Clustering P2

ENSF 444

Review what unsupervised learning is

Discuss how this can be applied to partition data

Clustering Models: Agglomerative

Hierarchical clustering that builds clusters from bottom up.

Works like this:

• Initially, each data point is assigned to its own cluster.
• Then, the two clusters that are closest to each other are merged into one.
• This process is repeated until the desired number of clusters is reached or another stopping criterion is met.

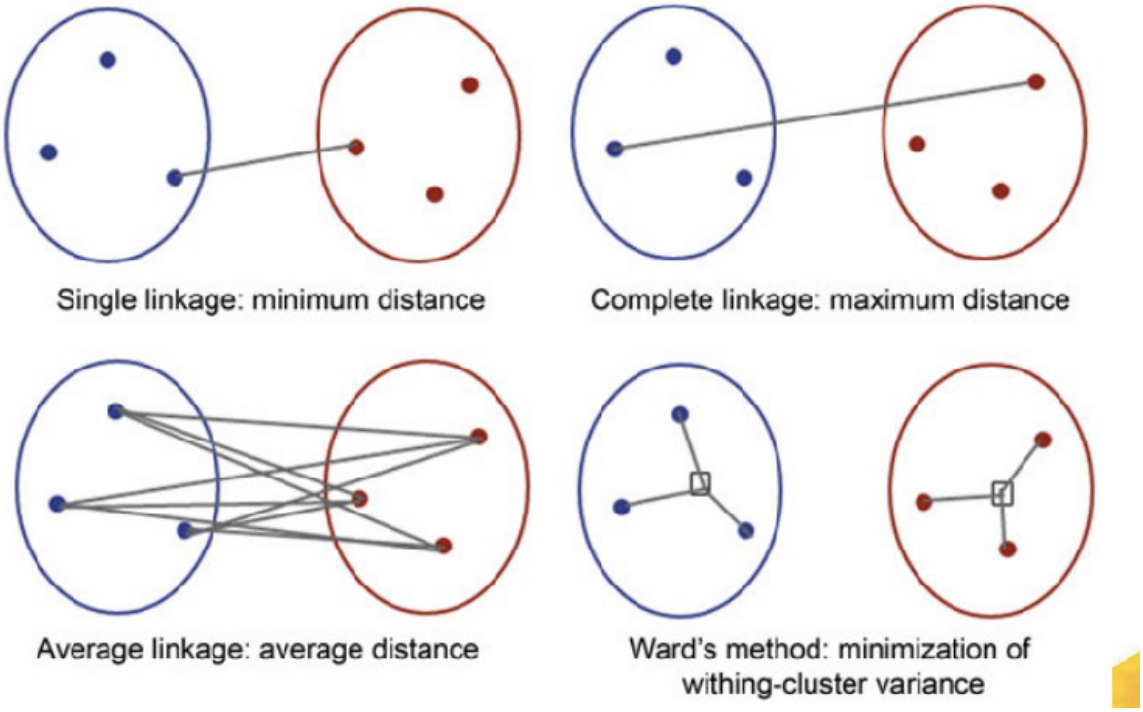
The most common stopping criterion is the number of clusters, which can be specified by the user. Scikit-learn implements this option in its agglomerative clustering module.

Linkage Criteria

Linkage criteria measure the similarity between two clusters and determine which ones to merge. There are four common options:

• **Ward**: This is the default option. It merges the two clusters that cause the least increase in variance within all clusters. This results in clusters of similar size.
• **Average**: This option calculates the average distance between all points in two clusters and merges the ones with the smallest value.
• **Complete**: This option (also called maximum linkage) finds the maximum distance between any two points in two clusters and merges the ones with the smallest value.
• **Single**: This option (also called minimum linkage) finds the minimum distance between any two points in two clusters and merges the ones with the smallest value.

Linkage Criteria (2/)



Ward works well for most datasets, but the other options might be better if the clusters have unequal sizes. For example, if one cluster is much larger than the rest.

Some of the hyperparameters for AgglomerativeClustering from the sklearn:

- **n_clusters:** The number of clusters to find. It must be an integer or None (if distance_threshold is not None).
- **metric:** The metric used to compute the linkage. Can be "euclidean", "l1", "l2", "manhattan", "cosine", or "precomputed". If the linkage is "ward", only "euclidean" is accepted.
- **linkage:** The linkage criterion to use, which determines the distance between sets of observations. Options are "ward", "complete", "average", or "single".
- **distance_threshold:** The linkage distance threshold at or above which clusters will not be merged.

