INTERNATIONAL JOURNAL OF SCIENTIFIC PROGRESS AND RESEARCH (IJSPR) ISSN: 2349 - 4689
Volume-06, Number -01, 2014

Survey on Shortest Distance Computing by using Effective Indexing and Query Optimization Mechanism

Swati B. Ramteke[1], Prof. Anand Bone[2]
[1]Student, [2]Professor
[1,2]Dept. of Computer Engineering, SKN-Sinhgad Institute of Technology and Science, Lonavala, India

Abstract* - Now-a-days various application domains is significantly increasing, the several nodes may attain the scale of hundreds of millions or even more. Shortest distance query is a fundamental operation in huge-scale networks. Several previous methods in the review of literature take a landmark embedding method, which chooses a group of graph nodes as landmarks and computes the shortest distances from each landmark to all nodes as an embedding. To answer a shortest distance query, the pre-computed distance starting from the landmarks to the two query nodes are used to compute an approximate shortest distance based on the triangle inequality. In this paper, we analyze thefactors that affect the accuracy of distance estimation in landmark embedding. In exacting, it found that a internationally selected, query independent landmark set may introduce a large comparative error, particularly for close by query nodes. To deal with this issue, we propose a query-dependent local landmark plan, which identify a local marker near to both query nodes and provides more accurate distance estimation than the traditional global landmark way. This paper proposes proficient local marker indexing and recovery techniques with a scalable sketch-based index structure that not only supports estimation of distances among nodes, excluding computes equivalent shortest paths themselves to achieve low offline indexing complexity and online query complexity.

Key Words: Local landmark, Sketched-based index, Time complexity, Space complexity, Shortest distance query.

I. INTRODUCTION

As the size of graphs that emerge nowadays from various application domains is dramatically increasing, the several nodes may reach the scale of hundreds of millions or even more. Due to the enormous size, even easy graph queries become demanding tasks. One of them, the shortest path query, has been extensively studied during the last four decades. Querying shortest paths or shortest paths between nodes in a large graph has important applications in many domains including road networks, communication networks, social networks, the Internet, and so on. Graphs are routinely used in the modern digital world ina number of settings, such as online social networks (like LinkedIn and Facebook), biological interaction models, transportation networks [14], the massive hyperlink graph between documents of theWorldWide Web, XML data, and many more. Due to the ever increasing size of the graphs of an interest, many seemingly straightforward operations become challenging. In this paper, we turn our attention to the computation of shortest paths between any two nodes in the graph, a problem with long algorithmic history. This operation forms a building block for many mining tasks, and is also an increasingly important application in itself over instances such asonline social networks. Consider the following two application scenarios for shortest path queries:

A. Social networks such as LinkedIn enable professional networking among individuals. A person interested in reaching out to the hiring manager of his favourite future employer would seek a shortest path to reach that person, starting from his friends.

B. Biological (metabolic) networks (such as the Biochemical Network Database [11]) model the complex chemical processes within an organism. A biologist may be interested in identifying biotransformation paths between two target metabolites to help in designing experiments in the wet lab.

Similar applications arise for almost every instance of graph data, which includes the commonly studied problem of finding a shortest route between two points in road networks [6]. In many cases, the graphs of interest comprise millions of edges and nodes. Thus, for performance reasons, each of the shortest path query instances has to be answered as fast as possible while minimizing the consumption of resources such as memory and processor cycles which can be further increase an efficiency finding shortest path.

II. Literature Review

What makes the shortest path computation particularly hardon large graphs? Dijkstra’s algorithm, the classical technique to compute the shortest path between two nodes in a graph has the asymptotic runtime complexity of O(m +n log(n)) , where n is the several nodes and m is the number of edges [5]. On one of the benchmark datasets that we use in this paper is any social network comprising about 3 million nodes and 220 million edges a straight forward implementation of Dijkstra’s algorithm takes more than 500 seconds on average. This is way too slow for most applications. The motive for this is that algorithm of Dijkstra has to construct and maintain shortest paths to all nodes in the graph whose distance to the source node is smaller than the distance from the source node to the destination node[2]. Consequently memory consumption of Dijkstra’s algorithm is very high, requiring to maintain a number of 2.5 million nodes in the heap for the Social site dataset, which is prohibitively expensive for simultaneous execution of many queries. The naive alternative of precomputing all-pairs shortest paths distances and maintaining them on disk for quick lookups is practically infeasible, requiring O(n2) space.

