International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:10, No:12, 2016

An Activity Based Trajectory Search Approach

Mohamed Mahmoud Hasan, Hoda M. O. Mokhtar

***Abstract*—**With the gigantic increment in portable applications use and the spread of positioning and location-aware technologies that we are seeing today, new procedures and methodologies for location-based strategies are required. Location recommendation is one of the highly demanded location-aware applications uniquely with the wide accessibility of social network applications that are location-aware including Facebook check-ins, Foursquare, and others. In this paper, we aim to present a new methodology for location recommendation. The proposed approach coordinates customary spatial traits alongside other essential components including shortest distance, and user interests. We also present another idea namely, *"activity trajectory"* that represents trajectory that fulfills the set of activities that the user is intrigued to do. The approach dispatched acquaints the related distance value to select trajectory(ies) with minimum cost value (distance) and spatial-area to prune unneeded directions. The proposed calculation utilizes the idea of movement direction to prescribe most comparable N-trajectory(ies) that matches the client's required action design with least voyaging separation. To upgrade the execution of the proposed approach, parallel handling is applied through the employment of a MapReduce based approach. Experiments taking into account genuine information sets were built up and tested for assessing the proposed approach. The exhibited tests indicate how the proposed approach beets different strategies giving better precision and run time.

***Keywords*—**Location-based recommendation, map-reduce, recommendation system, trajectory search.

I. Introduction

FOR a long time the quantity of web clients has been expanding because of the quick development of the computerized world. Today, with the advancement in social networks, more individuals can see, talk, make new companions in different places, and also share data [1]. Each one of those components contributes towards the regularly developing number of web clients around the world.

In all cases, the conspicuous change of the web nature from "Just-Read" to "Social-Intelligent" turns into a key column towards the improvement of numerous current applications like Facebook and Twitter [2]. Also, the enormous utilization of these applications gave us huge measures of data and learning about users' behaviors, trends, and interests as it is utilized through recommender frameworks which will assemble and break down this information to fulfill user`s query [3].

With the wide spread of those social applications, a few new applications are advanced. Location-based suggestion is one of those applications that is at present utilized by many users to help them select the venues to visit. For the most part, recommender frameworks work for proposing the most fascinating things or items in view of user`s profile [4]. Hence, scientists are attempting to develop algorithms and assessment strategies that are proficient to keep up and streamline recommendations to be more relevant for the users, recommender frameworks work with some techniques which include [5], [6]:

1) Recommending things to users rapidly.
2) Recommending items in light of relevant data.
3) Recommending location in light of users` interests and other user's locations.

In addition, locations recommendation is thought to be the principle worry in location-based social network applications (LBSN) as it spotlights on recommending things taking into account users present location with other given characteristics like (users history, users friends, similitudes between users, and so forth.) [7], [8]. Moreover, it is genuine that people tend to lean toward locations beforehand attempted by their companions or relatives, a man is normally more fulfilled around a spot, or a film in the event that others have effectively attempted it before and ideally prescribed it too. This user-to-user fulfillment move contrasts in degree in light of the relations between individuals [9]. A key element in LBSN recommendation is the spatial location of required spot because of the way that the vast majority of the clients for the most part lean toward venues close to their locality [4], [10], in this manner, as the separation between the user and prescribed spots diminishes, the users will be more satisfied [11], [12]. Location recommendation has enhanced from recently prescribing a particular single one to an arrangement of numerous locations associated with others inside an unequivocal direction represented as trajectory [13], [14] Adjacent to the part of organizations in advancing location-based recommendation, is another pillar in this research direction. Applications like: Foursquare [15], Facebook [16], and Sindbad [17] are all cases of LBSN applications that rely on users' trajectory data. So, trajectory search process is defined as the quest for the best 1-N competitor route(s) from a vast arrangement set of paths to fulfill the users' prerequisites [18].

Contributions in trajectory search are lately enhanced because of the gigantic road networks and LBSN applications which give expansive scale information of locations and related properties [9], [19].

