Project Report: Clustering Algorithms

Mahdis Rahmani

1 Introduction

Clustering is a fundamental unsupervised learning technique that aims to group similar data points into clusters. This project explores and applies several clustering algorithms to synthetic datasets and evaluates their performance using different metrics. The project demonstrates how clustering works on distinct datasets with varied shapes and structures, providing insights into the effectiveness of each algorithm.

2 Datasets Used

Three synthetic datasets were generated using `sklearn.datasets`:

- **Blobs**: A dataset with well-separated clusters of points, generated using the `make_blobs` function.
- **Circles**: Data arranged in concentric circles, generated using the `make_circles` function.
- **Moons**: A dataset with two crescent-shaped clusters, created using the `make_moons` function.

These datasets were chosen because they challenge the clustering algorithms to work on both simple (blobs) and complex (circles and moons) structures.

3 Clustering Algorithms

3.1 K-Means Clustering

K-Means is a widely used partition-based clustering algorithm that aims to minimize the variance within each cluster. The process involves:

(a) Randomly initializing centroids.

(b) Assigning data points to the nearest centroid.

(c) Updating centroids iteratively until convergence.

K-Means was applied to the blobs, circles, and moons datasets. For simple datasets like blobs, K-Means performed well, but it struggled with the more complex shapes, such as circles and moons, due to its assumption of spherical clusters.

3.2 Agglomerative Clustering

Agglomerative Clustering is a hierarchical approach where each data point starts as its own cluster, and clusters are merged based on proximity until a desired number of clusters is achieved. The linkage method used in this project was *ward*, which minimizes the variance between clusters.

Agglomerative Clustering performed well on all datasets, particularly with complex shapes like circles and moons, where K-Means struggled.

3.3 DBSCAN

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a density-based algorithm that clusters data points based on regions of high density. It is effective for datasets with arbitrary shapes and can identify noise points.

DBSCAN outperformed K-Means and Agglomerative Clustering for the circles and moons datasets, successfully identifying the non-spherical shapes. It also effectively handled noise, making it highly robust.

4 Evaluation Metrics

Two key metrics were used to evaluate the performance of the clustering algorithms:

- **Rand Index**: Measures the similarity between the true and predicted labels by considering all pairs of points and determining whether they are consistently assigned to the same or different clusters.
- **Jaccard Index**: Measures the similarity between the true and predicted labels by comparing the number of elements in common between the sets of true and predicted clusters relative to the total number of elements.

4.1 Results Summary

- **Blobs Dataset**: K-Means and Agglomerative Clustering performed well due to the distinct, well-separated nature of the blobs. DBSCAN also worked but was less effective due to the clear separation of clusters.
- **Circles and Moons Datasets**: K-Means struggled to capture the complex structure of these datasets, while Agglomerative Clustering and DBSCAN performed significantly better. DBSCAN, in particular, excelled at identifying the circular and crescent shapes in the data.

5 Visualizations

Scatter plots were used to visualize the clusters formed by each algorithm. Different colors were used to represent the clusters, revealing how well each algorithm adapted to the structure of the data. For instance, DBSCAN showed clear separation of complex shapes in the circles and moons datasets, while K-Means often resulted in overlapping clusters for these datasets.

6 Conclusion

This project highlights the strengths and limitations of various clustering algorithms:

- **K-Means**: Suitable for simple, well-separated data but struggles with complex, non-spherical clusters.
- **Agglomerative Clustering**: More flexible and performs better for datasets with hierarchical structures or clusters of varying sizes.
- **DBSCAN**: Excels at identifying clusters of arbitrary shapes and is robust to noise, making it the most effective for datasets like circles and moons.

The best overall algorithm for this project was **DBSCAN**, due to its ability to handle arbitrary shapes and noise. However, for simpler, well-separated clusters (like the blobs dataset), **K-Means** performed more efficiently. **Agglomerative Clustering** offered a balance, excelling in cases with hierarchical structures or varying cluster sizes.