TrajDash: GPS Trajectory and Mobility Analysis Dashboard (Demo Paper)

Amirhossein Sefati
University of Calgary
Calgary, Canada
Email: amirhossein.sefati@ucalgary.ca

John Krumm
University of Southern California
Los Angeles, USA
Email: jkrumm@usc.edu

Cyrus Shahabi
University of Southern California
Los Angeles, USA
Email: shahabi@usc.edu

Abstract—**Mobility trajectories are important for many different research and analysis tasks. We present TrajDash, the first online dashboard that gives access to multiple mobility datasets and enables users to query, analyze, visualize, upload, and download trajectories. Our dashboard makes it convenient to find relevant trajectory data, summarize its characteristics and statistics, download the data, upload data, and create maps and charts for presentation. In this paper, we explain the functions and implementation of our trajectory dashboard.**

1. Introduction

Human mobility trajectories are timestamped sequences of latitude/longitude that describe where people go. The data often originates from GPS receivers on individuals' cell phones. These trajectories are increasingly important for understanding movement patterns, analyzing traffic, predicting future locations, siting new facilities, finding the underserved, and detecting anomalous behavior. There are public [6], [7] and private [9] trajectory repositories that researchers, developers, and policymakers use to support these tasks.

Assessing and using trajectory data can be cumbersome, because there is no common storage location and no centralized tools for visualizing the data and computing statistics. As an example of this problem, a potential user of trajectory data may seek trajectories from a certain time period and a certain spatial region with minimum requirements on the temporal and spatial sampling gaps. As it is, this user would have to tediously discover, download, and analyze different trajectory datasets to find any that are suitable.

We solve this problem with an on-line trajectory dashboard that lets users explore multiple trajectory datasets with relevant queries, statistics, and visualizations. The dashboard supports downloading and uploading trajectories.

2. Related Work

While there are no other online trajectory dashboards, there are two examples of systems that overlap with the functionality of ours.

The University of California, Riverside, Spatio-Temporal Active Repository (UCR Star) is an online database and visualization tool for geospatial data [1]. It allows users to view, explore, download, and upload tables with arbitray columns. UCR Star presents summary statistics for each dataset, such as file size and number of rows. It uses indices for efficient queries and visualization. Among the currently 254 datasets, it contains a handful of public trajectory datasets. It is not focused on trajectories, however, and features no special functions aimed at analyzing and visualizing trajectories.

ArcGIS, a commercial product, can represent trajectories as trajectory datasets [4], and it provides methods for visualizing trajectories on a map or in a space-time cube [2], clustering [3], and interpolation for raster display [5]. As an extensible platform, ArcGIS could be programmed to reproduce the functionality of our dashboard, but we are aware of no publicly released implementations.

3. Dashboard Functionality

TrajDash supports several functions for trajectory data. The dashboard supports multiple data sources, such as different research datasets with timestamped latitude/longitude trajectories. Each data source is referred to by name, and each entity (person, animal, vehicle, etc.) in a data source is referred to by an ID.

3.1. Data Query and Download

Users can select trajectory data from a combination of different sources, geographic areas, and time periods. If the user specifies multiple criteria, e.g. a source inside a given rectangle over a certain time period, the system will return data that satisfies all the given criteria. The default output for a data query is a table displayed on the site with options for sorting by any column. The user has the option of downloading any table as a comma-separated (CSV) text file.

The user interface for data selection is shown in Figure 1. The query options are:

Select Source A user can select the source of the trajectory data from a dropdown list. The system currently supports three sources: T-Drive [6], GeoLife [7], and CabSpotting [8]. If the source is unspecified, the system will return data from all the sources.

Figure 1. Data can be selected in a variety of ways from the user interface. The results of a query can be downloaded as a CSV file.

Select User Each source has its data separated by the ID of the person or vehicle it came from, and a user can specify a particular ID for which to retrieve data.

Select Geographic Region Geographic regions can be specified by interactively drawing a rectangle on the map, giving the explicit latitude/longitude coordinates of a bounding box, or selecting a U.S. state. The result set will be limited to data points within the specified polygon.

Custom Query The dashboard provides the option of writing a custom SQL query for advanced extraction of data.

Download The results of any query can be downloaded as a CSV file.

3.2. Data Upload

Users can upload their own data to the platform by following these steps: (1) specify the owner of the dataset; (2) choose a name for the dataset and provide a reference link, if applicable; and (3) upload the CSV file containing all trajectory information. The CSV file should include the following specific columns:

Username Each point in the trajectory file should have a username, which specifies the user who reported the point.