In general, trajectory search depends on two fundamental segments: 1) ***similarity search*;** which is concerned with looking at trajectories and retrieving the ones that are similar to each other based on some distance function [13], [20], [21]. 2) ***trajectory mining*;** it concentrates on the best way to

Mohamed Mahmoud Hasan, Management Information Systems Department, Faculty of Commerce and Business Administration, Future University in Egypt, Egypt (e-mail: Mohamed.Hamada@ fue.edu.eg).

Hoda M. O. Mokhtar, Information Systems Dept., Faculty of Computers and Information, Cairo University, Egypt (e-mail: h.mokhtar@fci-cu.edu.eg).

preprocess trajectories before searching within trajectories through performing a number of processes like removing noise, grouping similar trajectories, distinguishing trajectories` features [22]-[24]. Some research works in trajectory search area took care of the possibility of activity trajectory which means to endorse best related trajectory(ies) that meet users' needs through using trajectory metadata like (nearest direction, activities needed, rating, preferences, etc.) [25].

Some of past works were not productive to accomplish this objective in view of considering the trajectory as limited arrangement of geo-spatial focuses in space (latitude and longitude), which is deficient for noting users query, so researchers' primary fixation was finding and dissecting more elements identified with direction to enhance search results [11]. Regardless, part of explores in trajectory search field relied on upon shortest path just, in spite of the many-sided quality of this methodology added to execution of the calculation extraordinarily with vast data sizes [18], [21]. For that, performance improvement is viewed as a basic challenge to answer user inquiry through huge data volumes, paying tension to existing works [13], [26] that focused on handling past issues, results exhibited it performs well with small datasets. So, the need for parallelization for speeding the inquiry system over huge volumes of data turns into a need.

This paper presents the parallelization of trajectory search to achieve a set of desired activities through different points of interest to get best N-relevant trajectory(ies) that meets inquiry necessity desired by the user utilizing Map-Reduce

II. Related Work

Several recommendation algorithms had been considered and introduced, yet the majority of past endeavors demonstrated that recommendation intricacy was exceptionally influenced by the improvement of web frameworks [3], and also the advances in mobile devices. At first, researchers concentrated on understanding the recommendation procedure and the diverse algorithms to apply. For instance, in [27], they began to study recommendation frameworks and their principle components, after that, recommendation systems were distinguished and characterized, several methodologies were introduced including: 1) Collaborative-filtering; this methodology relies on assessing and sifting items in view of individuals' opinion [5], 2) Content-based-filtering; in this approach the recommendation is based on the correlation between the content of the items and the user`s preference [3], to improve content-based filtering recommendation, additional contextual information were considered in including (geographic location, time, and preference), 3) Check-in; a process which depends on sharing user current location and giving additional data like sentiment, positioning, or even photos can likewise be included [9]. In view of this approach, the authors in [19] found some valuable certainties: first time visit is constantly favored, areas close-by user are all the more fascinating, and fellowship influences user's opinions [28].

The vast majority of past works that connected these methodologies utilized particular traits like (home area) in building suggestion frameworks for seeking similar trajectory(ies) fields. For that, new contributions began to change and improve the search procedure and expand recommendation proficiency [1], [29], [30]. So, endeavors in recommender frameworks in trajectory search field attempted to discover how to add additional components to crude location data including activities that user needs to perform. Utilizing such activity data for fitting venues that could be prescribed or recommended. Fusing activity data in the trajectory search process recover more important trajectories that absolutely or somewhat satisfy user's prerequisites [11].

As of now, the standard hypothesis of trajectory search procedure is to suggest the closest trajectory based on distance [20]. But, with the intricacy of including new elements and the expanding in information constrained numerous analysts to enhance search algorithms down taking care of these deterrents

In [26], they proposed search methodology utilizing the possibility of r-tree to group location points through their spatial existence as a change of reversed rundown approach which noted in as trajectory will be sought through their amassed rundown of activities for constructing such matrix for each with all holding trajectories to prescribe best trajectory that satisfy users necessities [13], [11], [18].

III. Problem Statement

With the abundance of location information that is effortlessly shared over the web and all the more particularly over social networks, it is a characteristic result to utilize these free data to enhance location recommendation and trajectory search services. Having a prior knowledge about user's interests, and what activity he desires or earlier learning of his routine (preferences); prompting him with the best matching route to follow so as to satisfy or perform every one of his demands. A key component to give such a route is to search through past trajectories of different users to recover the ones that did likewise rundown of activities at the nearest set of locations.

