Voronoi-Based K Nearest Neighbor Search for Spatial Network Databases

Ugur Demiryurek
Mohammad R. Kolahdouzan & Cyrus Shahabi
University of Southern California
Dept. of Computer Science
Los Angeles, CA 90089-0781
shahabi@usc.edu
http://infolab.usc.edu

Agenda
- Problem Definition
- Related Work
- Voronoi Diagrams
- Voronoi Based Network Nearest Neighbor (VN3)
- Experiments
- Discussion

Problem Definition

kNN Problem: Given a set of objects \( P \) and query point \( q \), find the \( k \) objects in \( P \) that are closest to \( q \).

Example 1:
- Finding the 3 closest shopping centers
Example 2: Finding the 3 closest restaurants to USC with Yahoo

Related Work

Query processing in SNDB: Papadias et al., VLDB 2003
Incremental Network Expansion (INE)
- Blind expansion hence redundant node access

Voronoi-Based K Nearest Neighbor Search for Spatial Network Databases

Related Work

Query processing in SNDB: Papadias et al., VLDB 2003
Incremental Network Expansion
- Incremental Network Expansion

Preliminaries: Voronoi Diagram

- Given set of sites (POI), a Voronoi diagram partitions the plane into disjoint Voronoi polygons for each site
- The region including a site $p$ includes all locations which are closer to $p$ than to any other object

Preliminaries: Voronoi Properties

- Property 1: Voronoi diagram is unique
- Property 2: Voronoi edges are shared by two generators in equal distance to neighboring generators
- Property 3: Nearest generator of $p$ is among the generators whose VP shares similar edges with $p$
- Property 4: Average number of Voronoi edges per VP is at most 6
Voronoi-Based K Nearest Neighbor Search for Spatial Network Databases

**Network Voronoi Diagram**
- Border Point: equal network distance to adjacent generators

**VN³ Approach**
- Offline Index Generation
  1. Network Voronoi Construction
  2. Index Generation (R-tree)
  3. Distance Precomputation
- Online Query Processing
  1. Find 1st NN
  2. Find k NN -> Filter & Refine

**Offline Step**
- Compute Network Index NVPs
- Precompute the Shortest Path distance from each generator to its border point
- Precompute the Shortest Path between border points

**Online Step**
- Find 1st NN
  - Search R-Tree to find the NVP that overlaps q
  - Report the generator of the NVP as the 1st NN; Cost: $O(\log n)$

**Performance of VN³**
- Data set:
  - Road network in Los Angeles (from NavTeq)
  - 110,000 streets, 79,800 intersections
  - Different points of interest: restaurants, auto services, schools, parks, shopping centers, hospitals
- Measured:
  - Query response time and CPU
    - Comparison with INE (Papadias et al. (vldb03))
    - Size of candidate set of VN³ filter
    - Comparison with R-tree-based KNN (Seidl et al. (sigmod98) and Hjaltason et al. (tod095))
Voronoi-Based K Nearest Neighbor Search for Spatial Network Databases

VN³ vs INE

VN³ finds the 1st NN using R-Tree
INE needs to expand the network around q

CPU Time:
INE uses a queue which is incrementally updated

<table>
<thead>
<tr>
<th>Data Source</th>
<th>VN³ Time</th>
<th>INE Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shopping</td>
<td>0.3</td>
<td>5.2</td>
</tr>
<tr>
<td>malls</td>
<td>0.2</td>
<td>4.1</td>
</tr>
<tr>
<td>schools</td>
<td>0.1</td>
<td>3.0</td>
</tr>
<tr>
<td>departments</td>
<td>0.05</td>
<td>2.0</td>
</tr>
</tbody>
</table>

Conclusion

- A novel approach for KNN queries in SNDB
- Based on:
  - Pre-calculating the solution space (first order Voronoi diagrams)
  - Pre-computing some distances (from borders to points inside each polygon)
  - Outperforms the only other solution for SNDB
  - Independent from object distribution
  - Outperforms the solutions for Euclidean space in “filtering”

Discussion

- What if the edge weights are changing?
- What if the both query and data objects are moving?
- What if you like to find the nearest hotel and gas station at the same time?