According to the findings in the literature [9], the problems with finding a path through a graph have usually been approached in one of two ways, which we shall call the Mathematical approach and Heuristic approach. The mathematical approach typically deals with the properties of abstract graphs and with algorithms that prescribe an orderly examination of nodes of a graph to establish a minimum cost path. The Heuristic approach typically uses special knowledge about the domain of the problem being represented by a graph to improve the computation efficiency of solutions to particular graph-searching problems.In order to expand the fewest possible nodes in searching for an optimal path, a search algorithm must constantly make as informed a decision as possible about which node to expand next. If it expand nodes which obviously cannot be an optimal path, it is wasting effort. On the other hand, if it continuously ignore nodes that might be on an optimal path, it will sometimes fail to find such path and thus not be admissible. An efficient algorithm obviously need some way to evaluate available nodes to determine which one should be expanded next.

A* search algorithm uses estimates on distances to the destinationto guide vertex selection in a search from the source. Pohl [4] studied the relationship between A* search and Dijkstra's algorithm in the context of the P2P problem.He observed that if the bounds used in A* search are feasible, A* search is equivalentto Dijkstra's algorithm on a graph with nonnegative arc lengths and therefore finds the optimal path. In classicalapplications of A* search to the P2P trouble, distance limits are contained in the domain description, with no pre-processing necessary. e.g., for Euclidean graphs, the Euclidean distance among two vertices gives a lower bound on the distance between them [9]. A* algorithm is Similar to Dijkstra’s algorithm but domain-specific estimates πt(v) on dist(v, t) (potentials). At each step it pick a labeled vertex with the minimum k(v)=ds(v)+πt(v). It gives best estimate of path length through v. Butin general, optimality is not guaranteed [10].

Nowadays commonly used embedding technique is marker embedding, where a group of graph nodes is selected as markers and the shortest distances from a landmark to all the other nodes in a graphs are pre-computed. Such pre-computed paths be able to used online to offer an approximate distance between two graph nodes based on the triangle inequality. According to the findings in the literature [7], the problem of selecting the optimal landmark set is NP-hard, by a reduction from the classical NP-hardproblems such as vertex cover or minimum K-center [12].As a result, the existing studies use random selection or graph measure-based heuristics such as degree, between centrality, nearness centrality, exposure, and etc. In spite of a variety of heuristics that try to optimize marker selection, every existing technique follow the triangulation based distance assessment, which calculate the shortest distance between a pair of query nodes as the sum of their distances to a landmark. As the landmark selection step isquery independent, the landmark set provides a single global view for all possible queries that could be diameter apart or close by.

Thus, it is hard to achieve uniformly good performance on the entire queries. As an effect, the marker embedding technique may introduce a large comparative error, particularly when the landmark set is far-away from both nodes in a query but the two nodes themselves are nearby [14]. For example, in a US road network with 24 million nodes and58 million edges, landmark embedding (with 50 randomlyselected landmarks) has a maximum relative error of 68 forone query between 10,000 arbitrary queries was tested.

According to the findings in the literature [8], in the context of road networks and moving objects, the original space contains two-dimensional objects: intersections (original nodes) connected by some streets. The query points in such spaces are usually moving objects (e.g. cars) travelling through the streets from a source to a destination and the KNN problem is defined as finding the closest points of interest (e.g. hospitals, gas stations) to the moving stuff. Various challenges in such scenarios are:

A. The distance function D between two original nodesin the road networks is usually specified as the length of the path between the nodes with some minimum weight (e.g., time to travel along the path). These weights result in complex algorithms for computation of distance functions (e.g. Dijkstra algorithm to findthe minimum weighted path in a network with complexity O(e + n logn), where e and n are number of edges and nodes in the network respectively).

B. When the query point q is a moving object, the distance function D from q to the points of interests is to be computed very often and in real-time. This renders the computation of complex distance functions impractical for real-time KNN queries for moving objects.

According to the findings in the literature [13], path-sketches technique can be effectively used in large graphs with small diameters (e.g., almost all online social networks). The path-sketches maintain the complete path information between everynode and a selected set of landmark nodes, computed as part of a graph preprocessing step. A set of lightweight can be developed, yet highly effective techniques that use path-sketches to significantly improve the quality of shortest path estimations. Along with estimates of shortest path distances, how to generate corresponding instances of shortest paths themselves with no computational overhead can be shown. In fact, this algorithm can be used to generate a queue of paths in increasing order of their lengths, an importantneed in many applications over social and biological networks. All these methods can be implemented in a fully functional large-scale graph processing engine,and evaluate against a number of real world large-scale graphs.

According to findings in the survey of [6], an alternative architecture for networkdistance prediction that is based on peer-to-peer is explored. Compared with client-server based solutions, peer-to-peer systems have potential advantages in scaling. Since there is no needfor shared servers, potential performance bottlenecks are eliminated, especially when the system size scales up. Performance may also improve as there is no need to endure the latency of communicating with remote servers. In addition, this architecture is consistent with emerging peer-to-peer applications such as media files sharing, content addressable overlay networks, and overlay network multicast which cangreatly benefit from network distance information.