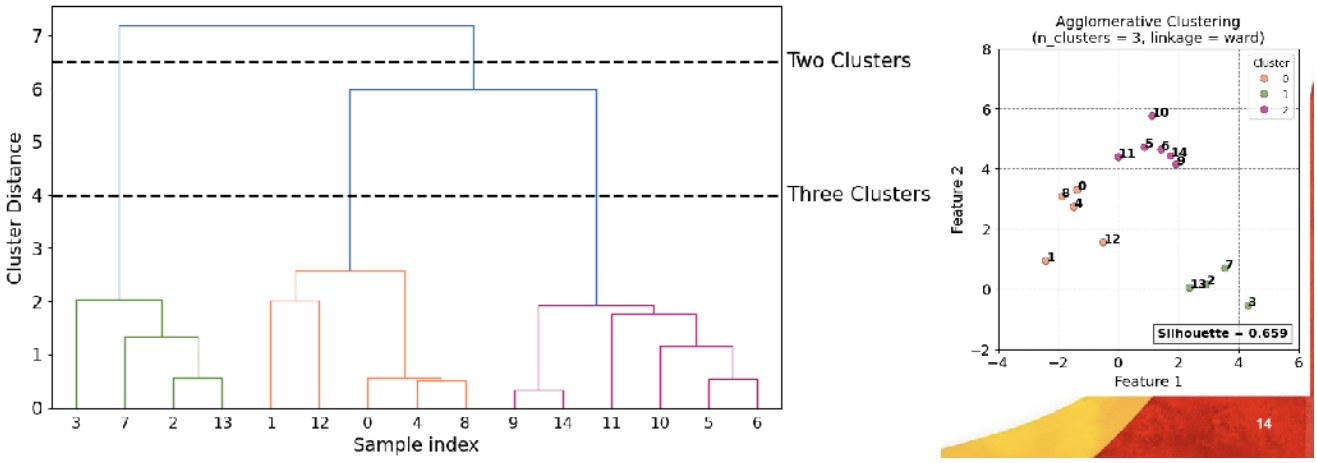
Hyper Parameters for AgglomerativeClustering above

Dendogram

A dendrogram is a tree-like diagram that shows how data points are grouped into clusters based on their similarity or distance. For example, suppose we have a synthetic two-dimensional data set with 15 points, as shown in the left plot below.

This is what the dendogram looks like

The dendrogram below shows how the clusters are formed and the distance between them. Each node represents a cluster, and the height of the node indicates the distance between the clusters that are merged. The lower the node, the more similar the clusters are.



Clustering Models: DBSCAN

DBSCAN is a clustering algorithm that stands for Density-Based Spatial Clustering of Applications with Noise. It has the following advantages:

• It does not require specifying the number of clusters in advance
• It can discover clusters with arbitrary shapes
• It can detect outliers or noise points that do not belong to any cluster

• DBSCAN is based on the concept of density in the feature space, which measures how close data points are to each other.
• DBSCAN identifies clusters as regions of high density, separated by regions of low density.
• DBSCAN is slower than agglomerative and k-means algorithms, but it can handle larger datasets and more complex data structures.

Hyperparameters

DBSCAN is a clustering algorithm that groups data points based on their **density**. It requires the user to specify two hyperparameters that control the density of a region:

- **min_samples**:
  - This is the minimum number of data points that must be within a certain distance to form a dense region.
  - A data point is considered a **core sample** (or core point) if it has at least min_samples neighbors, including itself, within a given radius.
- **eps**:
  - This is the radius that defines the distance between data points.
  - Two data points are considered neighbors if they are within eps distance of each other.
  - DBSCAN assigns the same cluster label to core samples that are neighbors, and to any other data point that is reachable from a core sample by a chain of neighbors.

DBSCAN identifies core samples as data points that belong to a dense region and clusters them based on their proximity. The choice of min_samples and eps can affect the number and size of the clusters that DBSCAN finds.

