A Hybrid Preference Oriented Collaborative Filtering Technique for Context-aware Point of Interest Search

Hamid Turab Mirza[1], Naeem Qasim[1], Ibrar Hussain[2], Mujahid Rafiq[2] ,Safdar Ali[2], Hafiz Mahfooz Ul Haque[2]

[1]Department of Computer Science, COMSATS University Islamabad, Lahore Campus, Pakistan
[2]Department of Software Engineering, The University of Lahore, Lahore, Pakistan

ABSTRACT
Nowadays, location-based services are in high demand due to the ubiquitous use of mobile devices. In such services, the location of a user is commonly utilized as a search criterion to find user's Point of Interest (PoI). Existing approaches suffer from various limitations such as ignoring user preferences in the search criteria, scalability issues along with data credibility problems in public evaluation strategies. Consequently, most users are not satisfied with the search results in the absence of such rich information. This work introduces a novel technique to search for K-nearest points which are preferable to the user, by utilizing searching time as well as query location. Specifically, this work has proposed a hybrid system that employs feedback learning algorithm, collaborative filtering and Google page rank algorithm. The feedback learning algorithm and collaborative filtering are used to enable continuous learning and improve the predictive accuracy respectively. Google page rank is utilized to increase the credibility of public evaluation while calculating the score of the PoI. The proposed system is experimentally evaluated on a benchmark data obtained from yelp.com. The results revealed a significant gain in performance and accuracy.

Keywords:
Location-based search; preference learning; feedback learning; collaborative filtering; recommendation system.

I. INTRODUCTION

Lately, web-based search became very common and popular due to the unprecedented growth of the internet. With the development of smart wireless devices such as GPS enabled devices, mobile phones, and PDA phones, the popularity of Location based Services (LBS) is also gaining much attention. LBS enable mobile users to search anything from anywhere at any time [1, 2]. Currently, there are a number of location-based search platforms such as iPeen[1], MapQuest[2], Google maps[3], PAPAGO[4] which provide facility to search nearby locations. Their search results depend upon a user's current query location. Furthermore, with the development of Web 2.0 technologies [3], people have made available their business information; like business hours, location, availability, and products along with their features.

By using detailed point's information such as business hours, location, special offers and price range etc., more advanced location-based searching systems can be developed which can generate more accurate and specific results. For example, searching for a nearby public transport, restaurant or a friend in an area, searching for a nearby university in a particular city or a shopping store in the adjacent shopping mall, searching for a petrol pump nearby or a hospital with best doctors in that vicinity etc. To elaborate the importance of this topic further, Let's suppose a scenario where a traveler who is visiting a specific city for the first time and is feeling hungry wants to find a restaurant nearby which comes into the price range of his preference and desire at that time but has no idea about the multiple choices available accordingly. Traveler's situation could be solved using different ways i.e. using a web search engine or by using restaurant recommendation websites but there are certain issues in these systems. First of all, the user will have to provide his/her current location to that search engine but because of the absence of any landmark, it may be difficult to identify his/her current location. Secondly, suppose if the recommender system identifies his/her location and shows nearby restaurants, even then the quality of a restaurant may not be guaranteed. There exist few websites (e.g. iPeen, PAPAGO) which recommend nearby restaurants by taking public likings into account, so in this mechanism quality of search can be increased. But the problem still persists in this approach which doesn't guarantee that public favor is same as of this specific user. Therefore, for better location-based search results, user's preferences should be augmented with his location.

Several works [4-9] have attempted to address most of the discussed issues but are mostly limited to only one

[1] http://www.ipeen.com.tw/
[2] http://www.mapquest.com/
[3] https://www.maps.google.com
[4] http://www.papago.com.tw/

Manuscript received January 5, 2026
Manuscript revised January 20, 2026
https://doi.org/10.22937/IJCSNS.2026.26.1.10

business domain such as restaurant or hospitals with a limited number of points. Additionally, these proposed approaches utilized the quantitative-based feedback learning and their Collaborative Filtering (CF) methods have scalability issues. Moreover, the problem with the traditional recommenders [8, 10-12] is that while ranking searched points they do not consider users' temporal information into account; this indicates that recommender may recommend a restaurant or any other business which is closed at that time while user wants to find some point which is still opened. Another problem inevitable in this situation is lack of user preference information for example; user wants to search for a restaurant with economical rates while the results shown by the systems may not accommodate this preference.

