Fast Computation of Clustered Many-to-many Shortest Paths

Aniket Sangwan - 180001005
Sarthak Jain - 180001047

Overview

There are certain situations in which it is necessary to efficiently compute shortest paths in a transportation network from each of a number of source nodes lying within a bounded area to each of a number of target nodes outside it. A number of successful algorithms based on a variety of approaches have been proposed in the literature for map matching sparse and noisy trajectories. They involve the computation of shortest paths from each candidate point inside the error region of a location measurement to each candidate point inside the error region of the subsequent location measurement. We aim at implementing an algorithm that helps in computing the MSP with high accuracy and reduced complexity.

Current Limitations

- Analysis indicates that, on average, about 90% of the overall computation time in a map matching algorithm is spent on shortest path computations when noisy trajectories are used. Since many applications need to map match streaming trajectories on the fly, there exists a need for accelerating the shortest path computations involved in map matching
- Some algorithms have been developed that are faster than the baseline algorithm but they offer significant drawback that they involve an offline preprocessing of the road network to construct a hierarchical representation making them impractical for networks where connectivity and edge lengths change dynamically.
- Most of the proposed algorithms consider both the source node and target node to be clustered. Our implementation provides the flexibility that the target node need not necessarily be clustered.

Objectives

- Learning basic shortest path algorithms like Dijkstra, Floyd-Warshall, Bellman Ford, etc.
- Implementing modified Dijkstra Algorithm to reduce the time complexity from $O(n^2)$ to O(nlogn).
- Implementation and analysis of existing MSP algorithms.
- To efficiently compute shortest paths from a cluster of source nodes to a set of target nodes.

Project Outline

The proposed algorithm consists of three computational steps as follows :

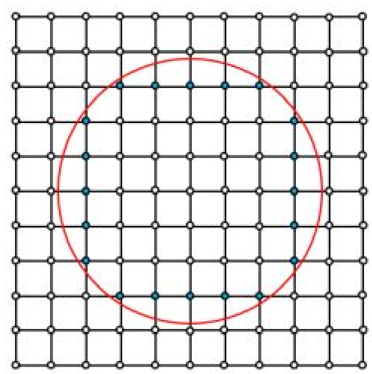

Fig. 2. The concept of potential exit nodes illustrated using a simple grid network. For the source region denoted by the red circle, the blue-shaded nodes are the potential exit nodes.

1. Identify the exit nodes in the source region and compute the shortest-path distances between each exit node and each target node.
2. Compute the shortest-path distances between each source node and each exit node.
3. By optimally combining the distances computed in the above steps, determine the shortest path distances between each source node and each target node.

After the implementation of the above algorithm, we intend to compare its complexity and efficiency with the existing algorithms and offer limitations to the implemented algorithm.

References

- George R. Jagadeesh and Thambipillai Srikantha. *Fast computation of clustered Many-to-many Shortest Paths and Its Application to Map Matching.* https://doi.org/10.1145/3329676
- Hannah Bast, Daniel Delling, Andrew V. Goldberg, Matthias Müller-Hannemann, Thomas Pajor, Peter Sanders, Dorothea Wagner, and Renato F. Werneck. 2016. *Route planning in transportation networks*.
- Sebastian Knopp. 2006. *Efficient Computation of Many-to-many Shortest Paths*. Diplomarbeit, Universität Karlsruhe.