TABLE I
Symbols and Definitions

Symbol	Definition
Dτ	A set of trajectories
τ	A trajectory
Aτ	Activities associated with trajectory τ
PL	A points of interest set needed by user
P	Point of Interest
AL	User required activity list
Pτ	Trajectory Point
Ap	Point of interest activities
x	Latitude of point
y	Longitude of point
Bτ	Best Related Trajectory
F(subτ)	List of filtered trajectories
sub τ	Sub Trajectory
R.d(p,Rp)	Related point distance
R.d(τ,PL)	Total related trajectory distance
RB	Region Bounding Trajectories

In this section the main notations and necessary definitions that we will used in the rest of the paper are introduced in Table I.

Definition 1. ***A trajectory*** (τ) is a continuous path that a user follows to reach his/her destination. A trajectory is usually represented as a set of 2-dimensional spatial coordinates of (x, y).

Definition 2. ***A point of interest*** (P) represents the location or venue (e.g. a club, a restaurant) that the user wants to visit. A point is represented as a pair (x, y), where x and y are the latitude and longitude of the point resp.

Definition 3. ***An activity*** (A_P) is an action, task, or interest that the user wants to perform when he/she visits the point of interest (P).

Definition 4. ***A Point trajectory*** ($P\tau$) is a set of trajectory points. Each point ($P\tau$) is a place that the user stops at to do an activity (A_P).

Definition 5. ***An Activity Trajectory*** ($A\tau$) represents aggregated list of total activities through the whole points of a trajectory.

Definition 6. ***A Filtered Trajectories*** f(sub τ) is a set of refined trajectories through removing points that do not hold any of users needed activities A_L, each contained item in this set called (Sub τ)

Definition 7. ***A Related point*** (R_P) is a set of all points from sub τ that contains all activities of corresponding point of interest p.

Definition 8. ***A Related point distance*** is the value calculated between each point in PL and corresponding R_P as R.d(p,Rp).

Definition 9. ***A Total related trajectory distance*** represents the sum of all distance values R.d(p,Rp) between all P_L and sub τ as R.d(sub τ,P_L) .

Definition 10. ***A Best Related Trajectory*** ($B\tau$) ss an activity trajectory that best matches the user's required activity list, $B\tau$ is chosen to be the trajectory that contains the maximum subset of activities in the user's activity list (A_L) with minimum traveling distance, $B\tau \subset D$.

Definition 11. ***A Main Coordinates (Co)*** a process which aims to extract main coordinates in given trajectory τ or interest points P_L to be compared with initiated region R_B.

IV. Proposed Solution

Given set of locations with list of activities to be performed in every location within travelling distance, the objective is to retrieve the most related trajectory(ies) identified with user's yearnings, with least cost (distance). The proposed algorithm is comprised of two principle stages; 1) Mapping trajectory set, 2) Recommendation List Refinement.

Specifically *Mapping Trajectory set* phase manages a set of trajectories for finding the best related trajectories that meet the user requests for accomplishing least distance value from his starting position, so to accomplish the progression of this phase, we built the accompanying procedures;1) ***Region initiation***, 2) ***Upgrade Region Limit***, 3) ***Extracting Accepted trajectories***, 4) ***Related distance calculation.*** Then, *Recommendation List Refinement* phase aims at filtering the list of trajectories resulted from previous phase based on user requirements. 1) ***Sort trajectories***, 2) ***Filter trajectories,*** 3) ***return best related-N***. So, the following sections are dedicated to describe main algorithm phases in details.

A. Phase 1: Mapping Trajectories Set (Dτ)

One of the key prerequisite of the proposed approach retrieving the best related N activity trajectories, where N is a whole number of trajectories, will be returned in view of user`s necessity (least distance value, activities should have been done in every p).