Works [2, 4] have focused on utilizing a user's current location, preference and query time but were unable to optimally rank nearby PoI's. There are platforms i.e., Google maps, and PAPAGO which attempted to rank points based on the distance between a user and the searched point. Other platforms (i.e. iPeen) used public evaluation strategy to rank points but none of them took user preferences into account. Moreover, to increase the accuracy and precision of ranking results two mechanisms have been used which are: relevance feedback [13] and collaborative feedback [4, 5, 8]; but none of the works have applied both mechanisms while considering user preferences [14, 15]. Moreover, none of the frameworks have taken Google ranking approach into account. Though the work of [1] simultaneously takes user preference learning, relevance feedback and collaborative feedback learning approaches [11] into account to build user query results. However, their work is specific to the shopping stores only and thus raises scalability concerns. Their proposed CF algorithm is not as scalable as user based collaborative filtering techniques are utilized.

Main contributions of this work are listed below:

1) This work proposes a new ranking algorithm by providing effective and realistic public evaluation score in case of PoI search. The proposed technique extends the public evaluation score to the weighted sum of business/recommended websites and Google search engine (page ranking).
2) We introduce item-to-item based collaborative filtering approach to efficiently learn particular user preferences. For PoI search, user's preference is learned by utilizing a similarity score between two users searching for a common interest. Conversely, this work presents a novel idea of item-to-item similarity to develop optimized learning of preferences.
3) Nevertheless, the proposed approach can be utilized for any types of PoI search items.

The rest of the paper is organized as follows. Section II reviews the related work. Problem definition is elaborated in Section III. In Section IV, we present the system framework and its components in detail. Complexity analysis and experimental design are discussed in Section V and VI. The result of experiments is reported in Section VII. Finally, we conclude our work in Section VIII.

II. RELATED WORK

In this section, we discuss previous studies related to location-based searching techniques, which can be divided into three categories (a) Location based Search (b) Ranking Systems (c) Recommendation Systems.

A. LOCATION BASED SEARCH

Location based searching services are accessible to users through mobile devices by utilizing the mobile network and the ability to make use of GPS service which detects the geographical position of the mobile device. Users may benefit from location-based search in a variety of contexts in work and personal life. Such as finding out a nearby doctor, locating a shop, checking weather conditions, investigation, tracking a parcel or vehicle using vehicle tracking services.

In recent years, location-based services have been studied by several authors [11, 12, 16]. In [2] the author proposed a location-based social networking service called GEOLIFE 2.0, which is a social networking service incorporating users, locations and user-generated GPS Trajectories. In the real world, people try to access sequence of location and generate many trajectories in form of GPS logs, based on these GPS logs three graphs can be built: a location-location graph, a user- location graph, and a user-user graph. [17] proposed a shop recommendation system based on individual user preferences and needs. The system recommends frequently visited shops to users using a custom developed algorithm. For popular location-based search services, Google maps provide the search capability for nearby targets. [1] proposed preference oriented data mining techniques for location-based store search which takes user preferences into account for searching nearby shopping stores. However, the above discussed works are not valid for all types of points like searching of nearby hospitals, parks etc. Moreover, none of the works utilized Google ranking algorithm to rank nearby points.

B. RANKING SYSTEMS

Ranking of results according to its relevance to the query is one of the most fundamental problems in the area of information retrieval. The articles, topics or items which are relevant to each other are ranked at the top of the list. Typically ranking is done by calculating the score of each item by using its attributes values. There are varieties of means to evaluate ranking functions but most popular are ranking top “k” items. One of the most used and popular examples of a ranking algorithm is page ranking approach used by Google to rank search result. PageRank is a way of measuring the importance of the website in a particular query. Other examples are CRR which was introduced in 2010 which is pointwise and pairwise Combined Regression and ranking algorithm. Bayes Rank was introduced in 2009 which is a list wise method that combines Plackett-Luce model and neural network to reduce the expected Bayes risk. Specifically, for LBS, different works [1, 3] introduced variations of ranking algorithms. The work presented in [1] proposed item ranking approach for hotels/stores which ranked stores based on the preferences of similar users. Ranking function presented by [1] utilizes public evaluation, distance, and user feedback to calculate the score of certain stores. However, considering only the public approach as evaluation criteria for ranking search items decreases the credibility of their proposed approach. Moreover, their public evaluations were collected from restaurant recommendation websites which may not be a credible way to collect data and use it as public feedback.

Google ranks businesses by giving scores based on the popularity of that business. Google uses page rank algorithm to calculate the rank of each website/business e.g. Facebook has the rank of 9, Google itself has the rank of 9 and LinkedIn holds a rank of 8. That rank shows that how much that business is popular and it gives us an abstract overview of the business’s credibility along with its usage. This work aims to improve ranking algorithm by implementing a hybrid approach to calculate the ranking of points constituting of not only stores but approximately all points by introducing Google ranking approach to calculate the score of each point.

C. RECOMMENDATION SYSTEMS

Recommender systems normally termed as recommendation engines work using a specific technique of information filtering system and help to present things like images, videos, places, books to the user which are new and the user has not rated them before. Such recommender systems compare the items based on specific characteristics and try to predict a rating for a new item that is most likely to be given by that user. In conclusion, the main purpose of a recommender system is to give a suggestion of new things or seek out the usages of some item for a specific user by showing that item in recommendation list of that user.

