Fall 2024 EC504 Project Proposal:

Hashing and Trees for Fast Approximate Nearest Neighbor *Geospatial* Search

Chris Krenz, ckrenz@bu.edu, U19402301

This proposal is based on the Suggested Project 5: Hashing and Trees for Fast Approximate Nearest Neighbor Search. I will implement multiple algorithms/data structures to perform approximate nearest neighbor search of geospatial data and compare their relative speed and accuracy. Specifically, given a pair of coordinates—and additional dimensions, such as elevation, land type, and population density—the program will identify zip codes near those coordinates. I will implement and compare search algorithms based on: Multi-Table Locality-Sensitive Hashing (LSH),[1] Approximate K-D Trees with Priority Search,[2,3] and (potentially) R-Trees.[4] The datasets I currently plan to use include OpenStreetMap,[5] USGS,[6] NASA Earthdata,[7] and GADM.[8] I also plan to implement a simple interactive application that will allow users to perform these searches in real time.

One downside of using geospatial data instead of, say, image data, is that the data intrinsically has few dimensions. We would expect LSH to perform better than K-D Trees with high-dimensional spaces, so I anticipate the K-D trees will perform similar to, if not better than, LSH. As such, I will attempt to merge additional dimensions of data, such as elevation, land type (rural, etc.), population density, point-of-interest proximities, socio-economic data, vegetation, average temperature, etc. However, this comes with a substantial challenge in data cleaning and merging, as most of the data sources I have found provide data in different formats, and not all datasets provide coordinate information, making cross-referencing of data more difficult. I will attempt to strike a balance between feasibility and dimensionality. The upside of this lower-dimension data is that the datasets are a much more manageable size than most image datasets.

Feature List

- Primary Goals
  - Construction of searchable geospatial dataset
    - At a bare minimum, this dataset should contain data from OpenStreetMap to provide coordinates and GADM (or a comparable dataset) to provide zip codes. I am hopeful I can also integrate additional datasets listed above to increase dimensions.
  - Search based on multi-table Locality-Sensitive Hashing
    - To implement Multi-Table LSH, I'll create multiple hash tables where each table uses a different set of hash functions. For each query, I'll combine the results to find candidate nearest neighbors, improving the chance of capturing similar items. The goal is to reduce false negatives and boost recall.
  - Search based on K-D Trees with Priority Search
    - I will first construct a k-d tree, then during search, I will maintain a priority queue where nodes are prioritized by their (multi-dimensional) distance from the query coordinates. This prioritization will speed up the nearest neighbor search.
  - Algorithm to measure the speed and accuracy of these search algorithms
    - I will measure the speed by tracking the average query time per search as well as the time needed to build the data structure. For accuracy, I will compare the suggested options to the ground truth of

closest zip-codes. If a method retrieves one of the top few actual nearest neighbors, that will be considered a successful search.

- Stretch Goals
    - Implement R-Trees
        - To implement R-trees, I will create a tree structure where each node represents a bounding rectangle that encloses spatial objects. As data is inserted, nodes are split dynamically to minimize overlap between bounding boxes.
    - Implement Interactive App
        - Ideally I will be able to implement a GUI that displays the map that auto-focuses to the target region, places a pin at the selected coordinates, as well as highlights the zip-codes predicted by the algorithm. However, I may need to scale this back to a minimal GUI, some search boxes, drop-down lists, and checkboxes to fully specify a query and a frame to display the results in text form. Regardless, the user should be able to select one of the algorithms from a drop-down to compare performance.

Completed and Planned Tasks

- Completed
    - Topic chosen and researched: Week of Oct 14th
    - OpenStreetMap data acquired: Week of Oct 14th
- Planned (*see Timeline/Gantt chart for estimated timeline/schedule*)
    - Data Collection, Merging, and Cleaning
        - In addition to OpenStreetMap data, I am currently planning on using USGS, NASA Earthdata, and GADM data. I am figuring how to get the best cross-sections of USGS and Earthdata data, but GADM data has thus far proven inaccessible (download links not working; I will continue searching; ultimately this dataset may not even be necessary given other sources).
        - As I gather data, I will gauge the number of data sources I can feasibly access and merge (and therefore the number of dimensions in the final dataset). If necessary, I may decide to limit the geographic space (e.g. to North America) if doing so will enable me to have higher dimensional data.
    - Implementation of Multi-Table LSH and Approximate KD Trees with Priority Search
        - Once the data has been acquired, merged, and cleaned, I do not anticipate substantial challenges with implementing these algorithms.
    - Implementation of R-Trees (Stretch Goal)
        - Implementing R-Trees might be slightly more challenging as they are fundamentally different from KD-Trees and intended for range queries to produce exact matches, rather than approximate matches.
        - I welcome feedback on the importance of implementing this algorithm in a solo project, given the other objectives.
    - Creation of Interactive App (Stretch Goal)
        - I am hopeful I will be able to implement some sort of interactive application to make testing and running program easier. However, more sophisticated features, like a map display with real-time highlighting of or zooming in on target regions may not be feasible in the allotted time. As such, I may end up implementing a simpler version as described above.
    - Demo Preparation
        - The demo will hopefully consist of a live demo, with GUI, showing the program's search feature. Alternatively, I could perform the search from CLI or else prepare slides that highlight results—and metrics—of prospective searches.

Timeline/Gantt Chart

	Week of...						
	Oct 21	Oct 28	Nov 4	Nov 11	Nov 18	Nov 25	Dec 2
Data Collection							
Data Merging							
Data Cleaning							
Multi-Table LSH							
Approximate KD Trees with Priority Search							
Implementation of R-Trees (Stretch Goal)							
Algorithm Testing and Bug-fixing							
Gather Metrics and Benchmark Algorithms							
Creation of Interactive App (Stretch Goal)							
App Testing and Bug-fixing (Stretch Goal)							
Demo Preparation							

References

1. Datar, M., Immorlica, N., Indyk, P. & Mirrokni, V. S. Locality-sensitive hashing scheme based on p-stable distributions. in *Proceedings of the twentieth annual symposium on Computational geometry* 253–262 (Association for Computing Machinery, New York, NY, USA, 2004). doi:10.1145/997817.997857.
2. A comparison of k-nearest neighbour algorithms with performance results on speech data - KU Leuven. https://kuleuven.limo.libis.be/discovery/fulldisplay/lirias1945818/32KUL_KUL:Lirias.
3. Arya, S., Mount, D. M., Netanyahu, N. S., Silverman, R. & Wu, A. Y. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *J ACM* **45**, 891–923 (1998).
4. Hadjieleftheriou, M., Manolopoulos, Y., Theodoridis, Y. & Tsotras, V. J. R-Trees – A Dynamic Index Structure for Spatial Searching. in *Encyclopedia of GIS* (eds. Shekhar, S. & Xiong, H.) 993–1002 (Springer US, Boston, MA, 2008). doi:10.1007/978-0-387-35973-1_1151.
5. Geofabrik Download Server. *GeoFabrik: OpenStreetMap Data* https://download.geofabrik.de/ (2018).
6. Data | U.S. Geological Survey. *al Survey* https://www.usgs.gov/products/data (2024).
7. Earth Science Data Systems, N. Find Data | Earthdata. https://www.earthdata.nasa.gov/learn/find-data (2021).
8. GADM. *Database of Global Administrative Areas* https://gadm.org/data.html (2022).