Finding related trajectory(ies) in Fig. 1 is our first worry in the solution, so for this issue, we go through following steps:

1) ***Trajectory Bounding Region Initiation***; this progression means to construct spatial locale region for pruning search area, so we emphasize all over existed points in the purpose of interest rundown PL that should be gone by, then we assemble principle facilitates Co(PL) for region limits creation RB. Co (τ) gathered for every trajectory τ in trajectory dataset Dτ to be contrasted with *Bounded Region* RB, and with activities required AL, current τ considered accepted on the off chance that it completely in RB and holds activities required from AL then added to related trajectory list R.
2) ***Bounding Region Validation***; after that, for each accepted trajectory that fits in RB and meets users' activities, which used to overhaul (update) region limits with this trajectory through contrasting its Co(τ) with Co(RB) to guarantee performance.
3) ***Trajectory Filtration*** F(sub τ); at that point accepted trajectory prepared to get just related points, with respect to every point in the trajectory we check it`s rundown of activities whether it contains AL, subsequently it will be added to a sifted list F(sub τ) everything is called sub trajectory(sub τ), F(sub τ) will be utilized to get related distance.
4) ***Related Distance calculation*** R.d(τ); thus ascertaining travelling total distance is a vital stride in proposed work to be within users desired value, so numerical value between relative trajectories (sub τ) in F(sub τ) and rundown of intrigued points of interest PL, that value will represent distance needed, this is finished by extricating list of activities Ap for every p in PL then having related point of interest from (sub τ) that holds Ap in related temp list Rp. Distance value is calculated between this p and corresponding related points in Rp to get related distance for this point as R.d(p,Rp) then aggregate related trajectory distance of τ communicated as whole of all related point distance came about between every inquiry point in PL and τ as R.d(sub τ,PL)= ΣR.d(p,Rp). To improve the inquiry, we monitor the base cost so far and prune a trajectory once its expense surpasses the base. The yield of this progression is a mapped subset of related activity trajectory Rτ which fits the required range and fulfills the required list of activities with least cost (distance).
5) ***Get Main Coordinate*** (Co); for a given set of points,

spatial location of every point such x and y speaking to (latitude, longitude) contrasted with each other, get most noteworthy (x,y) and least (x, y) to be included.

```
Input: Trajectories Dataset (Dτ), Activity List AL, Point of Interest List PL.
Output: list contains all related trajectories that might be recommended (Rτ) with corresponding distances Rd

Rτ =Null
Rd=Null
Rp=Null
Co=Null
Sub τ=Null
Distance=0
Total_Distance=0
1- Co=getMainCord(PL)
2- RB.InitiateBoundedRegion(Co)
3- ForEach trajectory τ ∈Dτ {
4- IF τ.compare RB {
5- ForEach point p∈τ{
6- IF p related to PL{
7-  Sub τ.add(p)
8-  }
9- }
10- ForEach p in p_l {
11- Rp=RelatedPoints(p,sub τ)
12- Distance=getRelatedPoints(p,Rp)
13- Total_Distance+= Distance
14- }

15- Rd.add(Total_Distance)
16- Co=getMainCord(Sub τ)
17- RB.Update_Region(Co)
18- Rτ.add (Sub_ τ)
19- }
20- }
21- Return Rτ,Rd
```

Fig. 1 Mapping Trajectories Set

B. Phase 2: Recommendation List Refinement

Having retrieved the set of related trajectories from **phase 1**, now it will be handled to get just N-best trajectories as indicated by its related distance value. For this objective, our next step is to refine this set by separating the best coordinating N-trajectory(ies)with least cost and all desired activities as appeared in Fig. 2.

We accept that travel time and distance are proportional; in this manner recovering a trajectory that permits the user to touch base at the purposes of interest applicable to his activities with the shortest distance infers briefest travel time too.

