INTERNATIONAL JOURNAL OF RESEARCH IN COMPUTER APPLICATIONS AND ROBOTICS
ISSN 2320-7345

OPTIMAL ROUTE QUERY BASED ON FORWARD SEARCH AND BACKWARD SEARCH

S.Karthik[1]

[1] *PG SCHOLAR, Sri Krishna College of engineering and Technology, Coimbatore,*
karthik.rathna10@gmail.com
Address: 1/1, Kaveri Nagar,Ramanathapuram,Coimbatore-641045,Tamilnadu, India
Phone: 0422-2321885
Mobile: +91 9940900780
Email ID: karthik.rathna10@gmail.com

Abstract

Given a set of spatial points DS, each of which is associated with categorical information, e.g., restaurant, pub, etc., the optimal route query finds the shortest path that starts from the query point (e.g., a home or hotel), and covers a user-specified set of categories (e.g., pub, restaurant, museum). The user may also specify partial order constraints between different categories, e.g., a restaurant must be visited before a pub. Previous work has focused on a special case where the query contains the total order of all categories to be visited (e.g., museum! restaurant! pub). For the general scenario without such a total order, the only known solution reduces the problem to multiple, total-order optimal route queries. As we show in this paper, this naive approach incurs a significant amount of repeated computations, and, thus, is not scalable to large datasets. Motivated by this, we propose novel solutions to the general optimal route query, based on two different methodologies, namely backward search and forward search. In addition, we discuss how the proposed methods can be adapted to answer a variant of the optimal route queries, in which the route only needs to cover a subset of the given categories. Extensive experiments, using both real and synthetic datasets, confirm that the proposed solutions are efficient and practical, and outperform existing methods by large margins.

Keywords: ***Forward Search, Backward Search, spatial, route queries, optimal***

1. Introduction

Consider a tourist who will have a free day to travel around Hong Kong. Without much knowledge about the city, s/he searches online maps to plan for a trip. Usually, s/he has a fixed starting point, e.g., her/his hotel, and certain objectives in mind, such as visiting a museum, dining at a fine restaurant, and enjoying a few drinks at a local pub. Meanwhile, some destinations may need to be visited in a certain order. For instance, the trip should have a pub after a restaurant. The ideal route should cover all the destinations, satisfy all order constraints, and minimize the total travel length. Searching for such a route is captured by the optimal route query which usually has a vast search space, and, consequently, is too tedious to be done manually

Route Defining

In this Module the initial road dataset is loaded first. Then the starting and destination points are defined. Then the intermediate points were defined and a visit order constraints is defined. Now the query is defined (Query contains starting point, destination point, visit order points).

Forward Search Algorithm – Distance Calculation

In this module the algorithm traces the location of each point and computes a route by repeatedly connecting the current location to the nearest point. Similarly the routes are calculated for all the points and its locations are extracted. Based on the location latitude and longitude position the distance is calculated.

Forward Search Algorithm – Optimal Route finding

In this module after finding all the estimated paths the minimum distance is extracted between the points. Based on the minimum Distance points the shortest route is found between the routes and the points are joined using forward join algorithm. Finally we will obtain an optimal route from the forward search algorithm.

Backward Search Algorithm – Distance Calculation

In this module the algorithm traces the location of each point and computes a route by repeatedly connecting the last location to the nearest point. Similarly the routes are calculated for all the points and its locations are extracted in reverse manner. i.e. we calculate the distance between the point from destination to source points

Backward Search Algorithm – Optimal Route finding

In this module after finding all the estimated paths the minimum distance is extracted between the points. Based on the minimum Distance points the shortest route is found between the routes and the points are joined using backward join algorithm.

2. RELATED WORKS:

D .Zhang, B.C. Ooi, and A. Tung. [1] Used tags to build a general data model. Based on the model, it describes a system framework to support co-location searches on various types of resources in Web 2.0 applications. Developing an efficient indexing and searching strategy, which is scalable in terms of both the number of query tags and the data size of resources, to answer the queries of co-located tag matching. And finally extend the widely accepted tf-idf method for the geographical context, called the geo-tf-idf ranking method, to measure the relevancy of the geo-tags with respect to the area in which they are located.

Y. Y. Chen, T. Suel, and A. Markowetz[2] focused on the efficiency of query processing in geographic search engines, e.g., how to maximize the query throughput for a given problem size and amount of hardware. Query processing is the major performance bottleneck in current standard web search engines, and the main reason behind the thousands of machines used by larger commercial players. The problem of efficient query processing in geographic search engines, where each document (web page) consists of a textual part (bag of words) and a page footprint. Formally, a footprint is a function that assigns a nonnegative integer to each location in the underlying geographic domain. A query consists of a set of terms and a query footprint. The query processing problem is to determine the set of documents that contain all the query terms and that also have a nonempty intersection between document footprint and query footprint, and to compute a relevance score according to a given ranking function on all such documents.

