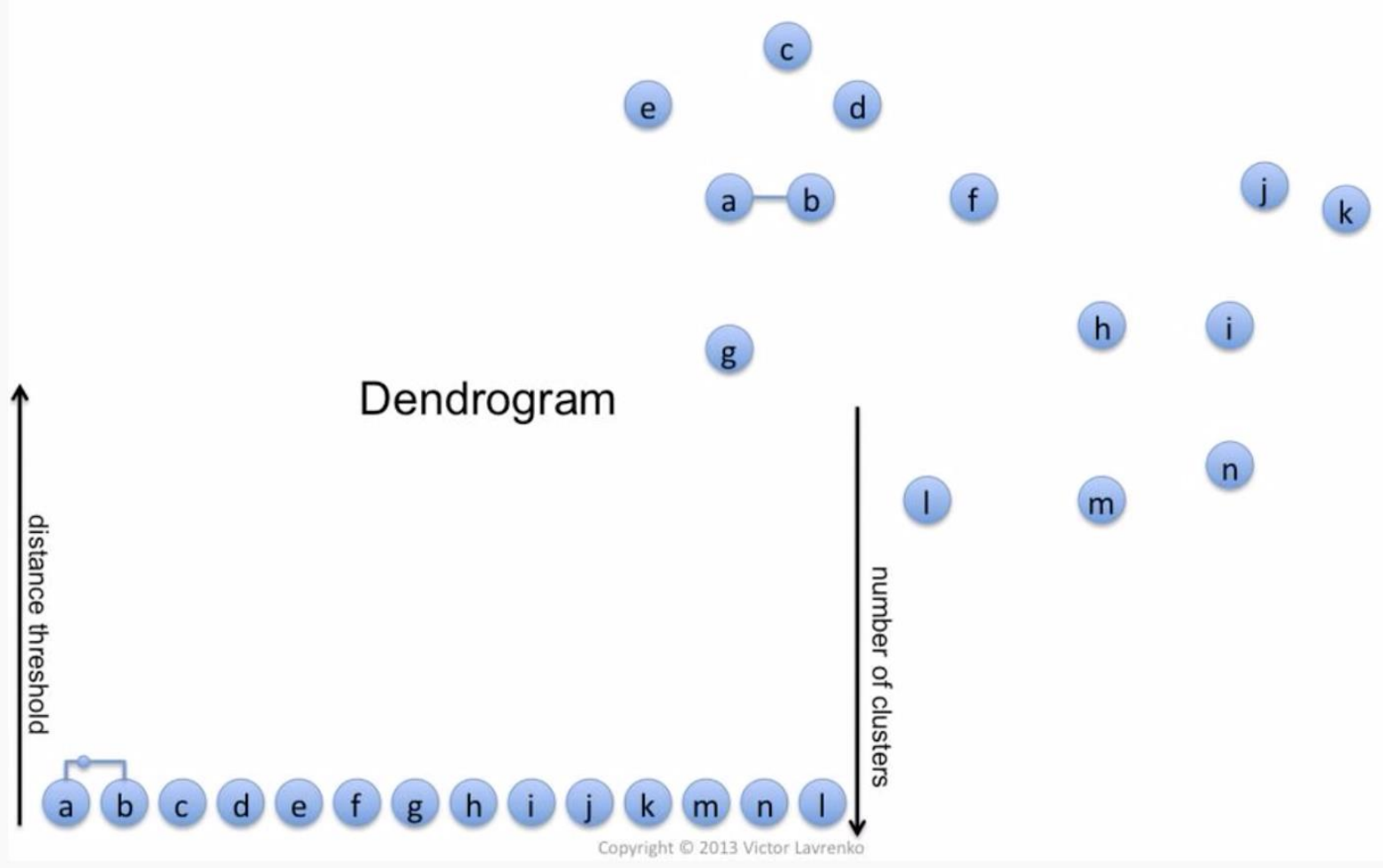
# Agglomerative Clustering: Example

1. Each point represents a Cluster



# Agglomerative Clustering: Example

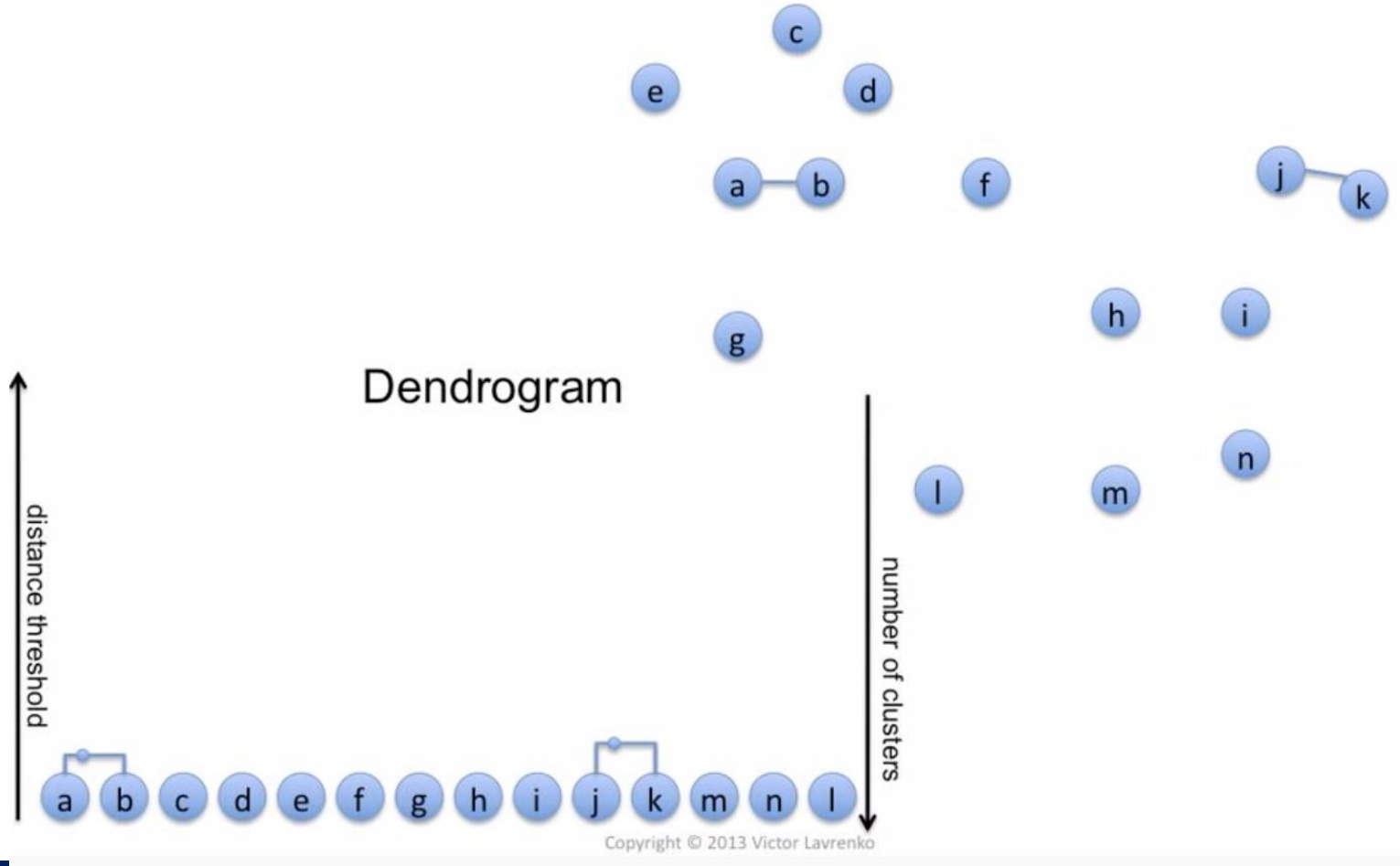
2. Look for a pair of clusters with minimum distance between them




- A & B are now one cluster
- The height of the edge in the dendrogram corresponds to the distance between the two clusters

