Spatial Data Science & Engineering

Project Phase 2

Maximum points Possible – 7

Spatial Data Visualization:

Deck.gl (https://deck.gl/) is a spatial and temporal data visualization library having lots of fancy visualization patterns (https://deck.gl/examples). The original library is built on JavaScript, and basic API for using the library is also on JavaScript (https://deck.gl/docs). However, this library also has a Python wrapper, named pydeck (https://pydeck.gl/installation.html). Among the visualization patterns available on deck.gl, a popular pattern used for visualizing the movement of moving object trajectories is the Trips layer (https://deck.gl/examples/trips-layer/).

Tasks:

In Phase-1, you wrote scripts on Apache Sedona for supporting the following functionalities of moving object trajectories: 1) loading trajectory datasets, 2) answering spatial range query, 3) answering spatiotemporal range query, and 4) answering KNN query. In Phase-2, you will design a front-end interface which will have options to take input for the 4 tasks performed in Phase-1 and visualize the movement of the output queries using the Trips layer (deck.gl or pydeck). You will also need to write an API which will communicate and transfer input/outputs between the front-end interface and the Phase-1 project running on Apache Sedona. Your tasks can be summarized as follows:

1. Design a front-end interface hosted on your local lost. You can use any language which you can connect with either deck.gl or pydeck. The interface will have at least 5 options: 1) option 1 to show the trips layer from deck.gl or pydeck, 2) option 2 to take input for loading trajectory dataset (option to browse a path from your local computer), 3) option 3 to take input for spatial range query (only the range of latitude and longitude), 4) option 4 to take input for spatiotemporal range query (only ranges of latitude, longitude, and timestamp), and 5) option 5 to take input for KNN query (trajectory id and k).

2. You will keep the Phase-1 project running on Apache Sedona on your computer. You can modify the method parameters in Phase-1 as necessary. Write an API to connect the running Phase-1 project to the front-end interface. When you provide input with the front-end interface, the API will send the inputs to the Phase-1 project. The output trajectories returned by the Phase-1 project will be send back to the front-end interface to display the trajectories using the trips layer. When the input is loading a trajectory dataset, display all the trajectories in the dataset using the trips layer. You can use either deck.gl or pydeck visualization library.
3. Optional: try to make the interface flexible for providing the inputs. For example, in order to take the spatial range, you may provide a rectangular tool to select the range from the map. For the temporal range, you may provide a horizontal bar to select the starting and ending timestamp on the bar. These are optional. Use whatever input options you are flexible with. The look of the interface is completely up to you. Try to make the input options flexible for the input provider as much as you can.

Demo:

There will be a demo session on November 28th in the class during class time. Each group will come with a laptop by setting up the full Phase-2 project. The group will run it on local host. Instructors will give input and verify the output visualization. The groups will be graded based on the demo.

Submission Instructions:

Submit the full project (Apache Sedona code, front-end code, API code) as a zip file by November 27th 11:59 PM.