Task

Unsupervised Learning – Hierarchical Clustering

Visit our website

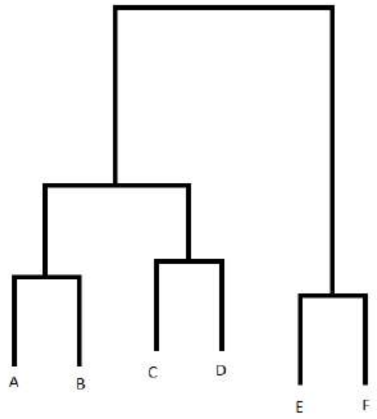
Introduction

WELCOME TO THE UNSUPERVISED LEARNING – HIERARCHICAL CLUSTERING TASK!

In this lesson, we will explore bottom-up or agglomerative clustering. Unlike K-means clustering, which requires you to specify the number of clusters (K), agglomerative clustering is a hierarchical method that starts with individual data points and merges them iteratively. This allows for a more exploratory approach to cluster identification.

AGGLOMERATIVE HIERARCHICAL CLUSTERING

Hierarchical clustering is an algorithm that builds, as the name suggests, a hierarchy of clusters, which form a tree-like structure called a "dendrogram", as in the image below.



Initially, each data point is assigned to a mini-cluster of its own (at the "leaves" of the dendrogram, A–F), and a measure of dissimilarity (such as Euclidean distance between their features) is defined. Using this dissimilarity metric, the algorithm consists of iteratively merging the most similar two clusters into one larger cluster, starting from the leaves and working upwards until all observations belong to a single cluster. The dissimilarity measure determines how far up the dendrogram the merge takes place, with a greater dissimilarity placing the merge higher up the dendrogram. This is called agglomerative clustering because the process is a bottom-up process, and it's called hierarchical because clusters are nested within each other. The dendrogram can then be "cut" at a particular height to obtain the clusters, which is discussed in greater detail later.

DISTANCE METRICS

Hierarchical clustering relies on measures of similarity or dissimilarity between data points to determine which clusters should be merged. Various distance metrics can be used, each with its own strengths and weaknesses. Common choices include:

- Euclidean distance: Measures the straight-line distance between two points in a Euclidean space. Suitable for continuous numerical data.
- Manhattan (city block) distance: Calculates the sum of absolute differences between corresponding coordinates. It is useful when the direction of differences matters.
- Cosine similarity: Measures the angle between two vectors. Suitable for comparing the orientation of data points in a high-dimensional space.

The choice of distance metric depends on the nature of the data and the specific goals of the analysis. Explore the `metric` parameter for the `AgglomerativeClustering` class in the **scikit-learn documentation** to find available distance metrics in scikit-learn.

DISSIMILARITY BETWEEN GROUPS

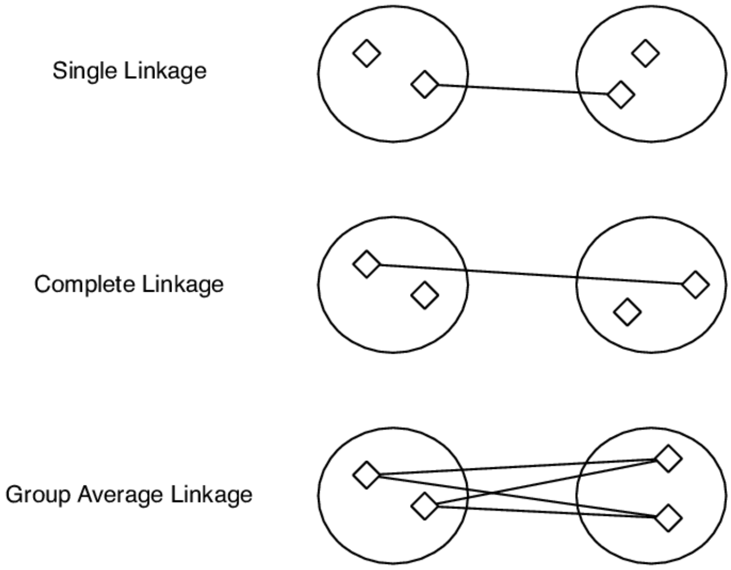
When each cluster contains only one sample, it's straightforward to calculate the distance between them using a simple measure like Euclidean distance. However, as the hierarchical clustering algorithm progresses, we start merging clusters that already contain multiple samples. This introduces a challenge: how do we measure the dissimilarity between two clusters, rather than just individual points?