Geom Using the longitude and latitude of the reported point, a geom column should be present for each point, specifying the actual coordinates of the reported point.

Timestamp Written in Unix seconds format, this column specifies the exact time that the point was reported.

Metadata Each point can have text metadata to provide additional information about that specific record. For example, if the points are reported for a taxi, it can include the occupancy of the taxi at that time and location.

Once the data is uploaded, TrajDash will process it to convert it into a format compatible with the platform. After data processing is complete, the specified dataset name will appear as an option when selecting the source. Users can then apply all supported tools and features to the newly uploaded data, enabling them to analyze and visualize their dataset.

3.3. Analysis

The dashboard supports features for analyzing trajectories interactively. This can help a user understand the trajectories and select which data is appropriate for more study.

Central Location For any ID in the datasets, the dashboard can show the median location on the map, which is the median of the ID's latitude and median of the ID's longitude. This is helpful for understanding the central location of each ID.

Histograms A user can get a histogram for any ID. The histogram can show the distribution of temporal and distance gaps between temporally adjacent points. It is displayed on the site along with options for modifying the bin width and size of the plot. The histograms can be downloaded as image files to be included in a report. An example histogram is shown in Fig. 2.

Outliers An outlier location point is defined as one where the speed to arrive at that point was too high. For each user, we store computed speed percentiles of 95, 98, 99, 99.5 and 99.9. The user can specify which percentile to use, and the query will show which points are outliers.

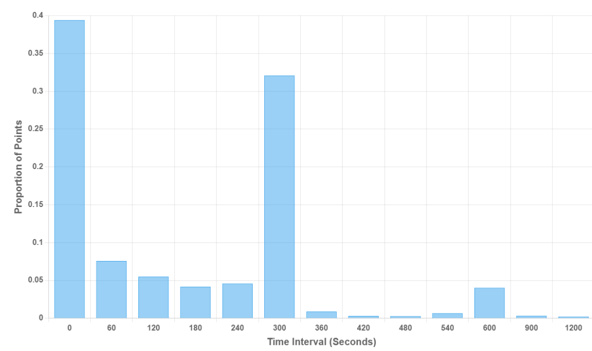

Figure 2. The dashboard can create a histogram of the temporal gaps between location points.

3.4. Visualizations

Most of the dashboard's visualizations are shown on an interactive map and a table as the result of different queries or analyses. Both the map and the table can be downloaded as a PNG and CSV file respectively.

Points The default visualization from a query is to show the location points as markers on a map.

Trajectory The user can optionally specify that the points on the map be connected by line segments in time order to show the trajectory of user(s) based on the filters provided in the query. An example is shown in Fig. 3.

Heat Map Instead of individual points, the user can see a heat map of all the points in a query showing the varying density of points in the specified region as shown in Figure 4.

Table Query results are always available to view as an interactive table that allows sorting by any column and downloading to a CSV file.

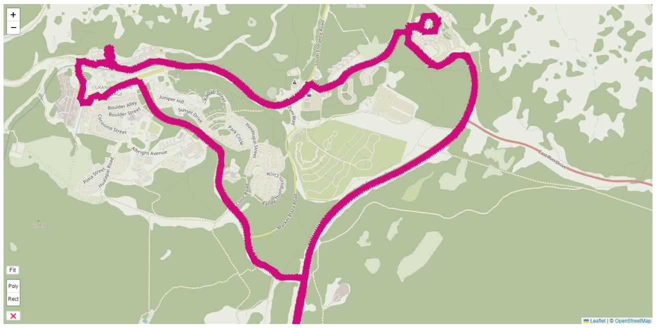

Figure 3. Trajectory map showing how user has moved based on a query.

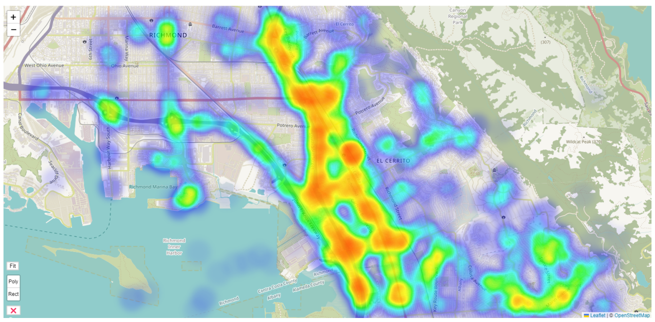

Figure 4. Heat map showing density of location points based on a query.

