Information Sciences 388–389 (2017) 62–83

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Parallel trajectory search based on distributed index

Hongzhi Wang*, Amina Belhassena

Harbin Institute of Technology, Harbin, Heilongjiang, 150001, China

ARTICLE INFO

Article history:
Received 25 January 2016
Revised 29 December 2016
Accepted 2 January 2017
Available online 6 January 2017

Keywords:
Trajectory search
Distributed R-tree
Activity trajectory
Spark

ABSTRACT

With the massive location-based information contained in public available GPS enabled mobile devices, there are problems of storage and queries processing. Trajectory search processing has received increasing attention in recent years. To organize the huge information, many studies have used indexing large trajectory data sets. However, the majority of existing studies have focused on a centralized system, while Spatial Hadoop is proposed to handle the spatial data by MapReduce method in a distributed system. However, this method is not suitable for reusing data without storing it on a disk. Therefore, we have developed a new R-tree index in the distributed system called the Distributed Trajectory R-Tree (DTR-Tree) with Apache Spark based on the MapReduce model. Furthermore, we have investigated a novel problem of a distributed trajectory query search with activities. Where a query q is given with a set of activities involved in each point of the trajectory and a threshold of distance, then the trajectory, which includes all the activities required with a minimal distance, is returned as a result. To optimize the trajectory processing search, we have used a novel strategy on separated smaller R-trees to obtain the sub-trajectories matching by q based on the spatial distance. In this paper, a balanced distributed index DTR-Tree is proposed to ensure the scalability and fault tolerance. Experimental results show the high efficiency of the proposed algorithms.

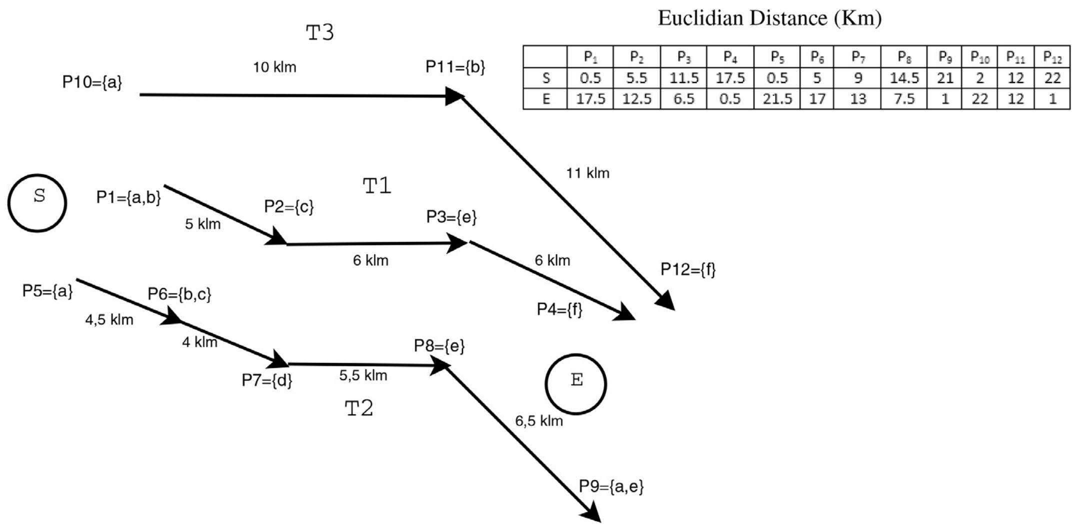
1. Introduction

With the proliferation of GPS enabled mobile devices and the rapid development of wireless sensor technology, a large amount of data capturing the motion history of moving objects, known as trajectories, is generated on an unprecedented scale. In such datasets, a trajectory is an ordered sequence of locations, and each location may have an associated description.

The trajectory similarity search has been extensively studied in the past two decades to achieve effective utilization of the trajectory data [1,5,14,15,28,30]. The trajectory similarity search is the basic operation of many trajectory-based applications such as trip planning and trajectory recommendation [6]. It has several different forms [6,9,36]. Among them, the keyword-based trajectory search is an important type of query. For such a query, a trajectory is modeled as associated geo-locations with keyword information. This query returns the k trajectories that contain the most relevant keywords. Furthermore, a minimum distance between locations in the trajectory may be necessary, which makes it a good choice to find the closest trajectory from their emplacements. Therefore, the process of this query is based on the shortest distance.