# Agglomerative Clustering: Example

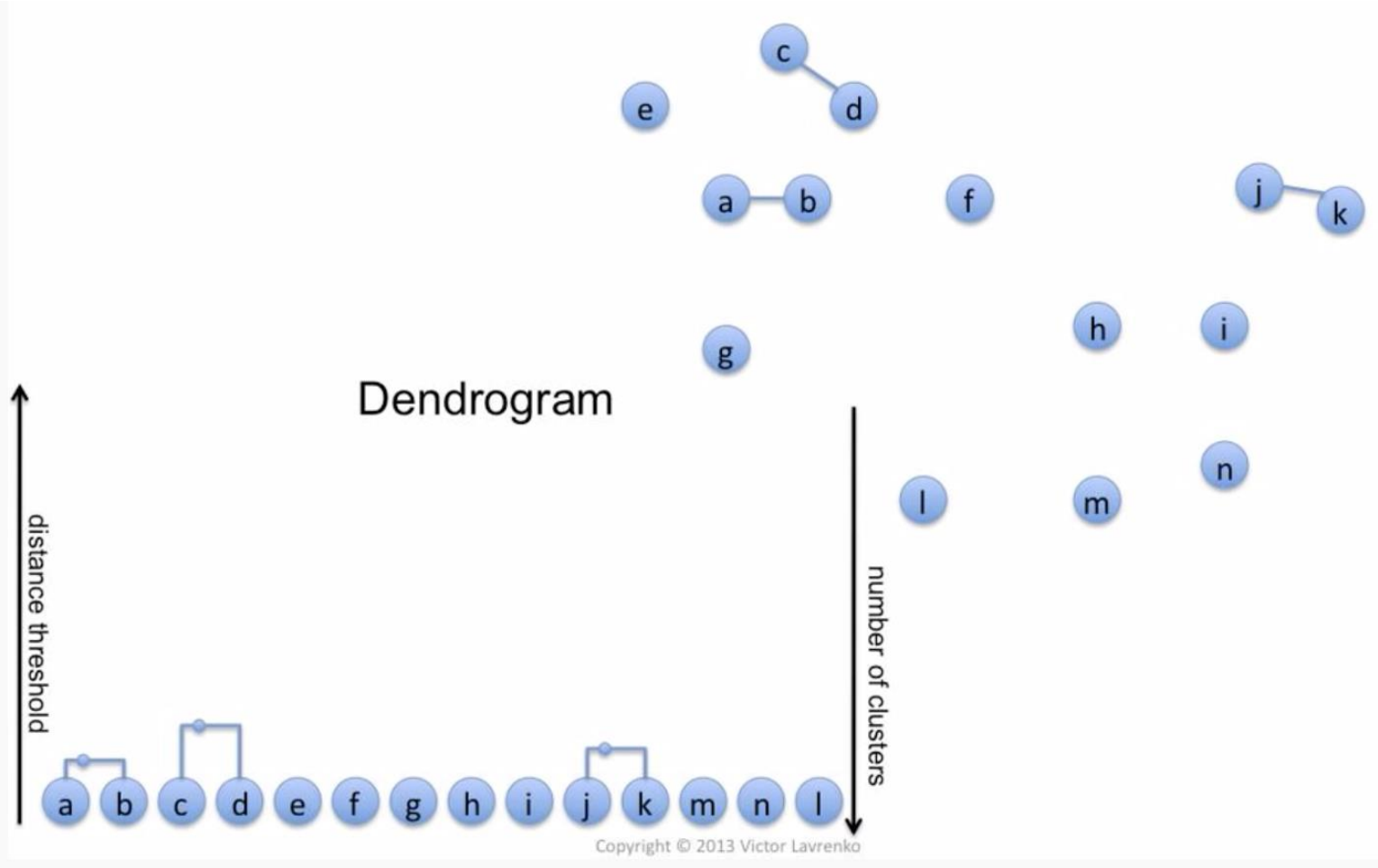
2. Look for a pair of clusters with minimum distance between them




- J & K are now one cluster

# Agglomerative Clustering: Example

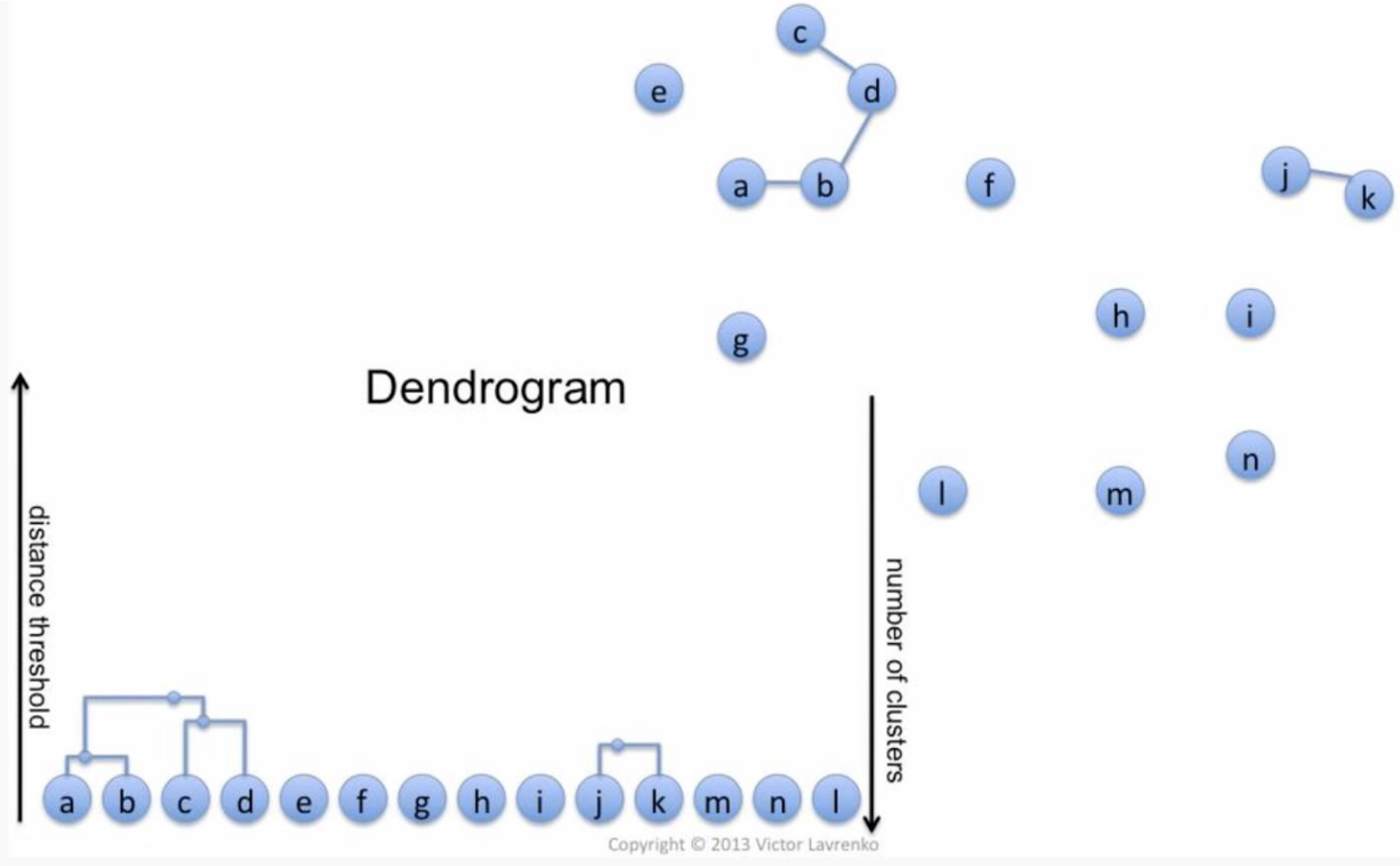
2. Look for a pair of clusters with minimum distance between them




