Location-Based Services Leveraging the Accurate and Efficient Way to Query Processing

SIRIGUPPA MANOJ KUMAR  
PG Student (Software Engineering)  
St. Johns College of Engineering and Technology  
Kurnool, Andhra Pradesh, India

T.K.SAGAR KUMAR  
Assistant professor, Department of CSE  
St. Johns College of Engineering and Technology  
Kurnool, Andhra Pradesh, India

Abstract—Location-based services (LBS) enable mobile users to query points-of-interest (e.g., restaurants, cafes) on various features (e.g., price, quality, and variety). In addition, users require accurate query results with up-to-date travel times. Lacking the monitoring infrastructure for road traffic, the LBS may obtain live travel times of routes from online route APIs in order to offer accurate results. Our goal is to reduce the number of requests issued by the LBS significantly while preserving accurate query results. First, we propose to exploit recent routes requested from route APIs to answer queries accurately. Then, we design effective lower/upper bounding techniques and ordering techniques to process queries efficiently. Also, we study parallel route requests to further reduce the query response time. Our experimental evaluation shows that our solution is three times more efficient than a competitor, and yet achieves high result accuracy (above 98 percent).

I. INTRODUCTION

The availability of GPS-equipped smartphones leads to a huge demand of location-based services (LBSs), like city guides, restaurant rating, and shop recommendation websites, e.g., OpenTable, Hotels, UrbanSpoon. They manage points-of-interest (POIs) specific to their applications, and enable mobile users to query for POIs that match with their preferences and time constraints. As an example, consider a restaurant rating website that manages a data set of restaurants P (see Fig. 1a) with various attributes like: location, food type, quality, price, etc. Via the LBS (website), a mobile user q could query restaurants based on these attributes as well as travel times on road network to reach them. Here are examples for a range query and a KNN query, based on travel times on road network.

II. RELATED WORK

2.1 Query Processing on Road Networks

Indexing on road networks have been extensively studied in the literature [19], [20], [22], [26], [28]. Various shortest path indices [19], [20], [28] have been developed to support shortest path search efficiently. Papadias et al. [26] study how to process range queries and KNN queries over points-of-interest, with respect to shortest path distances on a road network. The evaluation of range queries and KNN queries can be further accelerated by specialized indices [19], [22], [28]. In our problem scenario, query users require accurate results that are computed with respect to live traffic information. All the above works require the LBS to know the weights (travel times) of all road segments. Since the LBS lacks the infrastructure for monitoring road traffic, the above works are inapplicable to our problem. Some works [16], [21] attempt to model the travel times of road segments as time-varying functions, which can be extracted from historical traffic patterns. These functions may capture the effects of periodic events (e.g., rush hours, weekdays). Nevertheless, they still cannot reflect live traffic information, which can be affected by sudden events.

III. PROBLEM STATEMENT

In this section, we first describe the system architecture and then formulate the objectives of our problem. System architecture and notations. In this paper, we adopt the system architecture as depicted in Fig. 3. It consists of the following entities: _Online Route API. Examples are: Google/Bing route APIs [7], [4]. Such API computes the shortest route between two points on a road network, based on live traffic [6]. It has the latest road network G with live travel time information. Mobile User. Using a mobile device (smartphone), the user can acquire his current geolocation q and then issue queries to a location-based server. In this paper, we consider range and KNN queries based on live traffic. Location-Based Service/Server. It provides mobile users with query services on a data set P, whose POIs (e.g., restaurants, cafes) are specific to the LBS’s application. The LBS may store a road network G with edge weights as spatial distances, however G cannot provide live travel times. In case P and G do not fit in main memory, the LBS may store P as an R-tree and store the G as a disk-based adjacency list [26]. We then define route.

IV. CONCLUSION

In this paper, we propose a solution for the LBS to process range/KNN queries such that the query results have accurate travel times and the LBS incurs few number of route requests. Our solution Route-Saver collects recent routes obtained from an online route API (within d minutes). During query processing, it exploits those routes to derive...
effective lower-upper bounds for saving route requests, and examines the candidates for queries in an effective order.

We have also studied the parallelization of route requests to further reduce query response time. Our experimental evaluation shows that Route-Saver is 3 times more efficient than a competitor, and yet achieves high result accuracy (above 98 percent)

V. REFERENCES


AUTHOR’S PROFILE

S.Manoj Kumar received B.Tech Degree from St.Johns College of engineering and technology in Kurnool. He is currently pursuing M.tech Degree in Software Engineering specialization in St.Johns College of engineering and technology in Kurnool India.

T.K.Sagar Kumar Received B.tech degree from JNTU Anantapur and received M.tech degree from JNTU anantapur. He is currently working as assistant professor, Department of CSE, in St.Johns College of engineering and technology, Kurnool, Andhra Pradesh, India. His interests include Database management systems, compiler design.