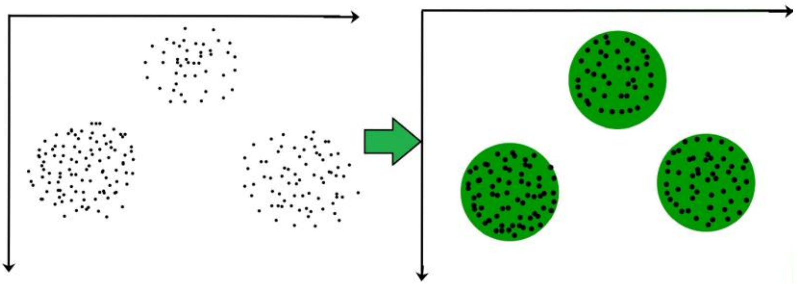
WEEK 10: DATA CLUSTERING-K-means

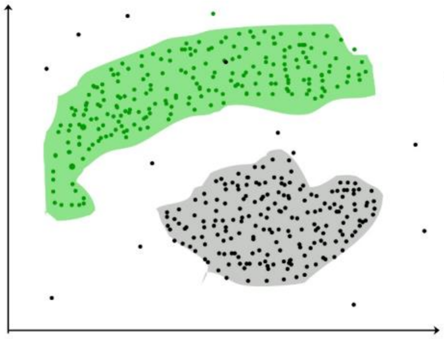
Introduction to Clustering: It is basically a type of *unsupervised learning method*. An unsupervised learning method is a method in which we draw references from datasets consisting of input data without labeled responses. Generally, it is used as a process to find meaningful structure, explanatory underlying processes, generative features, and groupings inherent in a set of examples.

Clustering is the task of dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group and dissimilar to the data points in other groups. It is basically a collection of objects on the basis of similarity and dissimilarity between them.

For example The data points in the graph below clustered together can be classified into one single group. We can distinguish the clusters, and we can identify that there are 3 clusters in the below picture.



It is not necessary for clusters to be spherical as depicted below:



DBSCAN: Density-based Spatial Clustering of Applications with Noise

These data points are clustered by using the basic concept that the data point lies within the given constraint from the cluster center. Various distance methods and techniques are used for the calculation of the outliers.

Why Clustering?

Clustering is very much important as it determines the intrinsic grouping among the unlabelled data present. There are no criteria for good clustering. It depends on the user, and what criteria they may use which satisfy their need. For instance, we could be interested in finding representatives for homogeneous groups (data reduction), finding "natural clusters" and describing their unknown properties ("natural" data types), in finding useful and suitable groupings ("useful" data classes) or in finding unusual data objects (outlier detection). This algorithm must make some assumptions that constitute the similarity of points and each assumption make different and equally valid clusters.

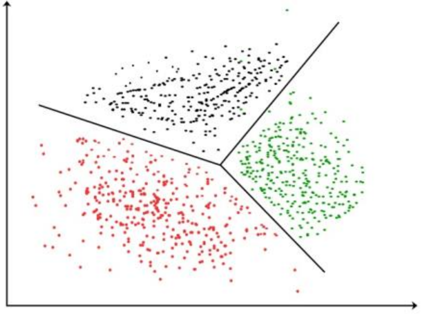
Clustering Methods:

- **Density-Based Methods:** These methods consider the clusters as the dense region having some similarities and differences from the lower dense region of the space. These methods have good accuracy and the ability to merge two clusters. Example *DBSCAN (Density-Based Spatial Clustering of Applications with Noise)*, *OPTICS (Ordering Points to Identify Clustering Structure)*, etc.
- **Hierarchical Based Methods:** The clusters formed in this method form a tree-type structure based on the hierarchy. New clusters are formed using the previously formed one. It is divided into two category
    - **Agglomerative** (bottom-up *approach*)
    - **Divisive** (top-down *approach*)

Examples *CURE (Clustering Using Representatives)*, *BIRCH (Balanced Iterative Reducing Clustering and using Hierarchies)*, etc.

