TrajAnalytics: A software system for visual analysis of urban trajectory data

Ye Zhao
*Computer Science, Kent State University*
Xinyue Ye
*Geography, Kent State University*
Jing Yang
*Computer Science, University of North Carolina - Charlotte*
Urban Trajectory Data

• Advanced sensing technologies and computing infrastructures produce massive trajectory data
  • GPS, Wi-Fi, Cellular, RFID, blogs, tweets, etc.
  • Taxis, fleets, public transits, human, etc.

• Massive trajectory data “sample” city transportation and human mobility patterns
  • Origin/Destination, Positions, speed, occupancy, fare, direction, latitude, etc.
  • Big, dynamic and complex spatiotemporal data

• Taxis generate about 20 percent of traffic flow on road surfaces of Beijing China
Urban Study with Trajectory Data

• Decipher the information hidden in the trajectories of large populations
• Optimize assessment and planning of transportation infrastructures and policies
• Improve life quality and environment
Trajectory Data Analysis

• Apply computing techniques and tools such as statistics and machine learning to big trajectory data
  • Extracting functional regions and borders
  • Detecting problems of transportation network
  • Recommending optimal routes
  • Analyzing refueling behavior and drivers’ income
  • More …
Trajectory Data Visualization

• Support data exploration and analytical reasoning with interactive visual interfaces
  • Visual representations
  • Interaction techniques
TrajAnalytics Software: Goal

• To support researchers and analysts in transportation studies to conduct data driven analysis utilizing real-world trajectory data

• A publicly available and easy-to-use visual analytics software system

• Visual analytics software helps users extract deep insights with
  • Iterative, evolving information foraging and sense making
  • Interactive process using domain knowledge
TrajAnalytics Software: Features

• Integrate scalable data management and interactive visualization with powerful computational capability
  • Powerful computing platform
  • Easy access gateway
  • Scalable data storage and management
  • Exploratory visualization
TrajAnalytics Software Design

• TrajBase: manage big data over distributed platform
• TrajQuery: quickly answer data queries with parallel computing
• TrajVis: interactive visualization interface for exploratory data analysis and sharing
  • Web-based over internet access
TrajAnalytics Software Framework

- TrajBase: Data Management
- Urban Trajectories
- TrajQuery: Data Access
  - Region/Point Queries
  - Trajectory Queries
  - Aggregation Queries
- TrajVis: Visualization Interface
  - Map Visualization
  - Information Visualization

Diagram showing the integration of TrajBase, TrajQuery, and TrajVis components.
TrajBase Data Management

- Utilize multiple types of database systems
  - Relational database with geospatial indexing
  - NoSQL database with geospatial indexing
  - Graph database with geospatial indexing
- Support distributed multi-node platforms
  - Local clusters
  - Clouds
- Enable data exploration with web browsers
TrajQuery Data Access

• Region query
  • Query trajectories and/or GPS points given a region

• Trajectory query
  • Query trajectories and/or GPS points given trajectory attributes

• Aggregation query
  • Query aggregated information (e.g., average speed) over given regions and trajectories with specific time periods
TrajVis: Map Visualizations

• Point visualization
  • O/D locations, trajectory paths

• Heat maps
  • Density of GIS Points (e.g. Kernel Density Estimation)
  • Attributes (e.g. speed)

• 3D Trajectories by adding time axis
  • Space-time path
  • Trajectory walls
TrajVis: Information Visualizations

- Plots, graphs and charts of aggregated data
  - Speed or travel time
  - Count of points, taxis, trajectories
  - Fare, passenger number

- Specific visual metaphors for domain specific data computed from trajectory data
  - Geographical attributes, categories, etc.
  - Topics, cluster, outliers
Software Engineering Process and Licensing

• Development procedure
  • Data collection: publicly available datasets
  • Requirement analysis
  • Function design
  • Demo release
  • Software evaluation
  • Software release and maintenance

• BSD license with minimal restrictions on the redistribution