4. TrajDash Architecture

4.1. Overview

TrajDash has a modular, layered architecture, as illustrated in Figure 5. The modularity promotes flexibility and maintainability with modern software elements. The architecture consists of six layers: (1) the Data Collection Layer for acquiring data, (2) the Data Management Layer for data storage, conversion to planar spatial data types, and queries, (3) the Service Layer for API integrations, and (4) the Presentation Layer for data visualization, user interactions, and data uploads and downloads. The following section provides a detailed description of each layer of the TrajDash architecture.

Figure 5. TrajDash Layered Architecture

4.2. Data Collection Layer

This layer is responsible for gathering different trajectory datasets and mapping them to TrajDash standard data format, supporting the data upload feature. It addresses two major challenges: trajectories have various data shapes and formats which adds to the complexities of integrating them automatically to TrajDash. We implemented a code base that utilizes different SQL practices to meet this challenge.

Trajectory datasets are often large in size. In the TrajDash data schema, there are some attributes, e.g., different speed percentiles, that should be calculated when uploading the data. We addressed this challenge by implementing micro-services that run in the background and we only notify the user after the computations and insertions are complete. The features of the PostGIS extension (as discussed in Section 4.3) on top of PostgreSQL is essential to address these challenges. The general flow of upload dataset is shown in Figure 6.

In addition to AWS cloud, we use Atlas services which provide several out-of-the-box solutions as a MongoDB cloud provider such as replica sets for higher availability, horizontal and vertical scaling for scalability, and IP whitelisting and data encryption for higher security. Moreover, for higher performance and security, we can use disaggregated datacenters when they become practical [10].

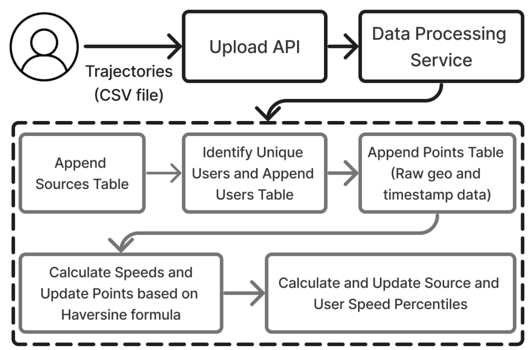

Figure 6. Upload dataset flow in TrajDash

4.3. Data Management Layer

The Data Management Layer serves as a bridge for integrating the data acquired by the Data Collection Layer to the TrajDash data format, which is essential for the entire application to be efficient and effective. As the data provided to the system can be uploaded by the authenticated users, we used MongoDB [11] for user-authentication and authorization process while using PostgreSQL [12] for conversion, storing, and querying the trajectories. Using different databases is essential to keep user's information secure and isolated from trajectory data.

Also discussed in [18], GIS-enabled PostgreSQL outperforms SQL Server and MySQL, two alternative databases for storing spatial data due to its rich built-in features. Users generate a JSON Web Token (JWT) which is used to authenticate each user when uploading data, the Service Layer (as discussed in Section 4.4) checks user authorization using the data stored in MongoDB, and then it allows data to be uploaded into PostgreSQL. In the background, data will be processed by activating another micro-service to integrate the uploaded data into the TrajDash data format. Figure 6 shows the flow of converting a user's uploaded data to TrajDash format and computing actual speeds and percentiles from the timestamped location points.

4.4. Service Layer

This layer is utilizing Node.js [13] along with Express.js [14] on top of it. This setup creates a non-blocking architecture for the endpoints by utilizing an event queue and worker thread pool. The Service Layer is responsible for creating a bridge between the Presentation Layer and the Data Management Layer to call the queries and analysis and send back information needed for tables, visualizations, and data uploads and downloads.

4.5. Presentation Layer

The Presentation Layer provides the user interface of TrajDash using React.js [15] with Openstreetmap (OSM) [16] and Leaflet [17] modules for mapping. It enables users to analyze, perform queries, and visualize trajectory datasets in an interactive environment. Additionally, users can download the results of analysis in both PNG and CSV formats. Also, Leaflet along with OSM is chosen as an open-source library which provides access to an interactive map while having rich features including being mobile-friendly. This layer also enables users to upload their datasets in an enhanced user-friendly way.

5. Demonstration

This section previews a demonstration of the TrajDash platform, highlighting its capabilities for trajectory analysis. The demonstration will include live interactions showcasing key features such as querying trajectory data, visualizing results in tables, plots, and maps, as well as database import and export functionalities. The following videos illustrate these features:

Analysis and Query : To demonstrate querying and analyzing trajectory data: https://youtu.be/w0a_zVZfL9E.

