Section 5: Hierarchical Clustering

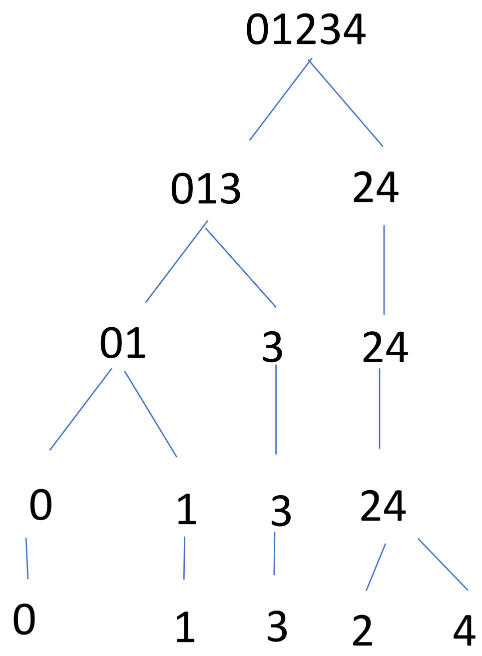
Section 5.1: Hierarchical Clustering: Algorithm

Connectivity Based Clustering

- Based on assumption that data points are more closely related to nearby data points than to far away data point
- Results depend on the distance metric used
- Bottom up: Agglomerative
- Top down: Divisive
- See Resources file for links to additional information

Hierarchical (Agglomerative) Clustering

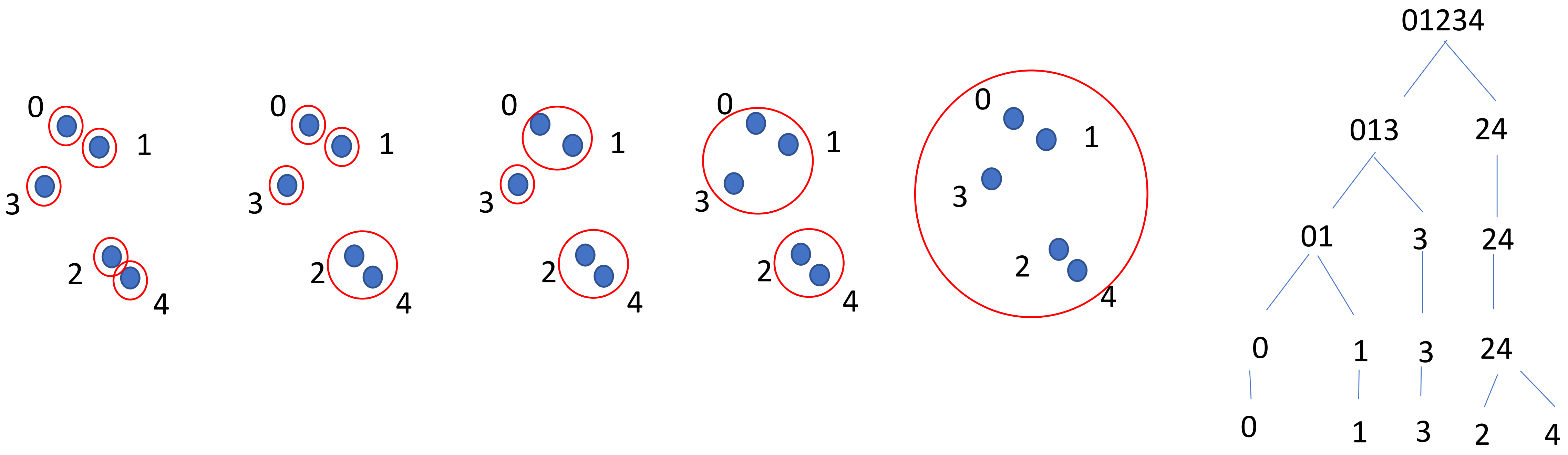
- Bottom up approach: each data point starts as its own cluster
- Nearby clusters are repeatedly combined until all points in single cluster
- Creates clusters at all levels
- Can be represented in a tree structure (dendogram)



Hierarchical Clustering - Algorithm

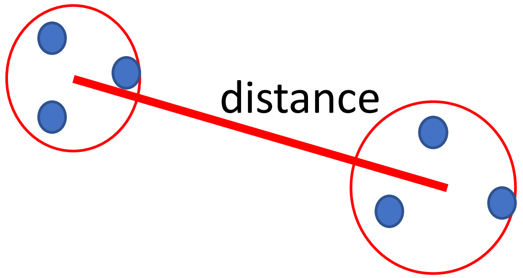
- Assume N data points and define each as a cluster

(1) Loop until there is a single cluster
  - Compute pairwise distances between each of the clusters
  - Combine clusters with the shortest pairwise distance into one cluster