Z. Chen, H.T. Shen, X. Zhou, Y. Zheng, and X. Xie[3] proposed that trajectory search has long been an attractive and challenging topic which blooms various interesting applications in spatial-temporal databases. In this work, a new problem of searching trajectories by locations is studied, in which context the query is only a small set of locations with or without an order specified, while the target is to find the k Best- Connected Trajectories (k-BCT) from a database such that the k-BCT best connect the designated locations geographically. Different from the conventional trajectory search that looks for similar trajectories w.r.t. shape or other criteria by using a sample query trajectory, we focus on the goodness of connection provided by a trajectory to the specified query locations. This new query can benefit users in many novel applications such as trip planning.

To answer the k-BCT query, we can separately issue at each query location an independent single point-based query to find out the nearest trajectories w.r.t. each location, and then the trajectories within the intersection of query results, if exist, are supposed to be close to all the query locations and thus with a good

connectivity. Based on this basic idea, we propose an Incremental k-NN based Algorithm (IKNN), which retrieves the nearest trajectory points w.r.t. each query location incrementally and examines the k-BCT from the trajectory points discovered so far. In this method, the pruning and refinement of the search are conducted by using the lower bound and upper bound of trajectory similarity that are derived from the found trajectory points, and the retrieval of nearest trajectory points is based on the traditional best-first and depth-first k-NN algorithms over an R-tree index.

X. Cao, G. Cong, and C. S. Jensen [4] proposeda new indexing framework for processing the location-aware top-k text retrieval (LkT) query. This framework integrates the inverted file for text retrieval and the R-tree for spatial proximity querying to obtain an Inverted file R-tree. Within the framework, an index approach called the IR-tree is proposed that is essentially an R-tree extended with inverted files. An associated algorithm is proposed for the processing of the LkT query that is able to prune the search space by simultaneously making use of both spatial proximity and text relevancy. Each node of the IR-tree records a summary of the location information and the textual content of all the objects in the sub-tree rooted at the node. The query processing algorithm utilizes the location index information to estimate the spatial distance of a query to the objects in the node's sub-tree, and it uses the text index to estimate the text relevancy scores for these objects. The paper's contribution is threefold. First, a new type of location-aware top-k text retrieval queries is introduced, LkT queries, that returns objects ranked according to a linear interpolation function that combines normalized location proximity and text relevancy. Second, to efficiently process the query, a new indexing framework that integrates location indexing and text indexing, IR-tree and an associated algorithm for processing the LkT query is proposed. A variant of the IR-tree, the DIR-tree, is proposed to incorporate document similarity when computing Minimum Bounding Rectangles, exploit document clustering to improve the indexing framework. Third, extensive experiments to evaluate the paper's proposals is conducted. Results of empirical studies with implementations of the proposed techniques demonstrate that the paper's proposals offer scalability and are capable of excellent performance.

3. PROPOSED SYSTEM

The proposed method implies a naïve approach to find the optimal route queries using two methodologies called Backward search and Forward Search. The backward search methodology computes the optimal routes in reverse order of its points. The forward search approach traverses the search space in a depth-first manner, and incrementally improves the bound for optimal route length. Forward search methods report results progressively, i.e., they first quickly produce one solution to the query, and then incrementally update it, until reaching the optimal one or being terminated by the user.

ADVANTAGES

- Increases performance level.
- Avoids redundant computations
- Efficient and robust solution.
- Returns optimal route and guarantees the quality of result.

5. CONCLUSION

This paper investigates the problem of optimal route query processing. Existing solutions are either limited to a specific setting of the problem, or incur expensive, redundant computations. Hence, novel and efficient solutions are proposed, based on two methodologies: backward and forward search. The solution BFS that combines merits from both backward and forward search achieves the best performance.

REFERENCES

[1] D .Zhang, B.C. Ooi, and A. Tung.Locating mapped resources in web 2.0.In ICDE 2010.

[2] Y. Y. Chen, T. Suel, and A. Markowetz. Efficient query processing in geographic web search engines. In SIGMOD, 2006.

[3] Z. Chen, H.T. Shen, X. Zhou, Y. Zheng, and X. Xie.Searching trajectories by locations: an effiency study.In SIGMOD, 2010.

[4] X. Cao, G. Cong, and C. S. Jensen.Retrieving top-k prestige-based relevant spatial web objects. PVLDB, 3(1):373–384, 2010.

[5] A. Guttman. R-trees: A dynamic index structure for spatial searching. In SIGMOD, 1984.

[6] F. Li, D. Cheng, M. Hadjieleftheriou, G. Kollios, and S.H. Teng. On trip planning queries in spatial databases.In SSTD, 2005.

[7] X. Ma, S. Shekhar, H. XiongsndP.Zhang.Exploiting a page level upper bound for multi-type nearest neighbor queries.In GIS 2006.

[8] M.Sharifzadeh, M.R.Kolahdouzan and C. Shahabi. The optimal sequenced route query. VLDB J.,2008.

[9] D.Zhang, Y.M. Chee, A.M., A.Tung, and M.Kitsuregawa. Keyword search in spatial databases: towards searching by document.In ICDE, 2009.

A Brief Author Biography

S.Karthik *– Completed B.E(COMPUTER SCIENCE AND ENGINEERING) in Tamilnadu College of Engineering from Anna University Coimbatore. Now pursuing M.E (SOFTWARE ENGINEERING) in Sri Krishna College of Engineering and Technology under Anna University, Chennai. My research interests include Data Mining and software engineering*