- **Partitioning Methods:** These methods partition the objects into k clusters and each partition forms one cluster. This method is used to optimize an objective criterion similarity function such as when the distance is a major parameter example *K-means, CLARANS (Clustering Large Applications based upon Randomized Search)*, etc.
- **Grid-based Methods:** In this method, the data space is formulated into a finite number of cells that form a grid-like structure. All the clustering operations done on these grids are fast and independent of the number of data objects example *STING (Statistical Information Grid), wave cluster, CLIQUE (CLustering In Quest)*, etc.

Clustering Algorithms: K-means clustering algorithm – It is the simplest unsupervised learning algorithm that solves clustering problem.K-means algorithm partitions n observations into k clusters where each observation belongs to the cluster with the nearest mean serving as a prototype of the cluster.

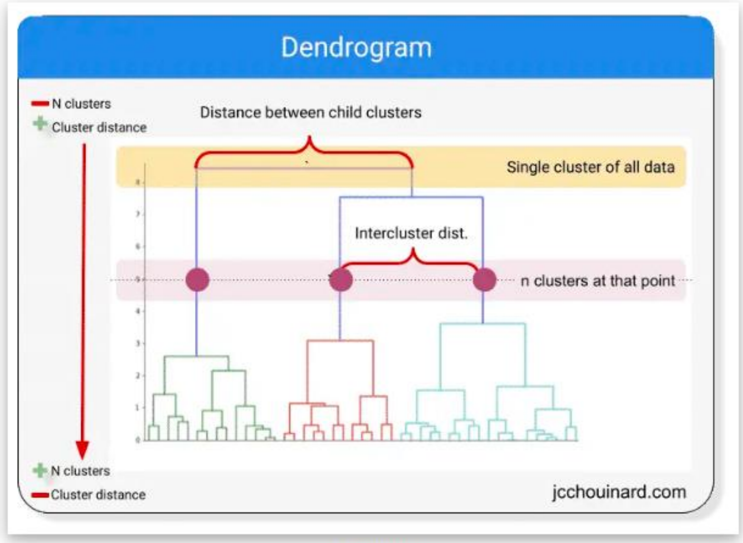


Hierarchical Clustering

Hierarchical clustering algorithm works by starting with 1 cluster per data point and merging the clusters together untilthe optimal clustering is met.

1. Having 1 cluster for each data point
2. Defining new cluster centers using the mean of X and Y coordinates
3. Combining clusters centers closest to each other
4. Finding new cluster centers based on the mean
5. Repeating until optimal number of clusters is met

The image below represents a **dendrogram** that can be used to visualize hierarchical clustering. Starting with 1 cluster per data point at the bottom and merging the closest clusters at each iteration, ending up with a single cluster for the entire dataset.



Dendrogram

Some examples of hierarchical clustering algorithms are:

- hirearchy from SciPy’s scipy.cluster

Hierarchical Clustering in Python Example

```
import matplotlib.pyplot as plt
import numpy as np
from numpy.random import rand
import pandas as pd
import seaborn as sns
```

```
from scipy.cluster.vq import whiten
from scipy.cluster.hierarchy import fcluster, linkage

# Generate initial data
data = np.vstack(((rand(30,2)+1), (rand(30,2)+2.5),  (rand(30,2)+4)  ))

# standardize (normalize) the features
data = whiten(data)

# Compute the distance matrix
matrix = linkage(  data,  method='ward',  metric='euclidean'  )

# Assign cluster labels
labels = fcluster(  matrix, 3, criterion='maxclust'  )
# Create DataFrame
df = pd.DataFrame(data, columns=['x','y'])
df['labels'] = labels

# Plot Clusters
sns.scatterplot(  x='x',  y='y', hue='labels',  data=df )
plt.title('Hierachical Clustering with SciPy')
plt.show()
```

How To Make Clustering in Machine Learning

To cluster data in Scikit-Learn using Python, you must process the data, train multiple classification algorithms and evaluate each model to find the classification algorithm that is the best predictor for your data

1. **Load data**

You can load any labelled dataset that you want to predict on. For instance, you can use fetch_openml('mnist_784') on the Mnist dataset to practice.