A simple Euclidean distance between single points is no longer sufficient when comparing entire groups. Instead, we need a way to determine how far apart two clusters are. This is where the **linkage criterion** comes in. The linkage criterion defines how to calculate the distance between clusters, allowing us to merge them based on their overall similarity.

Think of it like this: If we have two clusters, one containing points A and B, and another containing points C and D, we need a method to quantify how different or similar these two groups of points are. The linkage criterion gives us several ways to measure the distance between these clusters.

The following are three simple linkage criteria:

1. **Single linkage**
   The dissimilarity metric is calculated between all samples in the one cluster with all samples in the other cluster, and the **smallest dissimilarity** is chosen. This is called a "pairwise" comparison.
2. **Complete linkage**
   The dissimilarity metric is calculated between all samples in the one cluster with all samples in the other cluster, and the **largest dissimilarity** is chosen.
3. **Average linkage**
   The dissimilarity metric is calculated between all samples in the one cluster with all samples in the other cluster, and the **average dissimilarity** is chosen.



Types of linkages (Cong et al., 2015)

Scikit-learn's hierarchical clustering method also comes with an implementation of **Ward's method**. This more sophisticated measure computes the sum of squares of all possible cluster merges and chooses the optimum merge accordingly.

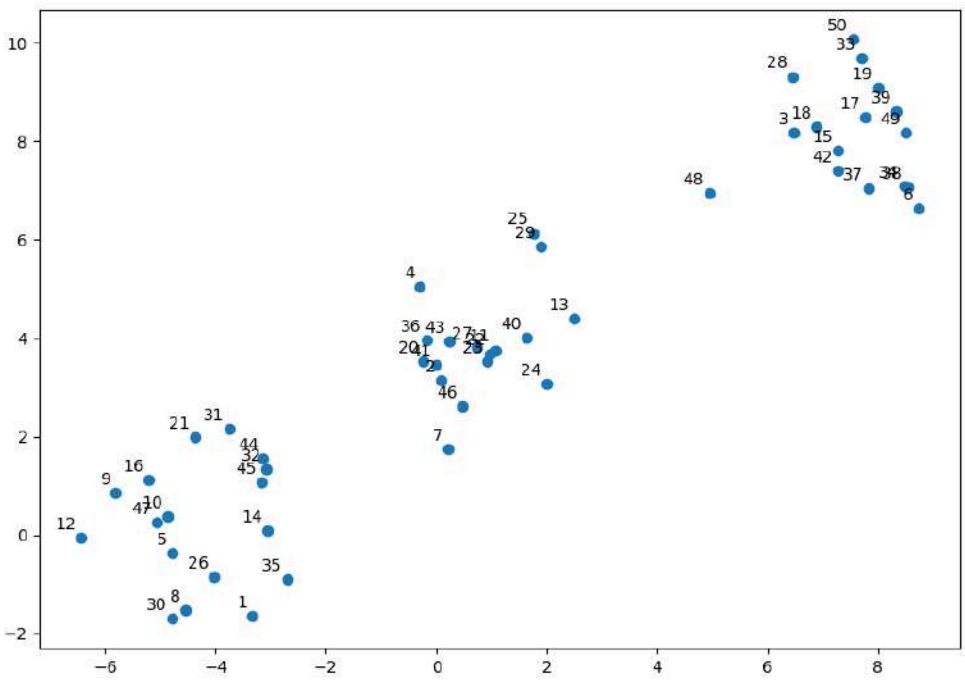
The choice of dissimilarity measure is very important, as it has a strong effect on the resulting dendrogram. Average, complete, and single linkage are most popular among statisticians. Average and complete linkages are generally preferred over single linkage, as they tend to yield more balanced dendrograms.