In spite of the fact that the best situation is to locate a definite match, where a past trajectory did the entire set of activities in the movement list and expended the minimum cost, this situation is not generally ensured. Consequently, in this progression, we attempt to unwind our choice criteria. We utilize the accompanying methodologies:

```
Input: Related List of trajectories (Rτ), Related distance list (Rd), user defined threshold (δ)
  Output: Best Trajectories List (Bτ)

  Bτ = NULL
  Distance=0
  Index=0
    1- Foreach number in N {
    2- Distance= Rd .getMin()
    3- IF Distance <= δ{
    4-  index= Rd.indexof(Distance)
    5-  Bτ = Bτ ∪ Rτ .get(index)
    6-  }
    7- }
    8- Return Bτ
```

Fig. 2 Recommendation List Refinement

1) ***Exact Match***: is the credulous methodology which recommends a way just on the off chance that it absolutely fulfills the entire activity list overlooking the calculated cost. On the off chance that more than one trajectory has an accurate match, the one with slightest expense is returned as shown in Figs. 1 and 2.
2) ***Partial Match+shortest distance***: in this approach, an exact match is not required, trajectory is returned in the event that it surpasses the user activity threshold however has the slightest expense appeared in Figs. 1 and 3.

```
  Input: Related List of trajectories (Rτ), Related distance list (Rd), user defined threshold for travelling distance (δ1). user defined threshold for needed activities (δ2), N-trajectories needed.
Output: Best Trajectories List (Bτ)

Bτ = NULL
 Count=0
 Distance=0
  1- Foreach n in N {
  2- Distance= Rd .getMin()
  3- Count=τ.AL.length()
  4- IF Distance <=δ1 && Count >=δ2{
  5- index= Rd.indexof(distance)
  6- Bτ = Bτ ∪ Rτ .get(index)
  7- }
  8- }
  9- Return Bτ
```

Fig. 3 Recommendation List Refinement Partially

3) ***Partial Priority Match***: in this approach, the client is required to set order for the desired activities, in light of that request the trajectory exercises list Aτ whether it matches AL (as indicated by user preferences) is returned in the event that it has the minimum expense, appeared in Figs. 1 and 4.

```
Input: Related List of trajectories (Rτ), Related distance list (Rd), user defined threshold for travelling distance (δ1), N-trajectories needed.
Output: Best Trajectories List (Bτ)

 Bτ = NULL
 Count=0
 Distance=0
 Activity_list=NULL
1- Foreach n in N {
2- Distance= Rd .getMin()
3- IF Distance <= δ1 {
4- index= Rd.indexof(Distance)
5- T= Rτ .get(index)
6- Activity_list= T.get(Aτ )
7- IF Activity_list.checkOrder(AL) {
8-  Bτ = Bτ ∪ T
9-   }
10-  }
11- }
12- Return Bτ
```

Fig. 4 Recommendation List Refinement Partially Ordered

V. Experiments Configuration

Proposed algorithm has been tested through real dataset of total check-in processes made from Foursquare [12] which have around 35k trajectory records and with 100k locations and 600k activities in USA. Then, MapReduce cluster built with 3GB RAM DDR3 and 2.4 GHz-i5 Processor with 50 GB storage drive settled in oracle virtual machine [31] on 64-bit Linux operating System with HDFS. Through these gathered information papers, primary commitment was contrasted with the another trajectory similarity search method which is trajectory grid index (GAT) [18].

At first, paper algorithm is executed with MapReduce frame work [32] using single node which is exhibited in Fig. 5 by observing the changes in user input N which represent number of best-returned trajectories. Also, in Fig. 6, number of activities desired per location, then point of interests that will be visited were changed and noticed in Fig. 7.

The proposed contribution advanced with parallelization we apply algorithm on multi hubs (Nodes) with 2GB trajectory dataset file from SNAP data sets [33] using Gowalla trajectory data [34] to show how Map Reduce quicken execution. Utilizing amazon web administrations [35] and the outcomes are demonstrated in Fig. 8.

VI. Experiments Evaluation

In this section, users' queries were tested and evaluated through the showed comparisons with past contributions in Figs. 5-8. These experiments proved the paper contribution and showed that the performance of users' query is significantly optimized and accelerated compared with the previous works.

