UCLA Samueli School of Engineering

LA Hillside Road Demand Analysis

Junwon Choi, *B.S. in Data Theory (University of California, Los Angeles, 2024)*
Sensors and Robotics for Infrastructure Lab (SRI Lab)

Objectives

<u>Main:</u> Provide an alternative method of **estimating road demand in the hillside street segments of the LA transportation system** using graph data and the LEHD Origin-Destination Employment Statistics (LODES) dataset from the U.S. Census Bureau.

<u>Additional:</u> Visualize traffic estimates based on demographic data.

Packages/Models Used (Python)

Spatial Data Packages

- ❑ GeoPandas: geospatial data handling
- ❑ Shapely: geometry (block) object handling + manipulation
- ❑ Rtree: spatial indexing
- ❑ NetworkX: shortest path algorithm implementation
- ❑ SciKit-Gstat: geostatistics models (Variogram, OrdinaryKriging)

Key Models

- ❑ Dijkstra Algorithm: shortest path computation [NetworkX - single_source_dijkstra_path]
- ❑ Kriging Interpolation: weight interpolation for unidentified points [SciKit-Gstat - Variogram + OrdinaryKriging]

Datasets

- Block shapefile: geocode + polygons of blocks in LA
- Population Origin-Destination: number of jobs/workers based on home and work Census Block
- Graph representation of LA (hillside): Node: Coordinates of road (intersection) points
  Edge: Roads (node connections)

Step 1: Block Determination

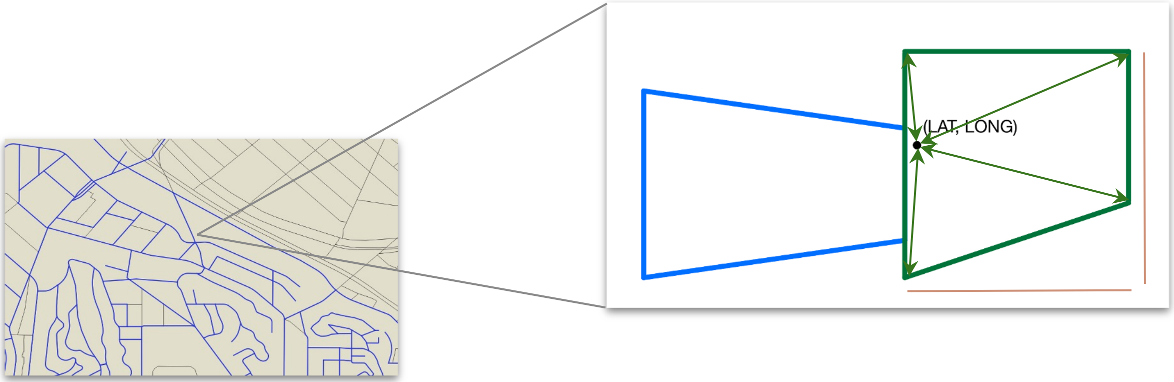
Task: Identify which block each node belongs to.

Approach: Check node against block boundaries.

→ Node must be contained within the 4 block boundary coordinates.

Challenge: Computationally demanding.

Solution: Utilize R-tree indexing for efficient finding of intersecting search rectangles.



Step 2: Node Origin-Destination Data Pairing

Task: Combine results from Step 1 with the Origin-Destination dataset to append job/worker quantity for each node.

Approach: Series of indexing, merging, expanding rows to produce merged dataframe.

Challenge: Duplication of block usage.

Solution: Create (distributed) adjustment columns to account for block usage duplication.

Origin-Destination

Block geocode (work)	Block geocode (home)	S000	…	SI03
⋮	⋮	⋮		⋮
060371352052007	60014001001003	21	…	3
⋮	⋮	⋮		⋮

\+

Step 1 Results

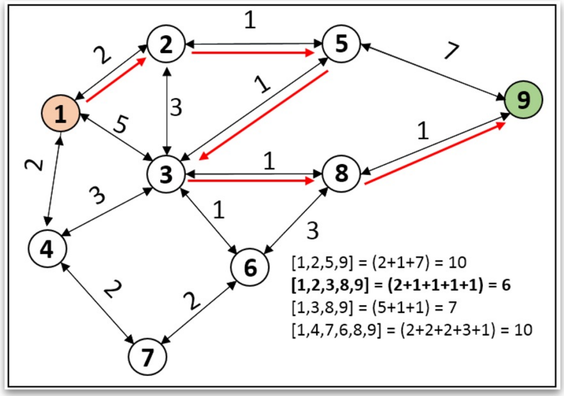
Node #	Block geocode	Block polygon
⋮	⋮	⋮
123	060371352052007	<POLYGON ((-118.668 34.183, -118.668 34.183, -118.667 34.183, -118.667 34.18...>
⋮	⋮	⋮

Step 3: Shortest Path Determination

Task: Implement shortest path algorithm to approximate optimal traveling path between two nodes.