The coordinates-based approaches fornetwork distance prediction in the peer-to-peer architecture can be proposed to ask end hosts to maintain coordinates (i.e.a set of numbers) that characterize their locations in the Internet such that network distances can be predicted by evaluating a distance function over hosts coordinates [6]. Another benefit of coordinates-based approaches is that coordinates are highly efficient in summarizing a large amount of distance information. Thus, this approach can be used to trade local computations for significantly reduced communication overhead, achieving higher scalability.

A global architecture for Internethost distance estimation and distribution which we call “IDMaps” (Internet Distance Map Service) can be used to find the distance information used by SONAR/HOPS [7]. The basic IDMaps architecture has been discussed and show, through the Internet experiments and simulations, that our approach can indeed provide useful distance information. We believe highly accurate distance estimates (say, within 5% of the distance measured by the end-host itself) are impossible to achieve efficiently for a large scale Internet service. While we may be able to achieve this level of accuracy for each path measured, an estimate based on triangulation of such measurements will see an accumulation of the error terms. Instead, the goal was to obtain accuracy within a factor of 2 with very high probability and often better than that. So this level of accuracy can be expected to be adequate for SONAR and HOPS servers.

III. CONCLUSION

We have prepared a survey report on different topics for finding shortest path in any network based system. This report covers all factors that affect the accuracy of distance estimation and various algorithms with their space and time complexities. Based on such report, we found that an efficient sketch based Query dependent local landmark scheme can be used to significantly reduce the distance estimation error as compared to other algorithms and the exact shortest distance between any two nodes in the network system can be calculated with minimum space and time complexity.

REFERENCES

[1] R. C. Prim. “Shortest Connection Networks and some Generalizations,” Bell System Technology Journal, 36:1389–1401, 1957.

[2] E.W. Dijkstra, “A Note on Two Problems in Connexion withGraphs,” Numerische Mathematik, vol. 1, no. 1, pp. 269-271, 1959.

[3] P.E. Hart, N.J. Nilsson, and B. Raphael, “A Formal Basis for theHeuristic Determination of Minimum Cost Paths,” IEEE Trans.Systems Science and Cybernetics, vol. SSC-4, no. 2, pp. 100-107, July1968.

[4] I. Pohl. “Bi-directional Search In Machine Intelligence,” volume 6, pages 124{140. Edinburgh Univ. Press, Edin-burgh, 1971.

[5] M.R. Garey and D.S. Johnson, “Computers and Intractability: A Guideto the Theory of NP-Completeness,” W.H. Freeman, 1979.

[6] F. B. Zhan and C. E. Noon. “Shortest Path Algorithms: An Evaluation using Real Road Networks,” Transportation Science, 32(1):65–73, 1998.

[7] P. Francis, S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt, and L. Zhang,“IDMaps: A Global Internet Host Distance Estimation Service,”IEEE/ACM Trans. Networking, vol. 9, no. 5, pp. 525-540, Oct. 2001.

[8] C. Shahabi, M. Kolahdouzan, and M. Sharifzadeh, “A RoadNetwork Embedding Technique for K-Nearest Neighbor Search inMoving Object Databases,” Proc. 10th ACM Int’l Symp. Advances inGeographic Information Systems pp. 94-100, 2002.

[9] A.V. Goldberg and C. Harrelson, “Computing the Shortest Path: A* Search Meets Graph Theory,” Proc. 16th Ann. ACM-SIAMSymp. Discrete Algorithms (SODA ’05), pp. 156-165, 2005.

[10] A.V. Goldberg, H. Kaplan, and R.F. Werneck, “Reach for A* Efficient Point-to-Point Shortest Path Algorithms,” Proc. SIAM Workshop Algorithms Eng. and Experimentation, pp. 129-143, 2006.

[11] J. K¨untzer, C. Backes, T. Blum, A. Gerasch,M. Kaufmann, O. Kohlbacher,“The Biochemical Network Database,” BMCBioinformatics, 8, 2007.

[12] T. H. Cormen, C. E. Leiserson, R. L. Rivest, andC. Stein. “Introduction to Algorithms,” MIT Press, 3rdedition, 2009.

[13] A. Gubichev, S. Bedathur, S. Seufert, and G. Weikum, “Fast andAccurate Estimation of Shortest Paths in Large Graphs,” Int’l Conf. Information and Knowledge Management pp. 499-508, 2010.

[14] Miao Qiao, Hong Cheng, Lijun Chang, “Approximate Shortest Distance Computing: AQuery-Dependent Local Landmark Scheme,” IEEE Transaction on Knowledge and Data Engineering, Vol. 26, NO. 1, January 2014.