- C & D are now one cluster

# Agglomerative Clustering: Example

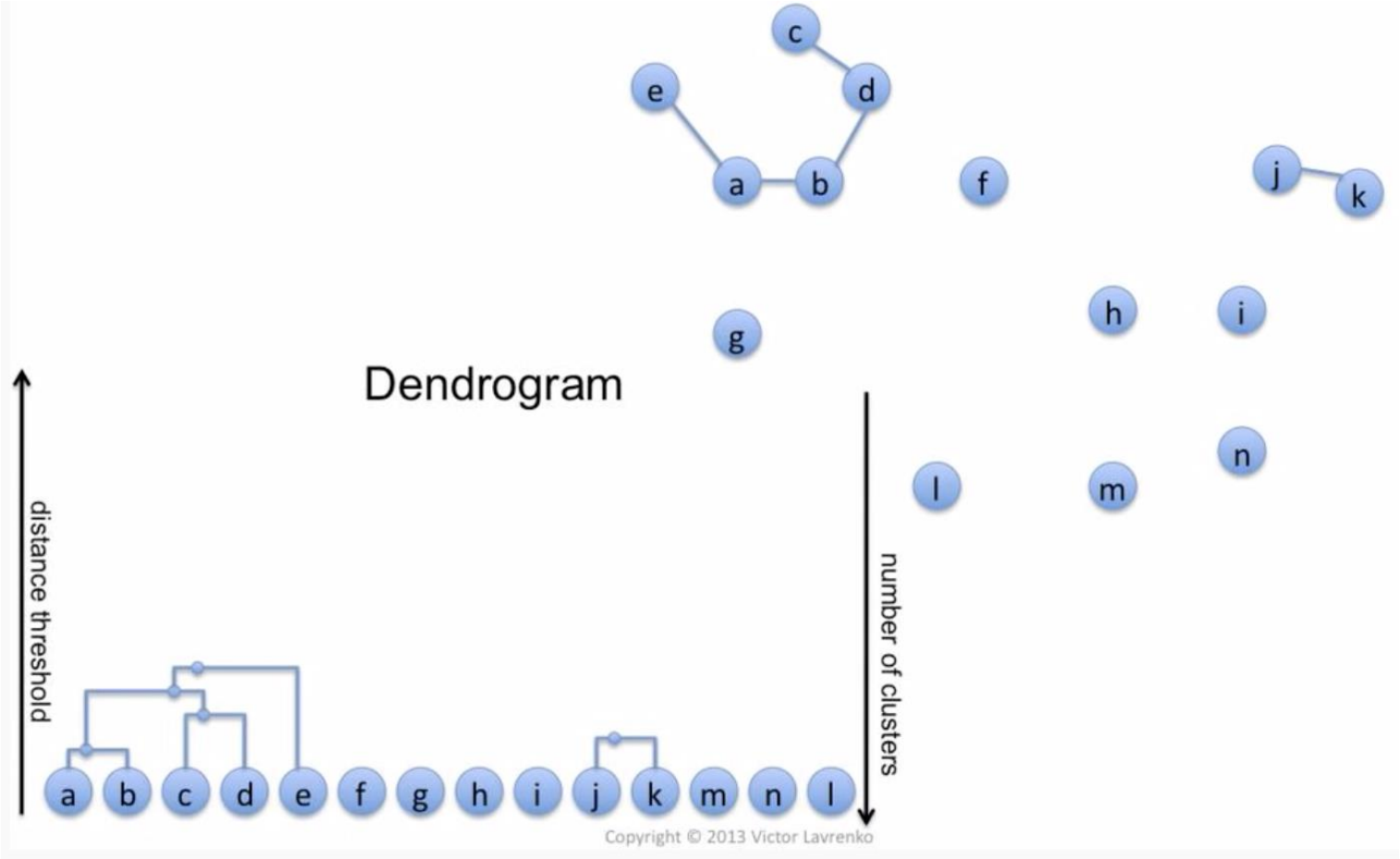
2. Look for a pair of clusters with minimum distance between them