Collaborative filtering is a technique mostly used by recommendation systems for their purpose [11, 18, 19]. [5] worked on the location-based PoI systems and investigated some issues in these systems. This research has proposed an enhanced CF-based collaborative filtering technique and proposed a restaurant recommender system. In [20] a new methodology has been proposed for online learning of social computing based interest sharing. This research proposed a model for users using the internet that allows them to discover their common PoI like sets of URL frequently searched.

In [21] the authors proposed a methodology to find out service similarity for privacy by using location-based search queries. Their proposed research was a user-centric and location-based architecture that was capable enough to customize query results so that it can include neighbor points of interest. In [22] a survey is presented on location positioning and privacy preservation methods in location-based service. This survey was based on existing methods which were dealing with localization techniques for both outdoor as well as indoor techniques of location privacy protection. The author proposed a taxonomy to help researchers to quickly understand existing works, challenges and possible improvements.

The work presented in [23] proposed a technique which was a personalized manufacturing service recommendation system using semantics-based collaborative filtering. This was a novel collaborative filtering method for automating the semantics of manufacturing services. In [24, 25] proposed a time-aware recommender system utilizing the dependency network of items. In [26] proposed a recommender method in a collaborative tagging system by utilizing time-sensitive topic recommendation. In [27] the authors proposed a hybrid recommender system which was based on user-recommender interaction. [1] have proposed a recommender system by utilizing user to use collaborative filtering algorithm which calculates the similarity of users and recommends stores to the user but due to lots of comparisons between users and volatile calculations, it may not be considered as a scalable algorithm [12, 28, 29].

Keeping in mind existing lacks in previous studies our research presented a novel approach which consists of a hybrid ranking algorithm by utilizing Google ranking approach and utilizing item-to-item based collaborative filtering techniques to recommend a PoI.

III. PROBLEM DEFINITION

Definitions of basic concepts and terms are given below before formally defining the problem.

Definition 1. (Preferences/Features) $F = \{f1, f2, f3....f|F|\}$ defines list of all features, every user will have certain features.

Definition 2. (Users) $U = \{u1,u2,u3....u|U|\}$ defines system users which will use system to search PoI's.

Definition 3. (Similar Users) $SU= \{u1, u2, u3....u|SU|\}$ defines a list of all users who are similar to other users due to the similarity of preferences.

Definition 4. (Location) $L = (x, y)$ defines coordinates of location L. x will denote latitude and y will denote longitude.

Definition 5. (Distances) $Dist = \{dist1, dist2....dist|Dist|\}$ defines distance between searching user and searched PoI in Kilometers. The ranking system will use Dist to rank points.

Definition 6. (Time Slots) $T = \{t1, t2, t3....t|T|\}$ defines list of time slots in a day. User's searched query will always be under certain time slot. Every time slot in a day will be unique because user preferences will be different in different time slots. Our proposed approach will filter user preferences according to time slots.

Definition 7. (Points) $P = \{p1,p2....p|P|\}$ defines list of points that user wants to search. Every point will contain certain features like geographical location, public evaluation feedback, opening time, closing time etc.

Definition 8. (Ratings) $RV = \{1, 2, 3, 4, 5\}$ defines list of rating values which user will use to rate PoI's. Rating values vary on a scale of 1 to 5 where 1 is for not preferred and 5 is for most preferred.

Definition 9. (Rating Date) $RD = \{date1, date2....date|Date|\}$ defines list of dates in which user rated the PoI's. Rating date could be different from query date because there is a chance that user may give rating days after querying.

Using the above definitions, suppose a user U places a query at time T from location L to search nearby points P with available features F. So by using user's current location and searching time, current problem is to develop an optimized location based point searching system which, by using user current location and searching time, will be able to provide list of nearby points that will be preferred by the current user. Moreover, it is expected that after learning in a short span of time system will return a more accurate ranking list. To be specific we aim to develop an optimized location based point searching system under distance, time and user preferences constraints. These notations are given in Table 1

TABLE.1

Notation Table

Notation	Description
F	Collection of preferences
U	Collection of users
SU	Similar users whose preferences matches with others
Dist	Distance between user and points
L	Location of point and users
T	List of time slots
P	Collection of all points need to be searched
V	Rating values used in feedback
RD	Rating dates

IV. SYSTEM ARCHITECTURE

Figure 1 shows the proposed system architecture. This research work aims to extend the POLS [1] framework and our approach attempts to extend POLS to multiple business points abstaining us from the limitation of any specific domain, unlike [1] which is restricted to shopping stores, different features and diversified nature of businesses are the reason POLS approach is unable to scale for multiple businesses.

