UVSQ université PARIS-SACLAY | UFR des Sciences CAMPUS DE VERSAILLES

Jan-26

AMDM Project

Analyzing Human Activity and Mobility

Karine Zeitouni

Instructions:

Prepare a zip file that includes a detailed summary report containing the queries and screenshots of the results, along with the Jupyter notebook for the Python section. A presentation and a live demonstration will be scheduled.

Tools and libraries:

Postgres/PostGIS, MobilityDB, Python libraries: pandas, geopandas, movingPandas, sqlalchemy, shapely, hvplot,…

Part 1

In this scenario, we consider a GPS track from a single user collected continuously over a week. We provide 2 CSV files: the GPS log file and a self-report that reflects changes in the user's place or activity.

Tasks:

1. Use Python to preprocess the GPS log by loading the CSV file into a DataFrame.
2. Segment the data by day.
3. Utilize the MovingPandas library to create trajectories for each day and visualize them.
4. Clean the trajectories and visually check the results.
5. Identify stops (visits) and movements (travels) within the data. Further segment the trajectories into stops and moves.
6. Rank the "move segments" based on the total distance travelled, and then rank them by duration.
7. Verify whether these results align with the self-reported changes. Highlight any matches and discrepancies.
8. Propose a method to identify probable locations for "Home" and "Work."
9. Calculate the average daily time the user spent at home, as well as the average time spent at work.
10. Use MovingPandas (to_mf_json) to export the segmented trajectories in MF-JSON format.

Part 2

This part is related to trajectory analytics and employs a public dataset of multimodal trips in Beijing. Some trips have been annotated by the transport modes. We selected here annotated trips from one user. Here are the steps. You can find the details and the data formats in the user guide (see User Guide-1.3.pdf).

1. Load data from the trajectory file (20080815074715.plt) into a PostgreSQL table, let's call it *pg_table*. Note that the header (6 lines) should be ignored. Add the Geometry attribute. Then, use the Geometry Viewer of pgAdmin to visualize the data points.
2. Load this trajectory in a moving object table (*mdb_table*) using MobilityDB. Compute basic statistics (length, duration, mean speed, etc.).
3. Parse the "trajectory" folder to load the other data files into the PostgreSQL table *pg_table* (it's recommended to add an attribute to keep track of the source file). Then, add the Geometry to *pg_table*. Check visually.
4. Add these trajectories to the MobilityDB table *mdb_table* created above.
5. Calculate the total distance travelled, the total duration, and the average speed; then, refine these statistics per day.
6. Now, we will study the alignment between the trajectories and the labels. To do this, load the label file into a simple table, *lbl_table*. Provide a new version of moving object table (*mdb_table2)* annotated with the labels and segmented according to the time intervals in *lbl_table*.
7. Calculate a statistical summary for each trajectory in *mdb_table2*: report the label, the distance, the duration, and the speed (min, max, average, median, variance, standard deviation).
8. Use any statistical or machine learning technique to analyse de consistency between the label and the computed speed.