Import Dataset : To demonstrate importing trajectory dataset in preparation for analysis: https://youtu.be/dSCnlBqyj_w.

6. Conclusion

We have presented TrajDash, an online dashboard for mobility trajectories. It is designed for users to explore trajectory datasets with queries and visualisations. Users can download trajectory maps, histograms, and the actual trajectory data. TrajDash also allows users to upload their own data for exploration and sharing. The dashboard was built with modern web tools for presentation, services, authentication, data management, hosting, and cloud. While there are other services for exploring and downloading spatial data, TrajDash is the first one designed specifically for trajectories.

References

[1] S. Ghosh, T. Vu, M. A. Eskandari, and A. Eldawy, "UCR-STAR: The UCR Spatio-Temporal Active Repository," *SIGSPATIAL Special*, vol. 11, no. 2, pp. 34–40, Dec. 2019, doi: 10.1145/3377000.3377005.

[2] A. Buckley, "Working with Temporal Data in ArcGIS." 2018. [Online]. Available: https://www.esri.com/about/newsroom/arcuser/working-with-temporal-data-in-arcgis/. [Accessed: 24-Apr-2024].

[3] C.-C. Huang, "Detect Spatial-Temporal Point Clusters by Incorporating Time into Density-based Clustering." 2021. [Online]. Available: https://www.esri.com/arcgis-blog/products/arcgis-pro/analytics/detect-spatial-temporal-point-clusters-by-incorporating-time-into-density-based-clustering/. [Accessed: 24-Apr-2024].

[4] ESRI Corporation, "Create Trajectory Dataset (Data Management)." [Online]. Available: https://pro.arcgis.com/en/pro-app/latest/tool-reference/data-management/create-trajectory-dataset.htm. [Accessed: 24-Apr-2024].

[5] ESRI Corporation, "Work with trajectory data." [Online]. Available: https://pro.arcgis.com/en/pro-app/latest/help/data/imagery/work-with-trajectory-data.htm. [Accessed: 24-Apr-2024].

[6] Y. Zheng, "T-Drive trajectory data sample." Aug. 2011. [Online]. Available: https://www.microsoft.com/en-us/research/publication/t-drive-trajectory-data-sample/. [Accessed: 24-Apr-2024].

[7] Y. Zheng, H. Fu, X. Xie, W.-Y. Ma, and Q. Li, "Geolife GPS trajectory dataset - User Guide." Jul. 2011. [Online]. Available: https://www.microsoft.com/en-us/research/publication/geolife-gps-trajectory-dataset-user-guide/. [Accessed: 24-Apr-2024].

[8] M. Piorkowski, N. Sarafijanovic-Djukic, and M. Grossglauser, "CRAWDAD epfl/mobility," IEEE Dataport, 2022, doi: 10.15783/C7J010.

[9] "Veraset." [Online]. Available: https://www.veraset.com/. [Accessed: 05-Jun-2024].

[10] A. Sefati, M. Dolati, and M. Ghaderi, "Workload Placement with Bounded Slowdown in Disaggregated Datacenters," in *Proc. 19th Int. Conf. Network and Service Management (CNSM)*, 2023, pp. 1–9, doi: 10.23919/CNSM59352.2023.10327871.

[11] "MongoDB: The Developer Data Platform." [Online]. Available: https://www.mongodb.com/. [Accessed: 05-Jun-2024].

[12] "PostgreSQL: The world's most advanced open source database." [Online]. Available: https://www.postgresql.org/. [Accessed: 05-Jun-2024].

[13] Node.js Foundation, "Node.js." [Online]. Available: https://nodejs.org/. [Accessed: 05-Jun-2024].

[14] Express.js Contributors, "Express.js." [Online]. Available: https://expressjs.com/. [Accessed: 05-Jun-2024].

[15] React Contributors, "React." [Online]. Available: https://react.dev/. [Accessed: 05-Jun-2024].

[16] OpenStreetMap Contributors, "OpenStreetMap." [Online]. Available: https://www.openstreetmap.org/. [Accessed: 05-Jun-2024].

[17] Leaflet Contributors, "Leaflet.js." [Online]. Available: https://leafletjs.com/. [Accessed: 05-Jun-2024].

[18] Boston Geographic Information Systems, "Cross Compare SQL Server 2008 Spatial, PostgreSQL/PostGIS 1.3-1.4, MySQL 5-6." [Online]. Available: https://www.bostongis.com/content_name=sqlserver2008_postgis_mysql_compare. [Accessed: 06-Jun-2024].