- A, B, C & D are now one cluster

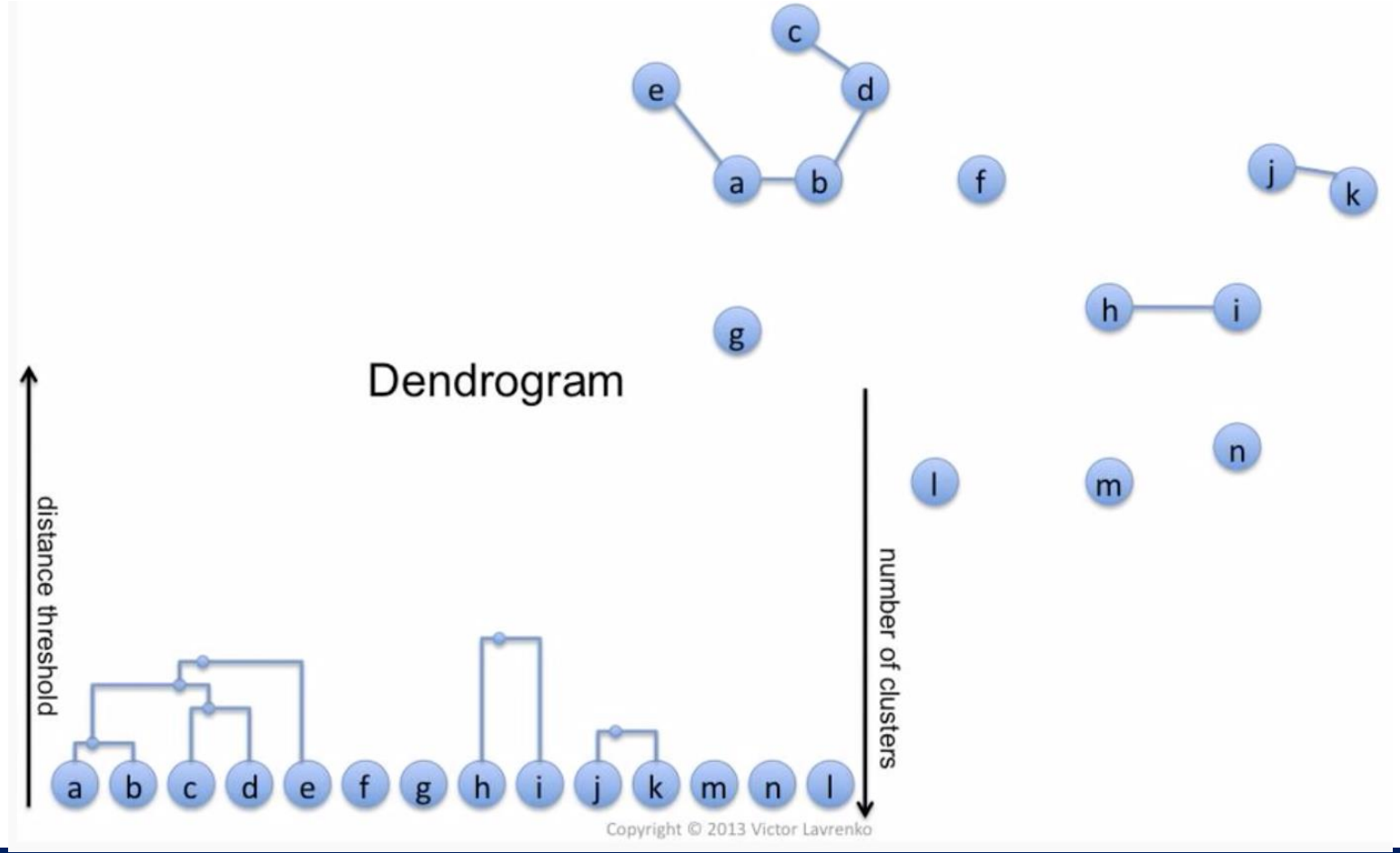
# Agglomerative Clustering: Example

2. Keep looking for a pair of clusters with minimum distance between them



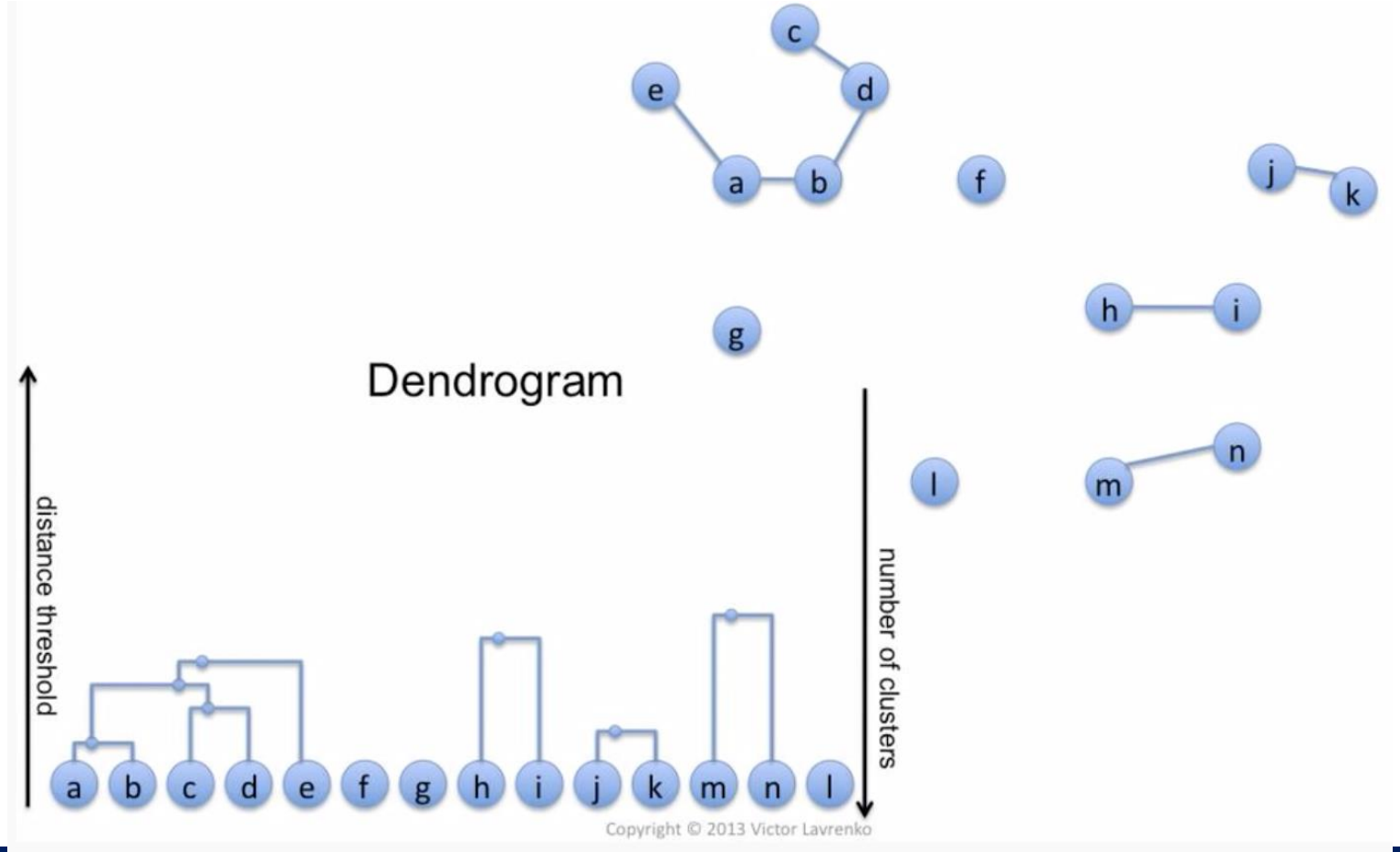
# Agglomerative Clustering: Example

2. Keep looking for a pair of clusters with minimum distance between them



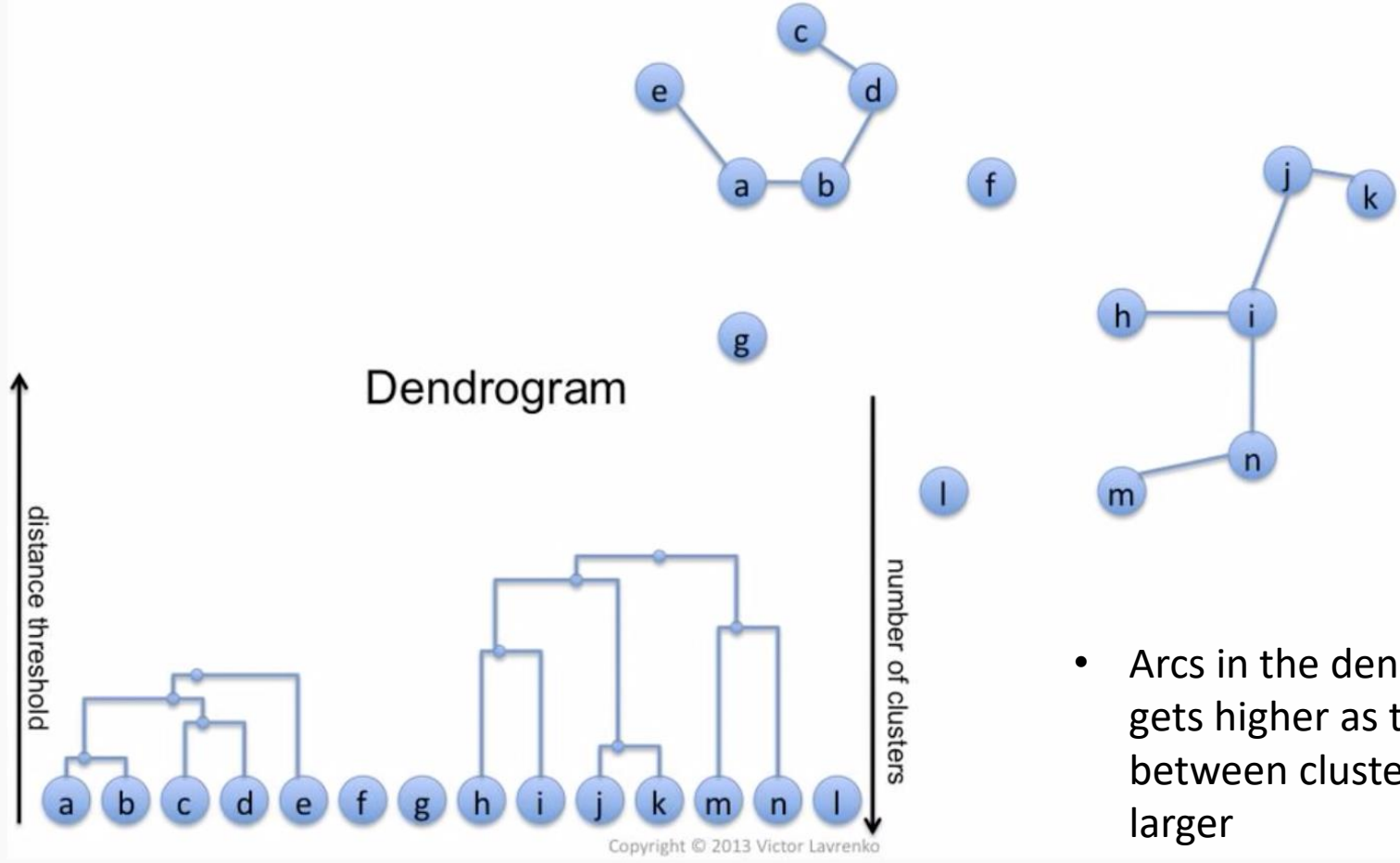
# Agglomerative Clustering: Example

2. Keep looking for a pair of clusters with minimum distance between them



# Agglomerative Clustering: Example

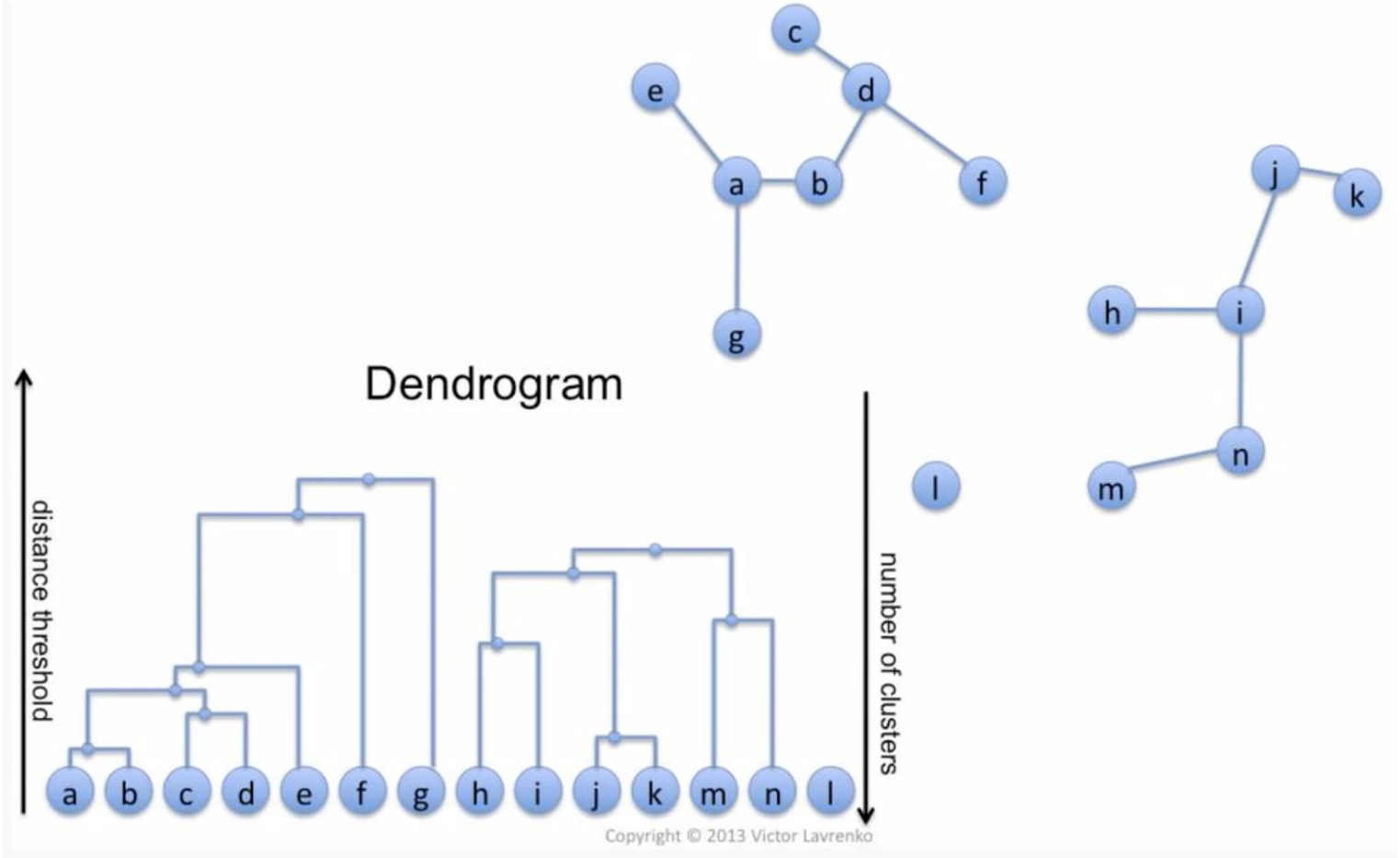
2. Keep looking for a pair of clusters with minimum distance between them