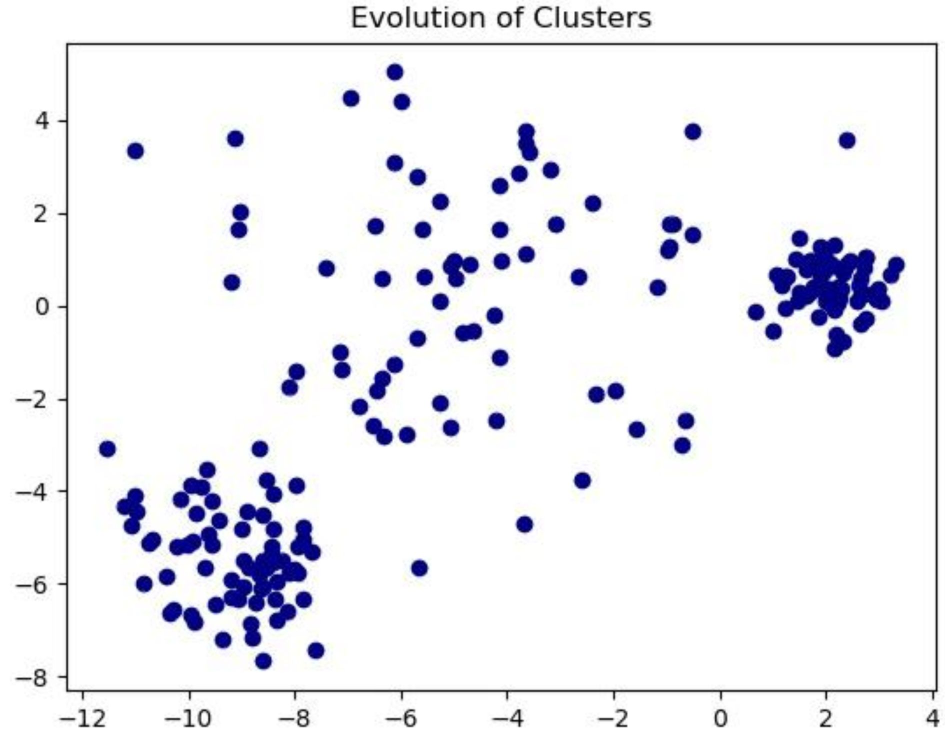
Distance between Clusters

- Can define distance between clusters in number of ways
- In this course, define distance between clusters as distance between means of clusters
- Can also define distance between clusters as minimum distance between any 2 points in clusters



Hierarchical Clustering - Example

- Dataset: sklearn varied_blobs1 dataset with 205 points
- Movie shows clustering at all levels to 3 clusters



Hierarchical Clustering: Complexity

Assume M data points and d dimensions

- If at level with m clusters need to compute $O(m^2)$ pairwise distances
- Recall that

$$\sum_{m=1}^{M} m^2 = O(M^3)\ as\ M \to \infty$$

- Number of operations for all levels is $O(M^3)\ \ as\ M \to \infty$
- Can get away with using memory that is $O(M)\ as\ M \to \infty$, as one can compute pairwise distances as needed and then discard
- Number of operations and memory required are both proportional to dimension d

Hierarchical Clustering: Notes

- Creates clusters of arbitrary shapes
- Clustering at all levels is unique if distances between clusters are unique
  - If dist(clusterA,clusterB) and dist(clusterC,clusterD) are smallest and same, then course code will combine first pair of clusters encountered with smallest distance
  - Can update code to combine clusters A and B and clusters C and D at same level
- User specifies distance measure
- Principal limitation: not feasible for large number of data points as number of operations is $O(M^3)\ as\ M \to \infty$

Section 4.2: Overview of Clustering Code Design

Clustering Code Design

- Create a design that can be used for:
  - Hierarchical Clustering
  - DBSCAN
  - K Means
  - Gaussian Mixture Model
- This section will discuss base class design
  - Key methods
  - Principal Variables
- If you would like to design code yourself, then stop video

Clustering Code Design: Basic Code Structure

- Each of the clustering algorithms uses an iterative approach for determining cluster assignment for each data point

(1) Make initial guess for cluster for each point

(2) Loop
   - Improve guess for cluster assignment for each data point
   - Stop when cluster assignments converge

clustering_base class: Principal Variables

Variable	Type	Description
self.X	2d numpy array	Contains the dataset Number of rows = number of dimensions for data Number of cols = number of data points Example: $\begin{bmatrix} 1 & 1.6 & -0.5 & -1.6 \\ 0.9 & 1.65 & -0.6 & -1.5 \end{bmatrix}$
self.clustersave	list of 1d numpy arrays	self.clusterave[i][j] is cluster assignment for iteration i, data point j Example: for 3 iterations: $[[-1 \quad -1 \quad -1 \quad -1], [0 \quad 0 \quad 2 \quad 1], [0 \quad 0 \quad 1 \quad 2]]$

clustering_base class – Key Methods

Method	Description
__init__	Initialize class and input relevant details for algorithm Example: K Means: number of clusters
initialize_algorithm	Initialize variables for the algorithm: Examples: Specify initial means for K Means algorithm Specify initial means, covariances, and weights for Gaussian Mixture Model
fit	The method uses iterative approach to determine cluster assignments at each iteration
get_index	Input: cluster_assignment, cluster Returns indices of data points in with cluster assignment = cluster
plot_objective	Input: title (string), xlabel (string), ylabel (string) Plot objective function with tracks progress of clustering if defined
plot_cluster	Input: nlevel (integer), title (string) ,xlabel (string),ylabel (string) Plot data points showing cluster assignments at a single iteration (clusters distinguished by color)
plot_cluster_animation	Input: nlevel (integer), interval (integer), title (string), xlabel (string), ylabel (string) Create animation showing data points and evolution of cluster assignments