Approach: Apply the Dijkstra Single-Source algorithm to all nodes in dataset.

nx.single_source_dijkstra_path - computes the shortest paths between a source node and all nodes reachable from that node



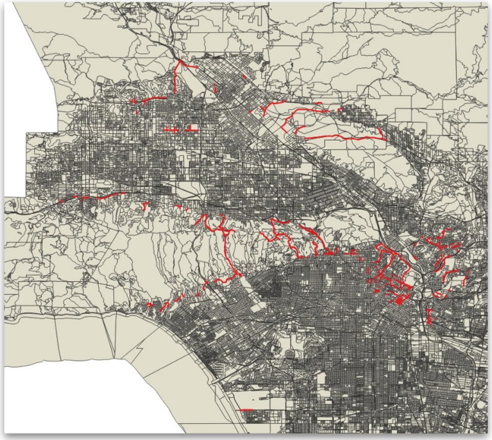
Dijkstra's Algorithm ([Source](#))

Graph in QGIS

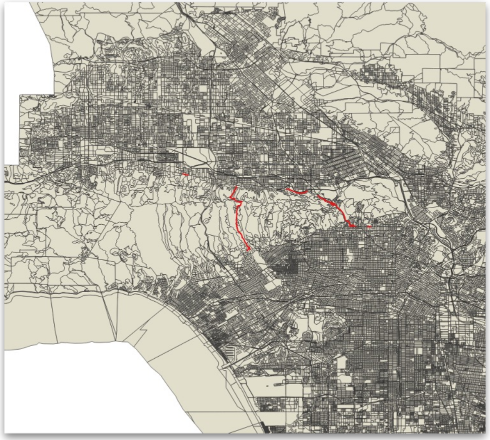
Step 4: Path Usage Estimation

Task: Compute road demand based on results from Step 3.

Approach: Efficiently iterate over all node-to-node connections from all shortest paths to compute relative usage for each road.



Graph in QGIS
["S000_adjusted" >= 0.01]

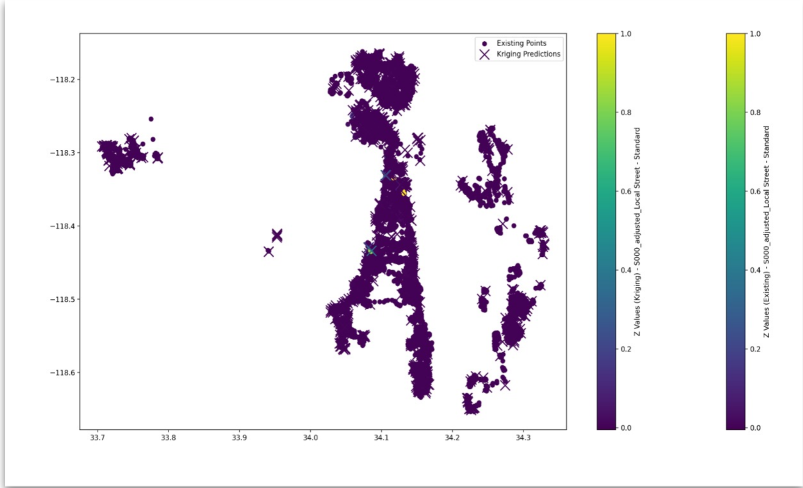


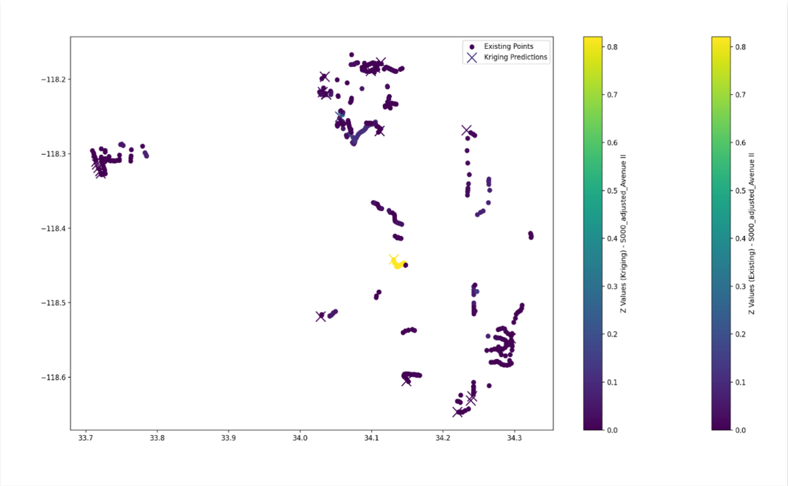
Graph in QGIS
["S000_adjusted" >= 0.50]

Step 5: Address Missing Weights - Kriging

Task: Address nodes with unidentified usage levels.

Approach: Utilize Kriging interpolation to estimate missing levels based on neighboring nodes.





Kriging results plotted
[Left: S000/Local Street, Right: S000/Avenue II]