DENDROGRAMS

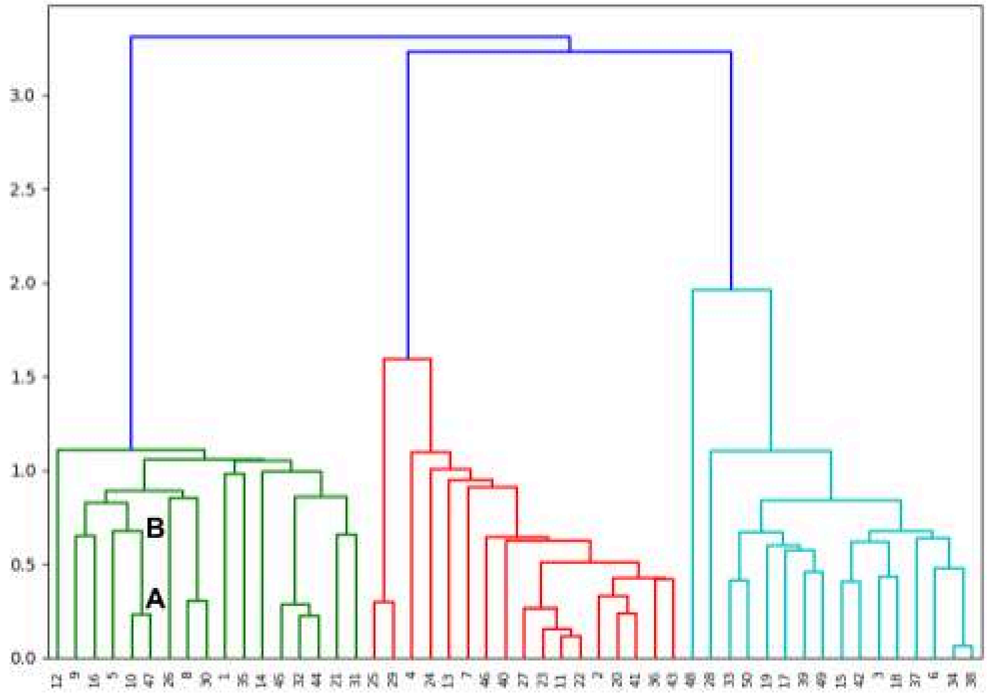
Agglomerative clustering will place all data points in a single final cluster unless the process is stopped early. This means we need to tell the algorithm when to stop, just like with K-means clustering. However, unlike with K-means clustering, we have an extra tool: a dendrogram that shows us what a range of clusters would look like.

A dendrogram contains two kinds of information. Firstly, it shows each merge that was performed. Secondly, the lengths of the vertical lines show the distance (Euclidean distance or whatever distance measure was specified) between the merged clusters. An important thing to note: larger distances between clusters mean that those clusters are more **dissimilar**.

In the scatter plot below, you can see three well-defined clusters. Logically speaking, we know that we're looking for three clusters in the dendrogram.