- Arcs in the dendrogram gets higher as the distance between clusters becomes larger

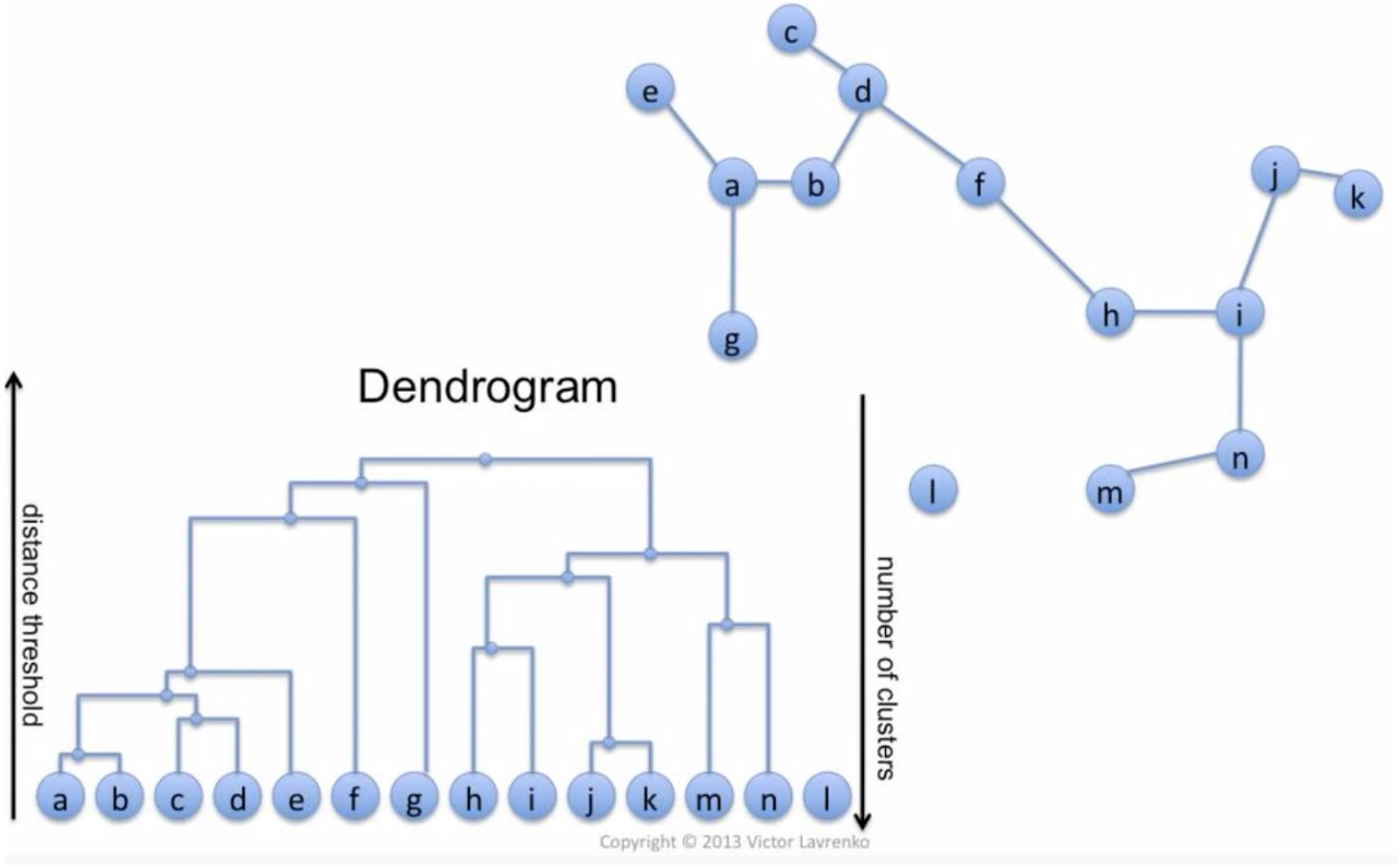
# Agglomerative Clustering: Example

2. Keep looking for a pair of clusters with minimum distance between them



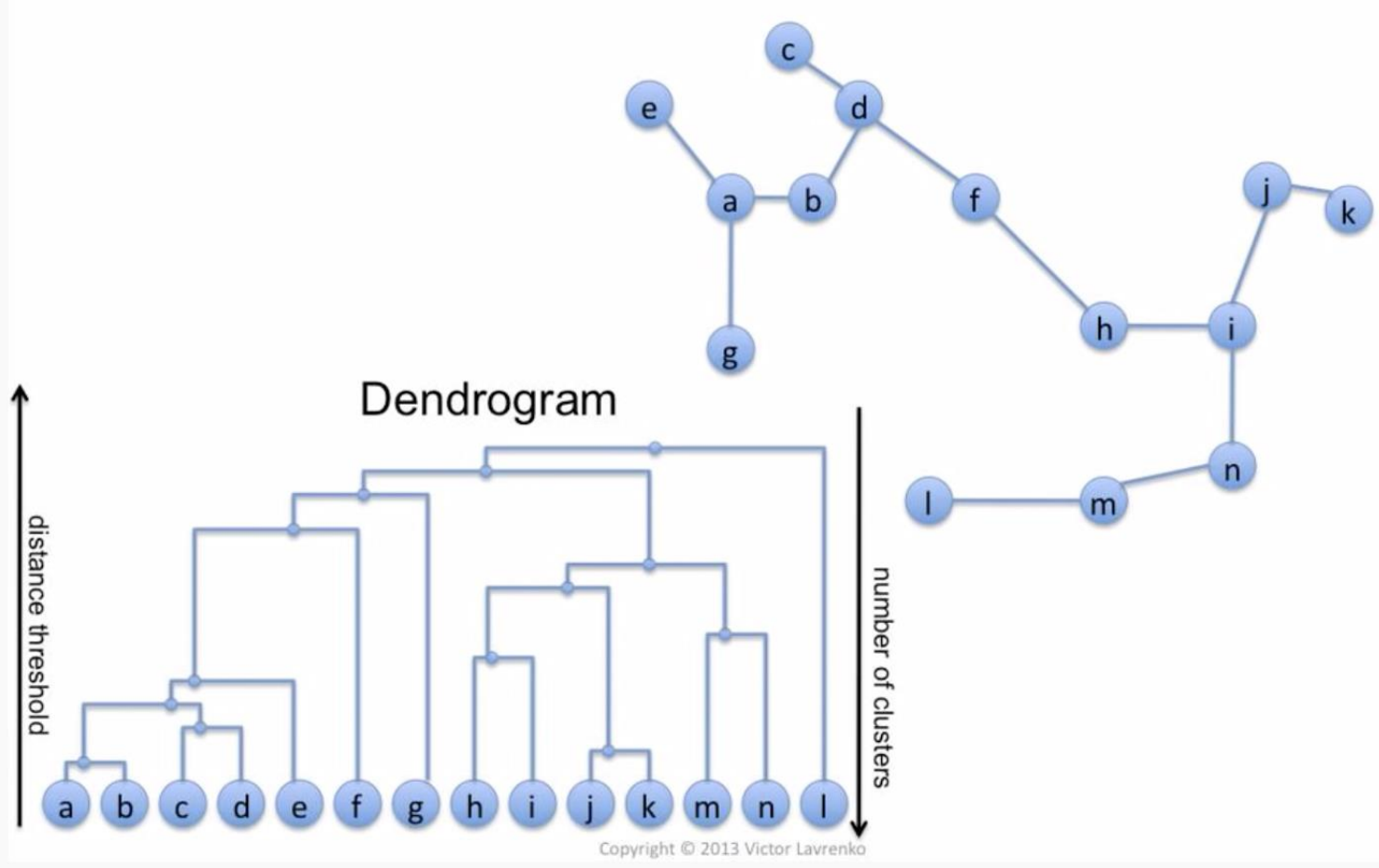
# Agglomerative Clustering: Example

2. Keep looking for a pair of clusters with minimum distance between them