Section 4.3: Hierarchical Clustering: Code Design

Hierarchical Clustering Code Design

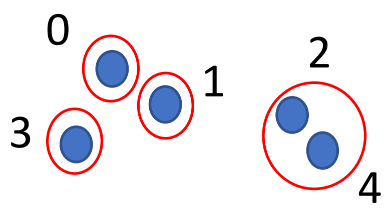
- This section presents design of the Hierarchical Clustering code
- Design is based on algorithm described in Section 5.1
- Stop video here, if you would like to do code design yourself

DBSCAN Code Design: To Do

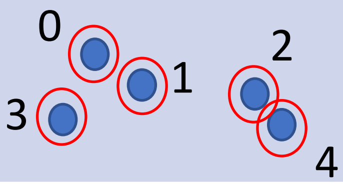
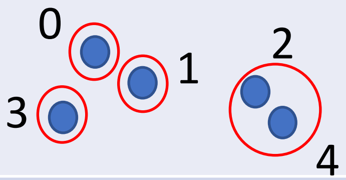
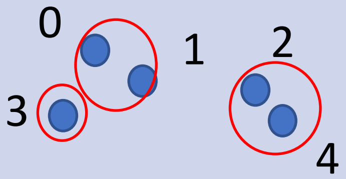
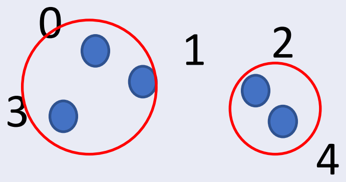
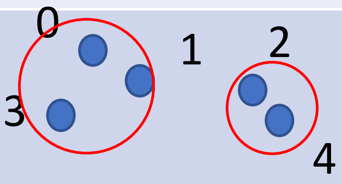
Component	Description
class hierarchical	class hierarhical derived from clustering_base
driver_hierarchical	driver for hierarchical

Hierarchical Clustering: list_cluster

- For hierarchical clustering introduce list_cluster variable that is more suitable for tracking clusters
- Variable list_cluster is a list of lists of clusters
  - Example list_cluster = [[0], [1], [2,4], [3], []]
  - Data point 0 is a cluster
  - Data point 1 is a cluster
  - Data points 2,4 are in a cluster
  - Data point 3 is in a cluster



Hierarchical Clustering: Example

Clusters	Description	
	Level 0: start with raw data points list_cluster: each data point is its own cluster clustersave[0]: set all labels to -1	list_cluster = [[0], [1], [2], [3], [4]] clustersave[0] = [-1,-1,-1,-1,-1]
	Level 1 Clusters [2] and [4] are closest so combine Assign label 2 to points 2 and 4 (smallest index value)	list_cluster = [[0], [1], [2,4], [3], []] clustersave[1] = [-1,-1,2,-1,2]
	Level 2 Clusters [0] and [1] are closest so combine Assign label 0 to points 0 and 1 (smallest index value)	list_cluster = [[0,1], [], [2,4], [3], []] clustersave[2] = [0,0,2,-1,2]
	Level 3: Clusters [0,1] and [3] are closest so append [3] to [0,1] (append cluster with smaller number of points to one with larger number of points) Assign label 0 to point 3 (smallest index value in cluster)	list_cluster = [[0,1,3],[],[2,4],[],[]] clustersave[3] = [0,0,2,0,2]
	Level 4: Combine final 2 clusters into single cluster Assign label 0 to points 2,4 (smallest index value in cluster)	list_cluster = [[0,1,3,2,4],[],[],[],[]] clustersave[4] = [0,0,0,0,0]

hierarchical class: Principal Variables

Variable	Type	Description
self.X	2d numpy array	Column j is the j’th data point
self.clustersave	list of 1d numpy arrays	self.clustersave[i][j] is the cluster assignment for data point j at level i of algorithm
self.list_cluster	list of lists	self.list_cluster[k] is the current list of data point indices for cluster k

hierarchical class – Key Methods

Method	Input	Description
initialize_algorithm		Initialize variables self.clustersave, self.list_cluster Return: nothing
fit	X (2d numpy array)	Performs hierarchical clustering algorithm Return: nothing
dist_between_clusters	idx1, idx2 (lists)	Determine the distance between cluster means, where clusters specified by indices in idx1 and idx2 Return: distance
combine_closest_ clusters		Combine closest clusters Return: nothing

Section 4.4: Hierarchical Clustering: Code Walkthrough

Hierarchical Clustering: Code Walkthrough

- Code is located in:

  Folder: UnsupervisedML/Code/Programs

  Files: hierarchical.py, driver_hierarchical.py
- Stop video here, if you would like to do coding yourself before seeing my implementation