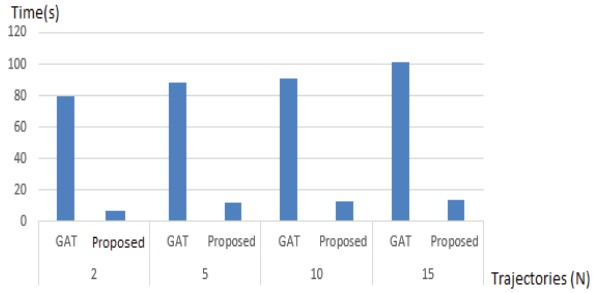

Fig. 5 Effect of changing N

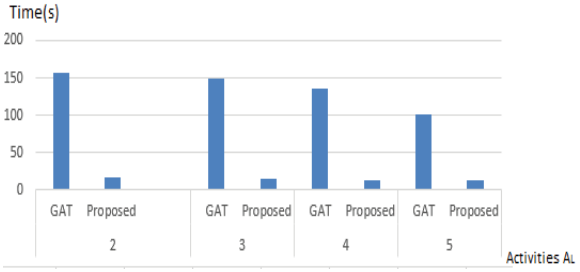

Fig. 6 Effect of changing activities per location AL

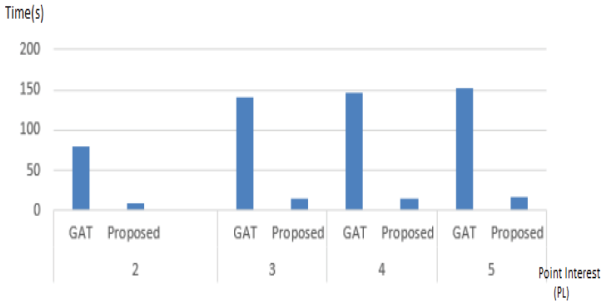

Fig. 7 Effect of changing point of interest PL

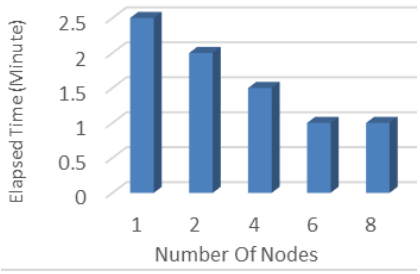

Fig. 8 Effect of cluster multi nodes setup

References

[1] Lee, F. Gregory, MaoYe and Wang-Chien, "Location recommendation for out-of-town users in location-based social networks," in The 22nd ACM international Conference on information & knowledge management, San Francisco, 2013.

[2] N. Li and G. Chen, "Analysis of a location-based social network," in Computational Science and Engineering, 2009.

[3] G. Adomavicius, B. Mobasher, F. Ricci and A. Tuzhilin, "Context-aware recommender systems," in Recommender systems handbook, Springer US, 2011, pp. 217-253.

[4] M. Prem and V. Sindhwani, "Recommender systems 2011.," in Encyclopedia of machine learning., US, Springer, 2011, pp. 829-838.