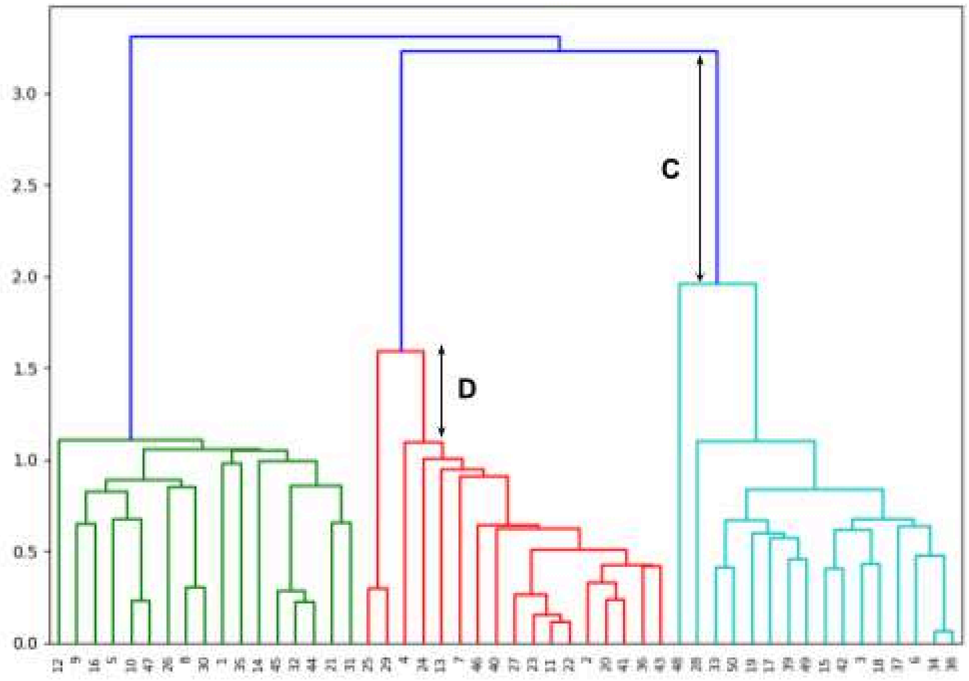


In the following example, data points 10 and 47 are merged and have very short vertical lines (A), which means they are similar. Pairs (9, 16) and (5, (10, 47)) are fairly dissimilar from each other internally, but their averages are fairly similar, so their merge has fairly short lines.



We can use the dendrogram to identify a clustering in which all clusters have low within-cluster variance by determining which merges would merge highly dissimilar clusters. Merges between dissimilar clusters (clusters that are further away from each other) look like lines which stretch out high and then merge at the top. To cut the process off before such clusters can be merged, we specify a minimum distance between clusters.

A good minimum distance can be determined by finding the section of the y-axis that corresponds to the largest section of the graph with no horizontal line in it. For illustration, see sections C and D in the image below, of which C is clearly the largest.



Recall that we are looking for the most *dissimilar* clusters. The larger the distance between clusters, the more dissimilarity there is. With the y-axis on the dendrogram representing distance, it follows that the blue lines represent our desired clusters, and because there are three blue lines, it makes sense that there are three clusters.

Instructions

Read the Jupyter Notebook in this task's folder before attempting the auto-graded task.

Take note:

The task below is **auto-graded**. An auto-graded task still counts towards your progression and graduation.

Give it your best attempt and submit it when you are ready.

When you select "Request Review", the task is automatically complete, and you do not need to wait for it to be reviewed by a mentor.

You will then receive an email with a link to a model answer, as well as an overview of the approach taken to reach this answer.

Take some time to review and compare your work against the model answer. This exercise will help solidify your understanding and provide an opportunity for reflection on how to apply these concepts in future projects.

In the same email, you will also receive a link to a survey, which you can use to self-assess your submission.

Once you've done that, feel free to progress to the next task.

Auto-graded task

- Create a Jupyter notebook called **h_clustering_task.ipynb**.
- Load the Iris dataset. Select two features from the data to use in this exercise and scale the data.
  i. Note that we will not use the target variable (species) in this task as this is an unsupervised learning approach.
- Using single and complete linkages, and Euclidean and city block distance metrics, print dendrograms for the different combinations of these. You should have four dendrograms.

- Pick one dendrogram to go forward with.

    i. Choose a fixed number of clusters based on the dendrogram of your choice. These should be the most dissimilar clusters.
    ii. Run agglomerative hierarchical clustering with that number of clusters (and the linkage method and distance metric used for that dendrogram).
    iii. Verify the clusters you obtained by using the silhouette score and comment on your confidence in your clustering solution.

Important: Be sure to upload all files required for the task submission inside your task folder and then click "Request review" on your dashboard.

Share your thoughts

Please take some time to complete this short feedback **form** to help us ensure we provide you with the best possible learning experience.

REFERENCES

Cong, Z., Fernandez, A., Billhardt, H., & Lujak, M. (2014). Service discovery acceleration with hierarchical clustering. *Information Systems Frontiers*, *17*(4), 799–808. **https://doi.org/10.1007/s10796-014-9525-2**