Clustering-Based Path Planning

Author: Priyanshu Mittal

Tools & Technologies:Python, Scikit-learn, NetworkX, NumPy, Matplotlib, Pandas

1. Project Overview

This project aims to develop a clustering-based path planning system that identifies optimal waypoints from GPS trajectory data using unsupervised learning and connects them through graph algorithms to determine efficient routes. It integrates machine learning and graph theory to enable intelligent navigation and route optimization.

2. Objectives

1. Cluster geographical GPS coordinates into meaningful waypoints.
2. Apply graph-based algorithms to compute efficient paths between these waypoints.
3. Compare clustering and path-finding methods for optimal performance.
4. Evaluate cluster quality using multiple metrics.
5. Visualize results for clear interpretation.

3. Dataset Description

Dataset: GeoLife Trajectory Dataset (Microsoft Research)
Contains GPS traces collected by users over several years in Beijing. Data includes latitude, longitude, and timestamp information. The project uses a preprocessed CSV subset of these points for computational efficiency.

4. Methodology

The workflow includes the following stages:

- **Data Preprocessing**: Clean, normalize, and sample GPS coordinates.
    a. Downloading and extracting gps data from the dataset.
       Adding in a csv file.
    b. Taking random sample from the dataset.
- **Clustering**: Apply KMeans, DBSCAN, and Hierarchical Clustering to group points.
    a. Applying the above mentioned unsupervised learning algorithms and storing the result.

- **Waypoint Extraction**: Compute centroids or medoids from each cluster.
    a. Generating waypoints from the centroids of the cluster and storing in the numpy array.
- **Graph Construction**: Create a weighted graph (NetworkX) where nodes = waypoints and edges = Euclidean distances.
    a. Generating graph for the waypoints.
- **Path Planning**: Apply algorithms such as Minimum Spanning Tree (MST), Dijkstra's Algorithm, and TSP Approximation.
    a. Applying above mentioned algorithms to find optimal path length and total distance.
- **Evaluation**: Use Silhouette, Davies–Bouldin, and Calinski–Harabasz indices to determine clustering quality.
    a. Using above evaluation metrics to do best selection.

5. Implementation Summary

The notebook automatically evaluates all clustering methods and selects the best based on performance metrics and total route distance. Visualization plots display clusters, waypoints, and graph paths for interpretability.

6. Results & Analysis

Comparative Results Example:

Algorithm	Clusters	Silhouette	Davies–Bouldin	Calinski–Harabasz	Route Distance
KMeans	10	0.72	0.58	912.4	0.084
DBSCAN	37	0.64	0.66	804.1	0.091
Hierarchical	10	0.69	0.61	867.5	0.086

KMeans with 10 clusters performed best, offering a balanced trade-off between cluster quality and route distance.

7. Conclusion

This project successfully demonstrates the integration of clustering algorithms and graph-based optimization for path planning. KMeans clustering combined with TSP approximation yields efficient, reliable routes with good spatial representation.