An efficient approach to trajectory similarity range query based on Fréchet distance

Jiajie Ma
Secondary Student Training Program
The University of Iowa

https://www.biz.uiowa.edu/faculty/xzhou/#service

Xun Zhou, Ph.D., Assistant Professor, Dept. of Management Science
Vahedian Khezerlou Amin, Graduate Research Assistant, Dept. of Management Science

Introduction/Background

Nowadays, massive amounts of spatial trajectories are generated by transport, media and many devices. In Geographic Informatics System (GIS), they are defined as functions of time that describe the path of moving. A basic problem in this area is to quantize the similarity of a huge amount of spatial trajectories quickly and accurately, where many different comparison methods have been proposed.

In 19th Century, an effective measure of trajectory similarity, Fréchet Distance, was introduced with hard computational complexity. Intuitively, it is defined as “the minimal length of a leash that is necessary” for a man walking on a trajectory to walk his dog on the other, “allowed to control their speed but not go backwards” (Alt & Godau, 1993, p. 76). Based on this concept, fast and efficient algorithm for computing Fréchet Distance between two polygonal curves using dynamic programming with run time $O(pq(p+q))$ (Alt & Godau, 1993, p. 78) was developed as a reference solution.

To further reduce overall run time of Trajectory Range Query in practice, many researchers also took an insight into some robust trajectory indexing systems. Some common trajectory retrieval strategies were based on an index structure called R-tree, where its nodes contain specific spatial objects (points, etc.) or their MBBs (Minimum Bounding Boxes). With R-tree and its variants, range and nearest-neighbor queries in space are performed efficiently. (Pfoser, Jensen, & Theodoridis, 2001, p. 398).

Research Objectives

The goal of this research is to find an efficient algorithm for trajectory similarity range queries based on Fréchet Distance in very large databases, ($given$: a query trajectory P and a large set of trajectory data and a range $\epsilon \geq 0$; find, all the trajectories Q such that $F(P,Q) \leq \epsilon$.) where trajectories would be treated as polygonal curves in a 2-dimensional Euclidean system. The efficient algorithm was to be proposed by improving from the baseline approach with effective spatiotemporal indexing frameworks.

Method

The baseline approach is primarily based on Alt and Godau’s idea of Free Space Diagram (1993). However, instead of simply implementing Alt and Godau’s method of dynamic programming with run time of $O(pq)$ (Alt & Godau, 1993, p. 80), we notice that the route searching process can be perfectly described by a directed acyclic graph and thus, using depth-first search can reasonably make the expected run time below $O(pq)$. Applying this, by exhaustively searching the entire database, an accurate answer to the query can be obtained.

The second method is improved from our baseline by employing a filter-and-refine approach. For the filter step, we construct an R-tree index by inserting all the points in the trajectory databases. When filtering trajectories, we introduce the lemma that to become a candidate, all the points of the trajectory should be in the buffer of the query. However, due to the high computational cost of a buffer, we instead build a MMB around the buffer of the query trajectory to reduce the complexity. After that, the refine step will apply the baseline algorithm on all the candidates so that an accurate answer to the query will be obtained.

The third method adopts a similar idea as the second one. However, instead of indexing points in R-tree, we directly retrieve trajectories. To achieve this, we build MMB for each trajectory and insert the boxes into R-tree. Similarly, it can be proved that to become a candidate, the entire MMB of the trajectory should be in the MMB of the query buffer. Theoretically, this will further reduce the expected run time as the number of elements in R-tree is significantly decreased. Again, an accurate answer to the query will be outputted after the refinement on all candidates.

After implementing all three algorithms in C++ (and adopting boost::geometry::index::rtree (1.6.2) for R-tree construction), programs are submitted to University of Iowa Argon Cluster for experiments. During the tests, 20242 trajectories (ACM GISCup 2017) are used in total with query ranges randomly selected from 500 to 6000 (1%~20% of the average length of all trajectories). Meanwhile, linear for R-tree type and 20 for the maximum entries in the nodes are selected as our R-tree parameter choices.

Results

R-tree construction time, average filter time, average refine time and overall runtime are measured for all methods. The trajectory R-tree (third) method significantly outperforms the baseline approach on overall run time by nearly 6 hours and is better than point R-tree method due to savings on R-tree construction time and filter time. Furthermore, it is expected that the advantage of this algorithm will be more remarkable in even larger databases due to a positive trend of the ratio between the number of answers and candidates.

Conclusions/Implications

This research offers an efficient approach to trajectory similarity range query based on Fréchet distance in very large databases. With an effective spatiotemporal index, it reduces the hard computational complexity of the problem and largely improves past works in practice. However, the adoption of MMB in filter step leaves relatively huge amount of dead space in range query, which evidently increases the overall number of candidates. Thus, the future work can either change the number of bounding boxes around the query trajectory, or attempt grid-based index, which will both make the filter step theoretically more accurate.

References

[1] Alt, H., & Godau, M. (1993). Computing the Fréchet Distance between two polygonal curves. *Internat. J. Comput. Geom*, 5(1), 75-91. https://doi.org/10.1142/S0218195995000064

[2] Pfoser, D., Jensen, C. S., & Theodoridis, Y. (2001). Novel Approaches to the Indexing of Moving Object Trajectories. Proceeding from VLDB 2000: *The 26th International Conference on Very Large Data Bases*. San Francisco, CA: Morgan Kaufmann Publishers Inc.

[3] ACM GISCUP 2017. (2017). Trajectory dataset sample [dataset-sample.tgz]. Retrieved from http://sigspatial2017.sigspatial.org/giscup2017/download