2. **Explore the dataset**

Use python pandas functions such as df.describe() and df.isnull().sum() to find how your data need to be processed prior training

3. **Preprocess data**

Drop, fill or impute missing, or unwanted values from your dataset to make sure that you don't introduce errors or bias into your data. Use pandas get_dummies(), drop(), and fillna() functions alongside some sklearn's libraries such as SimpleImputer or OneHotEncoder to preprocess your data.

4. **Split data into training and testing dataset**

To be able to evaluate the accuracy of your models, split your data into training and testing sets using sklearn's train_test_split. This will allow to train your data on the training set and predict and evaluate on the testing set.

5. **Create a pipeline to train multiple clustering algorithms and hyper-parameters**

Run multiple algorithms, and for each algorithm, try various hyper-parameters. This will allow to find the best performing model and the best parameters for that model. Use GridSearchCV() and Pipeline to help you with these tasks

6. **Evaluate the machine learning model**

Evaluate the model on its precision with methods such as the homogeneity_score() and completeness_score() and evaluate elements such as the confusion_matrix() in Scikit-learn

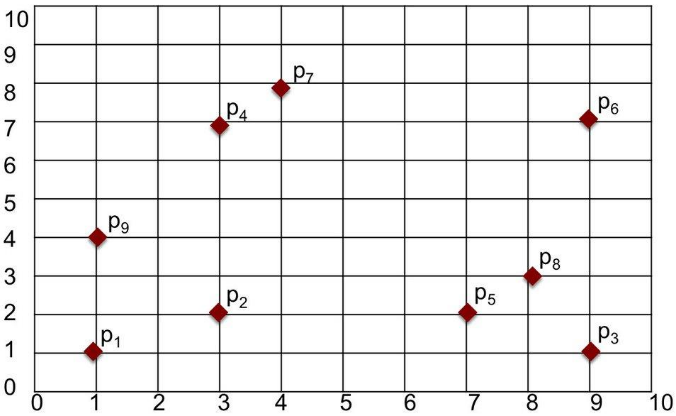
Reference

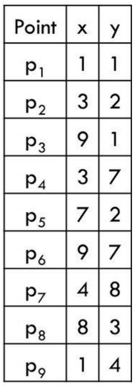
https://www.geeksforgeeks.org/ml-types-of-linkages-in-clustering/

refer: Choosing the right linkage method for hierarchical clustering
https://stats.stackexchange.com/questions/195446/choosing-the-right-linkage-method-for-hierarchical-clustering
https://statsandr.com/blog/clustering-analysis-k-means-and-hierarchical-clustering-by-hand-and-in-r/#data-1
https://online.stat.psu.edu/stat505/lesson/14/14.4

Questions

1. Consider the following data set and apply the hierarchical data-clustering algorithm, to identify the clusters. Solve it manually by considering all linkage functions (Single, Complete, Average, Centroid, and Ward) using Euclidean distance.





Point	x	y
p_1	1	1
p_2	3	2
p_3	9	1
p_4	3	7
p_5	7	2
p_6	9	7
p_7	4	8
p_8	8	3
p_9	1	4

2. Consider the above-mentioned data set in Q no 1 and apply the hierarchical data-clustering algorithm, to identify the clusters. Write a Python function (without using the **scikit-learn library**) to do the following:
a. Plot a graph that displays the number of clusters on the x-axis and the Sum of Squared Errors (SSE) on the y-axis.
b. Display the proximity matrix using Euclidean distance, Manhattan distance, and Minkowski distance.
c. Plot the dendrogram for single, complete, average, centroid, and ward linkage methods.

Additional Questions

Consider the above-mentioned data set in Q no 1 and apply the hierarchical data-clustering algorithm, to identify the clusters. Write a Python function (with using the **scikit-learn library**) to do the following:
a. Plot a graph that displays the number of clusters on the x-axis and the Sum of Squared Errors (SSE) on the y-axis.
b. Display the proximity matrix using Euclidean distance, Manhattan distance, and Minkowski distance.
c. Plot the dendrogram for single, complete, average, centroid, and ward linkage methods.