# Agglomerative Clustering: Example

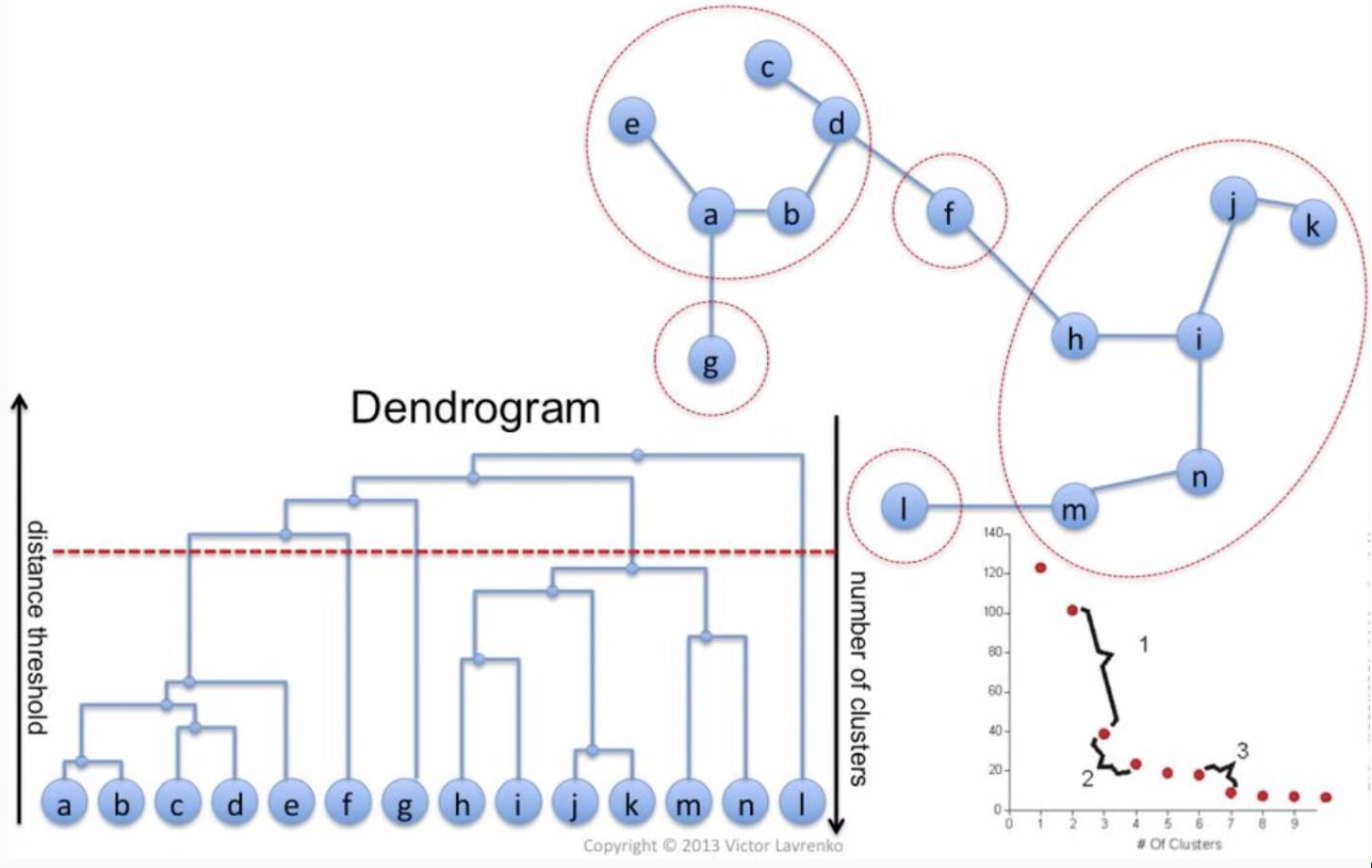
3. Stop when you end up with one cluster



# Agglomerative Clustering: Example

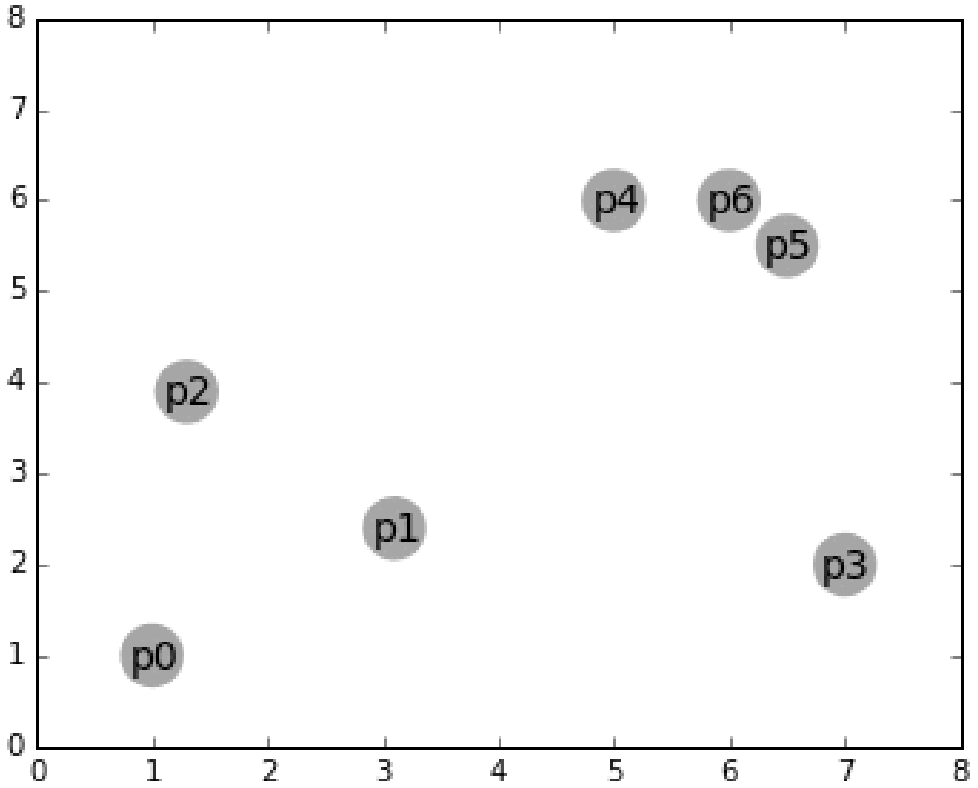
- How to decide on clusters?

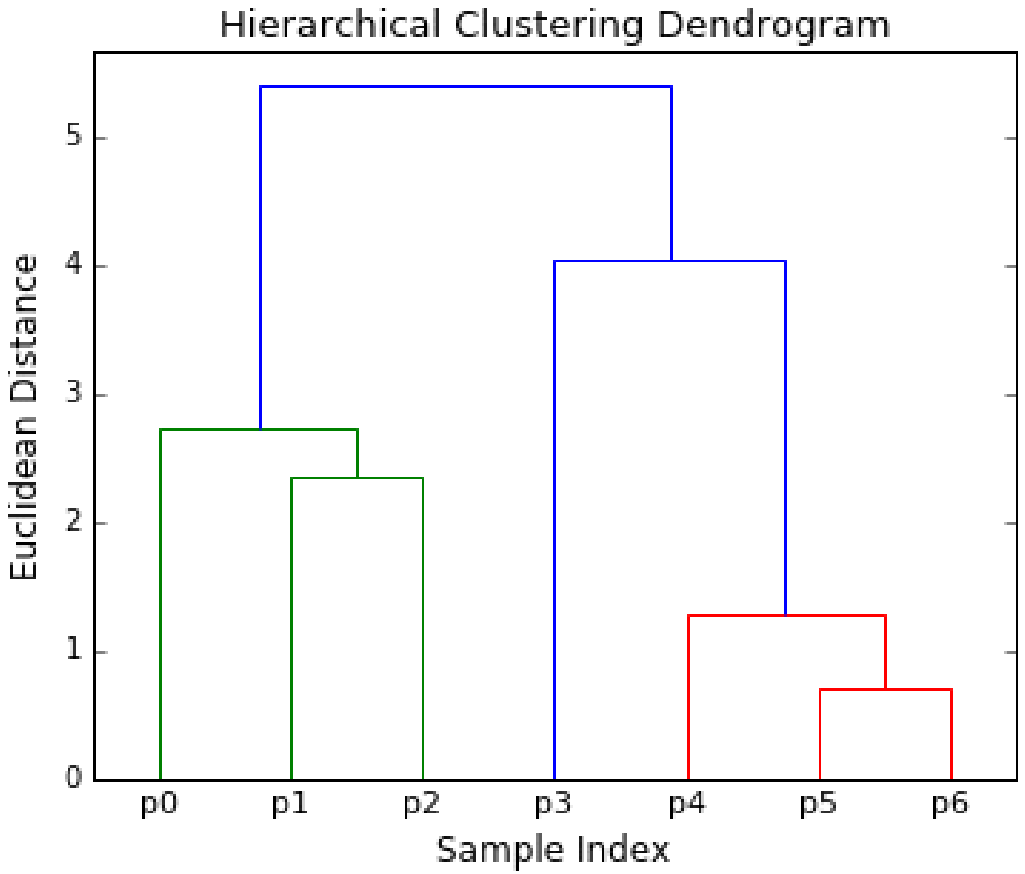
Pick a threshold distance and cut the tree at that distance