* Corresponding author.
E-mail addresses: wangzh@hit.edu.cn (H. Wang), amina_belhassena@hit.edu.cn (A. Belhassena).

http://dx.doi.org/10.1016/j.ins.2017.01.016
0020-0255/© 2017 Elsevier Inc. All rights reserved.



Euclidian Distance (Km)

	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}
S	0.5	5.5	11.5	17.5	0.5	5	9	14.5	21	2	12	22
E	17.5	12.5	6.5	0.5	21.5	17	13	7.5	1	22	12	1

Fig. 1. Trajectory query example.

A significant number of activity trajectories are used in real-life applications such as the description of points of interest in Google Maps[1], geo-tagged documents of Foursquare[2], and generated from the social media websites such as Facebook[3] and Bikely[4]. It involves some human life interests and favorites. Furthermore, many studies have been proposed to facilitate human life, such as route documents based activities [10], which is presented as a prototype system based on sensor annotated route data and that connect people with physical spaces and with other people, and Geolife projects [38–40] which represent a social networking services where GPS data is well managed and understood. It facilitated users to share their life experience and connect to each other with their location in the past.

Many types of queries have been proposed to retrieve these trajectory searches. One of the most interesting types of queries in recent trajectory research is to find the K activity trajectories that can cover intended activities in a given region with the minimum distance. The aim of this query is to attract people's attention in using both the spatial and activity domains. Consider the example demonstrated in Fig. 1, where T_1, T_2 and T_3 are historical activity trajectories, and query q is represented by the white circle. Each activity trajectory (e.g., T_1) passes several locations (e.g., P_1, P_2, P_3, P_4), and each place has one or more capable activities (e.g., $P_1 = a, b$). A tourist plan to visit some places in a city. For this, he specifies a query q with a start point S, a destination point E, a set of intended activities a, b, f and a distance threshold 25 km. T_2 does not include all the activities demanded by the user. Consequently, it will be eliminated. As all activities required by the tourist a, b, f can be fulfilled in T_1 and T_3, with the distance of 18 km and 24 km, respectively. The trip (S, P_1, P_2, P_3, P_4, E) of the historical trajectory T_1 with the minimum distance will be returned as a result.

The historical trajectory data are too large, and its process needs more computation, where it exceeds the capacity of traditional centralized technologies. The problem proposed in this paper is more challenging for the following reasons. To answer the query mentioned above, which returns the trajectory as a result based on the historical trajectory data, it is hard to design an index structure that can efficiently manage the huge information based on location, distance and activities. Efficient processing and analysis of large scale trajectory data have become an emerging and challenging tasks in industry and for researchers. Some models and algorithms are presented to handle the large-scale data trajectory [4,26].

Some efforts have been performed to handle large-scale data. MapReduce [11] is a programming paradigm used to handle large-scale data. It has been very useful with a high success at Google for many purposes. MapReduce-Merge was later proposed, when the MapReduce model did not directly support processing multiple related heterogeneous datasets to improve the MapReduce limitation [32]. SpatialHadoop[5] [13] was proposed to adapt the traditional spatial index structure, Grid, R-tree, and R+-tree, to form a two-level spatial index for MapReduce environments. The main drawback of the methods discussed above is that a large number of I/O disks is involved.

A MapReduce program consists of reusing a set of data across multiple parallel operations. It needs to reload data from the disk in each job and requires high costs. Apache Spark[6] handled this limitation by introducing a resilient distributed dataset RDD [33]. RDD is fault-tolerant, which can be rebuilt if a partition is lost, parallel data structure that provides

[1] https://maps.google.com/.
[2] https://foursquare.com/.
[3] https://www.facebook.com/.
[4] http://www.bikely.com/.
[5] http://spatialhadoop.cs.umn.edu/.
[6] http://spark.apache.org/.