Methodology for DBSCAN

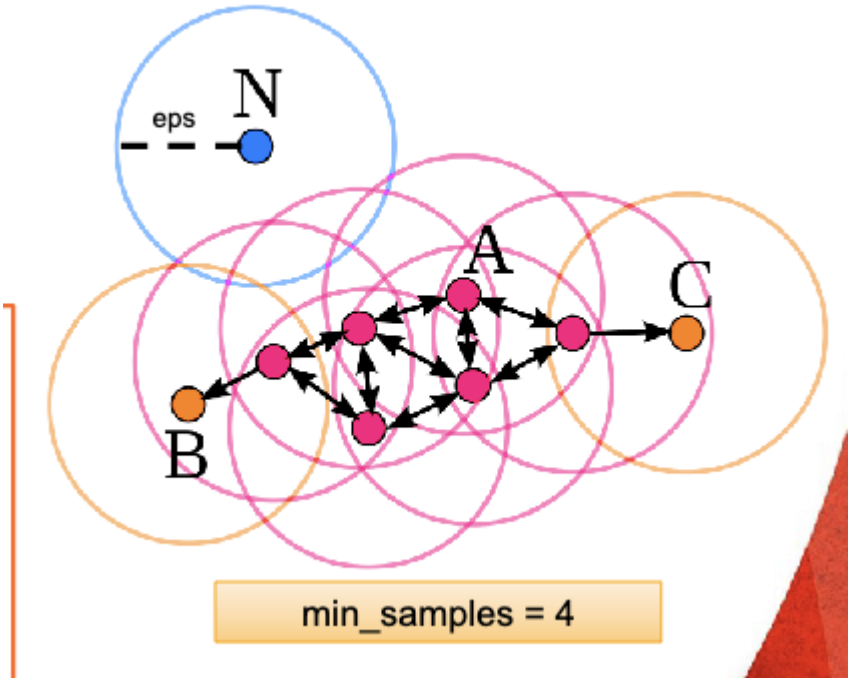
1. Randomly selects a data point that has not been visited yet
2. it checks how many data points are within the **eps** form the selected point. This is called the neighbourhood
3. If the neighbourhood has fewer than min_samples data points the selected point is marked as noise, means doesnt belong to any clusters
4. if the neighbourhood has at least min_samples data points, then selected point is marked as a core point and then assigned to a new cluster label. then algo visits all the data points in the neighbourhood and assigns them the same cluster label. If any of these data points are also core points, their neighborhoods are visited and labeled as well, and so on. This process continues until all the data points that are reachable from the selected point by a chain of neighbors are clustered.

   Steps for 4.
   1. If a point has enough nearby points (at least "min_samples"), it's called a "core point" and gets its own cluster label.
   2. Then, all the nearby points of that core point are also assigned to the same cluster label.

3. If any of these nearby points are also core points, their nearby points are visited and labeled as well.
4. This process keeps going until all points reachable from the original core point are grouped into the same cluster.

5. The algorithm repeats the above steps until all the data points have been visited and labeled

- Point **A** and the other pink points are **core points**, because the area surrounding these points in an *eps* radius contain at least 4 points (including the point itself).
- Points **B** and **C** are **not core points** but are reachable from A (via other core points) and thus belong to the cluster as well.
- Point **N** is a **noise point** that is neither a core point nor directly-reachable.



Types of points

1. **Core Points** - are data points that have at least min_samples data points within a distance of eps from them. They are the main drivers of the clustering process
2. **Boundary Points** - are data points that are within a distance of eps from a core point, but have fewer than min_samples data points in their neighborhood. They are assigned to the

same cluster as the nearest core point, but they do not expand the cluster.

3. **Noise Points** - are data points that are neither core nor boundary points. They are considered outliers and do not belong to any cluster.

- The clustering of the core points is deterministic, meaning that it does not depend on the order of the data points or the random selection of the starting point.
- The same core points will always be clustered together, and the same points will always be labeled as noise, regardless of how many times the DBSCAN algorithm is run on the same dataset.

Clustering Models: Gaussian Mixture Models (GMM)

K-means clustering is a popular method due to its simplicity and efficiency. However, it struggles with complex cluster shapes and structure because it assign points to the nearest cluster center, leading to spherical clusters of equal size.

This can be seen from the limitations with elongated, overlapping, or nested clusters.

To fix this, we use Gaussian Mixture Models. GMM is used for weird shapes making it better than K means when cluster shapes are weird.

- GMMs posit that data points are generated from a combination of multiple Gaussian distributions (aka normal distributions), each characterized by its mean, covariance, and mixture weight.
- The objective of GMMs is to infer these parameters and categorize each data point into the Gaussian distribution it most likely belongs to.
- Additionally, GMMs can quantify the uncertainty of cluster assignments by calculating the posterior probability for each Gaussian component given the data. This feature is particularly useful for pinpointing ambiguous data points or outliers.

How each shape is works

• "**spherical**": This option assumes that clusters are spherical, with equal variance in all directions—similar to the clusters formed by K-means. However, unlike K-means, GMMs with spherical covariance can have clusters of different sizes due to varying component weights.
• "**diag**": Here, clusters can stretch along the feature axes since each feature has its own

variance, but features are considered uncorrelated. This results in elongated, axis-aligned clusters.

• "**tied**": With a tied covariance matrix, all clusters share the same shape and orientation but can differ in size and position. This leads to elliptical clusters that have the same orientation across the feature space.

• "**full**": The most flexible option, where each cluster has its own covariance matrix, allowing for elliptical clusters of any orientation and shape. This flexibility comes at the cost of increased computational complexity due to the larger number of parameters to estimate.

What can these shapes look like?