- Or by looking for the elbow in the plot of Number of clusters vs. distance

# Agglomerative Clustering



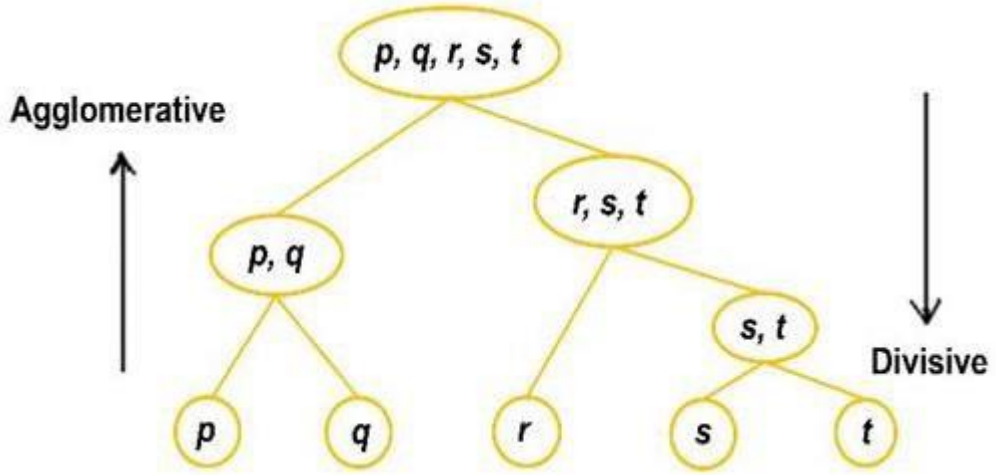
# Why Hierarchical Clustering Algorithm?

- help choosing # clusters beforehand
- Dendrograms help visualize different clustering granularities
  - No need to rerun algorithm
- Can often find more complex shapes than k-means or Gaussian mixture models.
- Suitable for clusters that have predominant ordering from top to bottom.

  For e.g: All files and folders on our hard disk are organized in a hierarchy.
-

# Hierarchical Clustering Types

- Divisive, a.k.a top-down: Start with all data in one big cluster and recursively split.
  - Example: recursive k-means
- Agglomerative a.k.a. bottom-up: Start with each data point as its own cluster. Merge clusters until all points are in one big cluster.
- - Example: single linkage

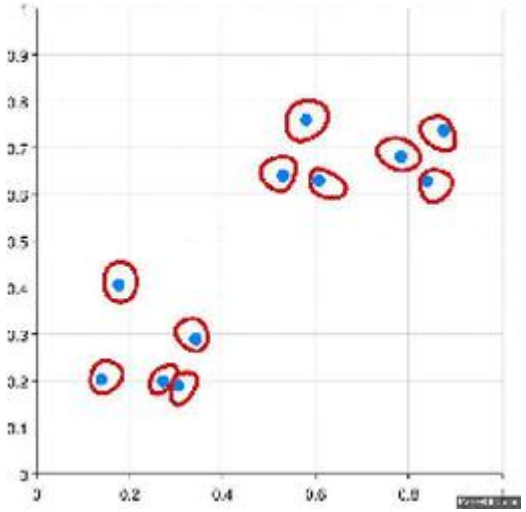


# Agglomerative Hierarchical Clustering

1.Make each data point a single-point cluster

2. based on the similarity of these clusters, we can combine the most similar clusters together

3.Repeat step-2 until you are left with only one cluster.



What is Hierarchical Clustering? - KDnuggets

https://www.coursera.org/learn/ml-clustering-and-retrieval

# Linkage Methods

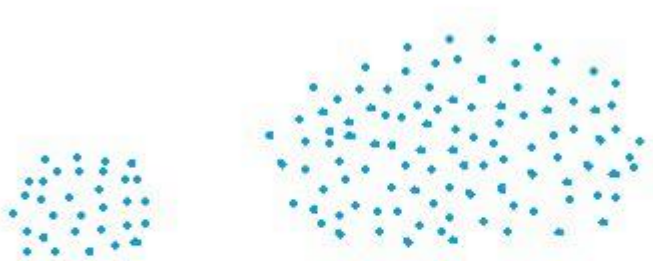
- Maximum or Complete-linkage: the distance between two clusters is defined as the longest distance between two points in each cluster. It tends to produce more compact clusters.
- **Minimum or** Single-linkage: the distance between two clusters is defined as the shortest distance between two points in each cluster. This linkage may be used to detect high values in your dataset which may be outliers as they will be merged at the end. It tends to produce long, “loose” clusters.
- Average-linkage: the distance between two clusters is defined as the average distance between each point in one cluster to every point in the other cluster.
- Centroid-linkage: finds the centroid of cluster 1 and centroid of cluster 2, and then calculates the distance between the two before merging.
- **Ward’s minimum variance method:** It minimizes the total within-cluster variance. At each step the pair of clusters with minimum between-cluster distance are merged.

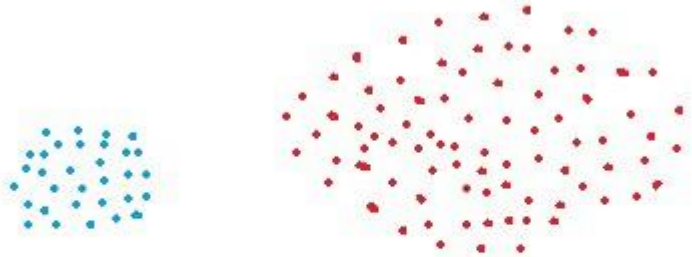
# MIN

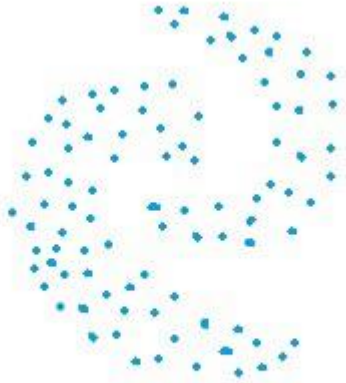
- Also known as single-linkage algorithm can be defined as the similarity of two clusters C1 and C2 is equal to the minimum of the similarity between points Pi and Pj such that Pi belongs to C1 and Pj belongs to C2.
- Mathematically this can be written as,
- $Sim(C1,C2) = \text{Min } Sim(Pi,Pj)$ such that $Pi \in C1$ & $Pj \in C2$
- In simple words, pick the two closest points such that one point lies in cluster one and the other point lies in cluster 2 and takes their similarity and declares it as the similarity between two clusters.

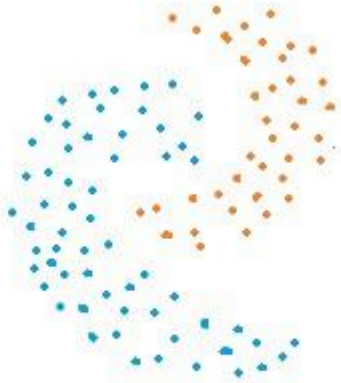
- Pros of MIN:
  - This approach can separate non-elliptical shapes as long as the gap between the two clusters is not small.
- Cons of MIN:
  - MIN approach cannot separate clusters properly if there is noise between clusters.