[5] J. B. Schafer, D. Frankowski, J. Herlocker and S. Sen, "Collaborative filtering recommender systems," in Foundations and Trends in Human-Computer Interaction, ACM, 2011, pp. 81-173.
[6] H. Gao, J. Tang, X. Hu and H. Liu, "Content-aware point of interest recommendation on location-based social networks," in 15 Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015.
[7] C. Hongbo, C. Zhiming, M. Arefin Shamsul and Y. Morimoto, "Place recommendation from check-in spots on location-based online social networks," in Networking and Computing (ICNC), 2012 Third International Conference on, 2012.
[8] H. Yin, Y. Sun, B. Cui, Z. Hu and L. Chen, "LCARS: a location-content-aware recommender system," in Proceedings of the 19th ACM SIGKDD international conferenceon Knowledge discovery and data mining, 2013.
[9] H. Wang, M. Terrovitis and N. Mamoulis, "Location recommendation in location-based social networks using user check-in data," in Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems., 2013.
[10] Noulas, S. Scellato, N. Lathia and C. Mascolo, "A random walk around the city: New venue recommendation in location-based social networks."," in 2012 International Conference on and 2012 International Confernece on Social Computing, 2012.
[11] S. Shang, R. Ding, B. Yuan, K. Xie, K. Zheng and P. Kalnis, "User oriented trajectory search for trip recommendation," in Proceedings of the 15th International Conference on Extending Database Technology, 2012.
[12] M. Sarwat, "LARS*: An efficient and scalable location-aware recommender system," in Transactions on Knowledge and Data Engineering, vol. 26.6, IEEE, 2014, pp. 1384-1399.
[13] Z. Chen, H. T. Shen, X. Zhou, Y. Zheng and X. Xie, "Searching trajectories by locations: an efficiency study," in Proceedings of the 2010 ACM SIGMOD International Conference on Management of data, 2010.
[14] J. Bao, D. Wilkie, Y. Z. Zheng and M. Mokbel, "Recommendations in location-based social networks: a survey," GEOINFORMATICA, pp. 525-565., 2015.
[15] (Online). Available: https://foursquare.com.
[16] (Online). Available: www.facebook.com.
[17] M. Sarwat, J. Bao, A. Eldawy, J. J. Levandoski, A. Magdy and M. F. Mokbel, "Sindbad: A Location-based Social Networking System," in Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, 2012.
[18] K. Zheng. S. Shang, N. Jing and Y. Y, "Towards efficient search for activity trajectories," in 29th International Conference In Data Engineering (ICDE), 2013.
[19] R. M. Feitosa, S. Labidi, L. S. A. dos Santos and N. Santos, "Social Recommendation in Location-Based Social Network using Text Mining," in 4th International Conference on Intelligent Systems, Modelling and Simulation, 2013.
[20] Ma, H. Lu, L. Shou and G. Chen, "KSQ: Top-k similarity query on uncertain trajectories" in IEEE Transactions on Knowledge and Data Engineering, 2013.
[21] L.-A. Tang, Y. Zheng, X. Xie, J. Yuan, X. Yu and J. Han, "Retrieving k-nearest neighboring trajectories by a set of point locations," International Symposium on Spatial and Temporal Databases, 2011.
[22] e. a. Monreale, "Where will I go next?: Predicting Future Categorical Check-ins in location-based Social Networks," in In Collaborative Computing: Networking, Applications and Worksharing , 2012.
[23] H. Su, K. Zheng and K. Zeng, "STMaker–A System to Make Sense of Trajectory Data," in Proceedings of the VLDB Endowment, Chicago, 2014.
[24] K. Zheng, Y. Zheng, J. Yuan and S. Shang, "On discovery of gathering patterns from trajectories.," in 2013 IEEE 29th International Conference in Data Engineering (ICDE), 2013.
[25] Y. Zheng, Z. Chen and X. Xie, "Searching Similar Trajectories by Locations". US Patent 12/794,538, 8 December 2011.
[26] Y. Gui-juna and Z. Ji-xianb, "A Dynamic Index Structure for Spatial Database Querying Based on R-Trees," in Proceedings of International Symposium on Spatio-temporal Modeling, Spatial Reasoning, Analysis, Data Mining and Data Fusion, 2005.
[27] J. Bao, Y. Zheng, D. Wilkie and M. F. Mokbel, "A Survey on Recommendations in Location-based Social Networks," ACM Transaction on Intelligent Systems and Technology., 2013.
[28] M. Sridevi, R. R. Rao and M. V. Rao, "A Survey on Recommender System," International Journal of Computer Science and Information Security, vol. 5, no. 14, p. 265, 2016.
[29] Hongbo, C. Zhiming and M. S. Arefin, "Place Recommendation from Check-in Spots on," in Third International Conference on Networking and Computing, 2012.
[30] R. Yerva, F. Grosan, A. Tandrau and K. Aberer, "TripEneer: User-based Travel Plan Recommendation Application," in 7th International AAAI Conference on Weblogs and Social Media., 2013.
[31] ORACLE, "Oracle VM VirtualBox," Oracle, (Online). Available: https://www.virtualbox.org/wiki/Downloads.
[32] K.-H. Lee, Y.-J. Lee, H. Choi, Y. D. Chung and B. Moon, "Parallel data processing with MapReduce: a survey.," ACM SIGMOD, pp. 11-20, 2012.
[33] Stanford, "Stanford Network Analysis Project," Stanford University, (Online). Available: https://snap.stanford.edu/index.html.
[34] Cho, S. A. Myers and J. Leskovec, "Friendship and Mobility: User Movement in Location-Based Social Networks," in ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2011.
[35] AMAZON, "Amazon Web Services," (Online). Available: https://aws.amazon.com/.