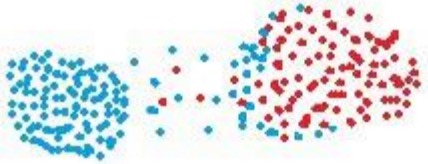
https://towardsdatascience.com/understanding-the-concept-of-hierarchical-clustering-technique-c6e8243758ec





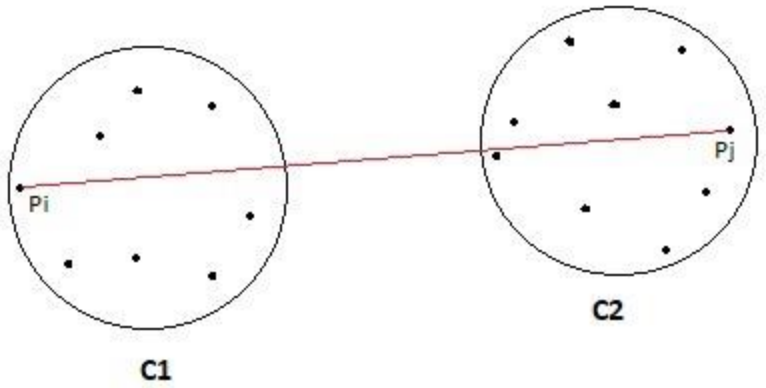




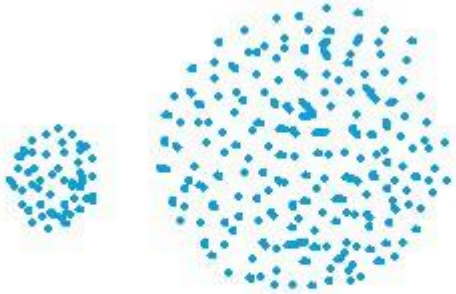


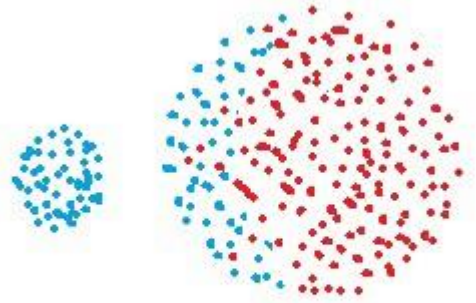
- **MAX:** Also known as the complete linkage algorithm, this is exactly opposite to the **MIN** approach. The similarity of two clusters C1 and C2 is equal to the **maximum** of the similarity between points Pi and Pj such that Pi belongs to C1 and Pj belongs to C2.
- Mathematically this can be written as,
- $Sim(C1,C2) = Max\ Sim(Pi,Pj)$ such that $Pi \in C1$ & $Pj \in C2$
- In simple words, pick the two farthest points such that one point lies in cluster one and the other point lies in cluster 2 and takes their similarity and declares it as the similarity between two clusters.

- Pros of MAX:
- MAX approach does well in separating clusters if there is noise between clusters.

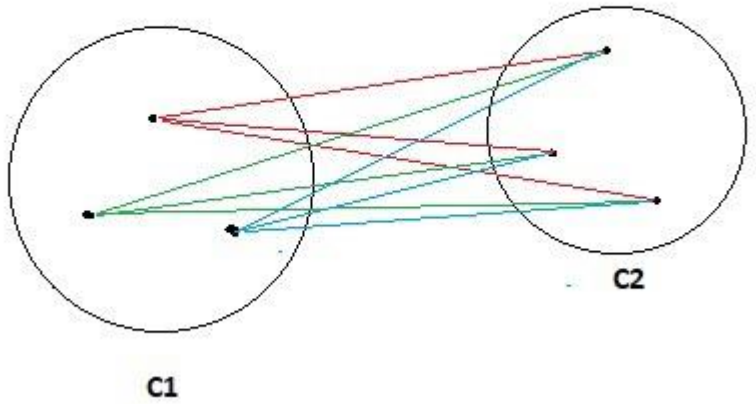



- Cons of Max:
- Max approach is biased
- Max approach tends to break large clusters.





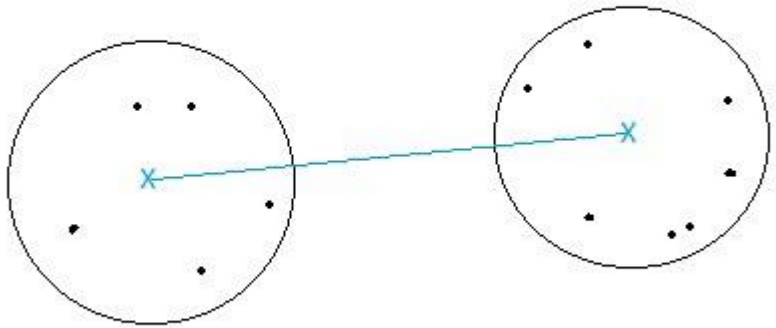
# Group Average

- Take all the pairs of points and compute their similarities and calculate the average of the similarities.
- Mathematically this can be written as,
- $sim(C1,C2) = \sum sim(Pi, Pj)/|C1|*|C2|$
- where, $Pi \in C1$ & $Pj \in C2$
- 

- The group Average approach does well in separating clusters if there is noise between clusters.
- Cons of Group Average:
- The group Average approach is biased towards globular clusters.
- Distance between centroids: Compute the centroids of two clusters C1 & C2 and take the similarity between the two centroids as the similarity between two clusters. This is a less popular technique in the real world.

# Ward's Method

- **Ward's Method:** This approach of calculating the similarity between two clusters is exactly the same as Group Average except that Ward's method calculates the sum of the square of the distances Pi and PJ.
- Mathematically this can be written as,
- $sim(C1,C2) = \sum (dist(Pi, Pj))^2 / |C1| * |C2|$



- **Pros of Ward’s method:**
- Ward’s method approach also does well in separating clusters if there is noise between clusters.
- **Cons of Ward’s method:**
- Ward’s method approach is also biased towards globular clusters.

- **Space complexity:** The space required for the Hierarchical clustering Technique is very high when the number of data points are high as we need to store the similarity matrix in the RAM. The space complexity is the order of the square of n.
- Space complexity = $O(n^2)$ where n is the number of data points.
- **Time complexity:** Since we've to perform n iterations and in each iteration, we need to update the similarity matrix and restore the matrix, the time complexity is also very high. The time complexity is the order of the cube of n.
- Time complexity = $O(n^3)$ where n is the number of data points.

# Example

create a proximity matrix

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

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

[Hierarchical Clustering | Hierarchical Clustering Python (analyticsvidhya.com)]

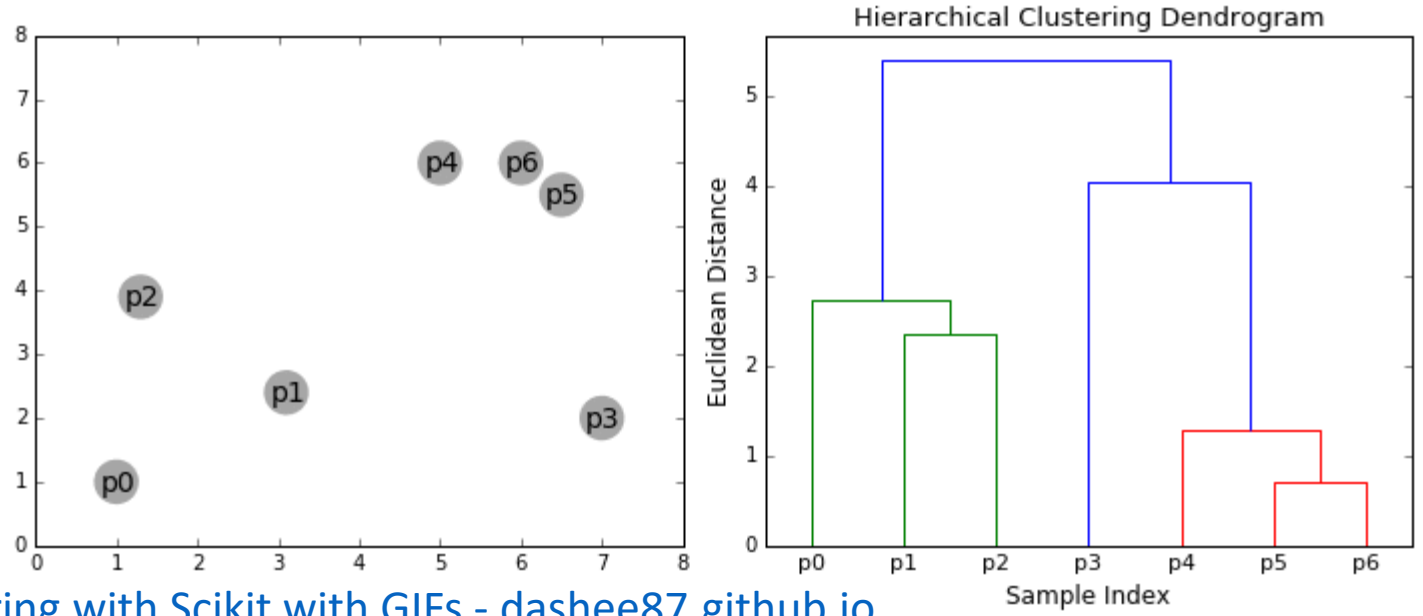
- Step 1: First, we assign all the points to an individual cluster:


- Next, we will look at the smallest distance in the proximity matrix and merge the points with the smallest distance. We then update the proximity matrix:

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

# What is a Dendrogram?

- A dendrogram is a tree-like diagram that records the sequences of merges or splits.
- Distance between data points represents dissimilarities.
- Height of the blocks represents the distance between clusters.




Clustering with Scikit with GIFs - dashee87.github.io