The proposed system consists of two parts client module and a server module. Client-side module is used to request services from the server side using location-based data. Moreover, the server module is further composed of two layers' middleware and data storage layer. Middleware intercepts initial requests and filters out PoI's based on user preferences and query time. After that middleware runs a ranking process to fetch related PoI, calculate their score and sort PoI according to score. PoI with the highest score tops the ranked list and results are sent back to the user.

Furthermore, the system merges Google ranking approach with existing ranking strategy to calculate user's most preferred PoI. Google ranking results are incorporated due to their high credibility. Moreover, a personal preference database to store positive and negative preferences of every user is proposed. Using this database, the system applies ranking algorithms [4, 5] to rank nearby PoI using the user's current location, searching time and personal preferences. The user gives feedback and personal preference database is updated using the proposed feedback learning algorithm. Every user feedback is stored in a global preference database where the system applies collaborative filtering (CF) algorithm [17, 30] to calculate similar users based on positive feedback. CF results update the personal preferences of the user by recommending points which a user may find informative. Feedback learning algorithm learns preferences right after receiving user feedback but collaborative feedback learning is carried out periodically in intervals because of its lengthy processing

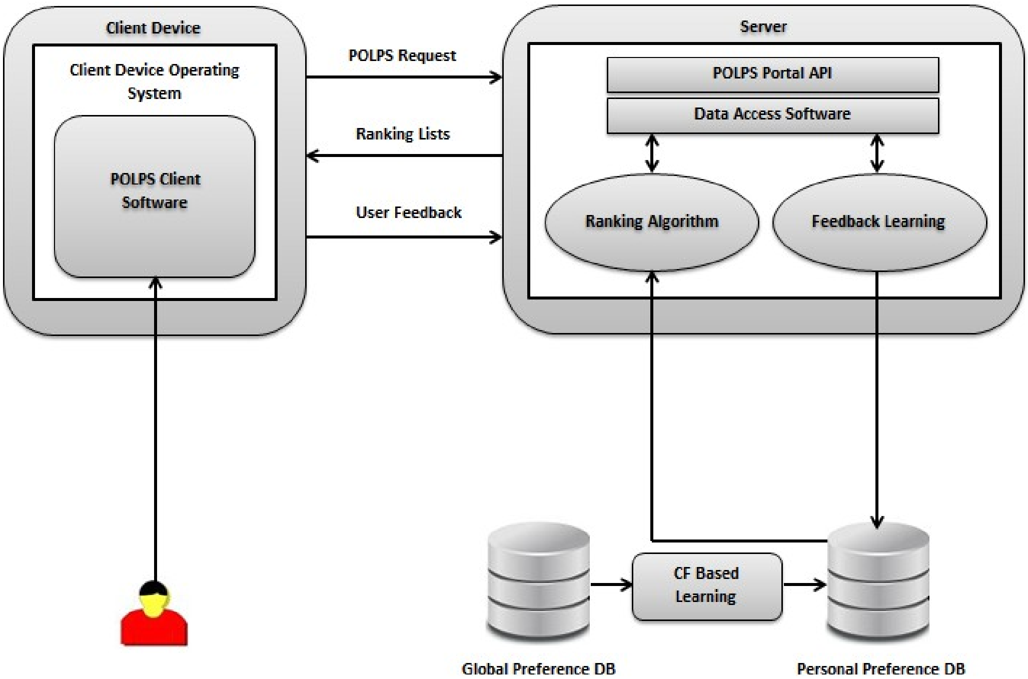

Figure 1: System Architecture

time and high memory consumption it is not possible to carry out in real time.

Proposed Ranking algorithm, feedback based algorithm and collaborative filtering based algorithm are elaborated below.

A. RANKING ALGORITHM

Purpose of a ranking algorithm is to provide a precise and accurate list of ranking points. It prioritizes PoI's using public evaluation, user's preferences, and current location. The studies presented in [2, 4, 5, 11, 15] only discussed one or two factors to rank PoI's. However, these studies did not consider user's personal preferences to search query result. Consequently, whenever user will query their system before applying the ranking algorithm, the system will also filter all those PoI that are closed (shops or business closed) at the time of query and will filter by comparing their query time and business hours. The system will assign a score to the remaining PoI by using the scoring function. A higher score will show that points are most preferred and should be shown accordingly. Assuming user u with searched point p scoring function to calculate the score of PoI p will be defined as

$$S(u, p) = \alpha \times (SpEval(u, p) + \beta \times SDist(u, p) + \gamma \times SPref(u, p) \quad (1)$$

Hence, Equation 1 will be the sum of three scores which are public evaluation, the distance score, and preference matching score. Here α, β and γ will be weights of all three scores such that the sum of all weights will be equal to 1 like $\alpha+\beta+\gamma=1$.

Here public evaluation is a score evaluated from points/businesses recommendation sites like YELP5. To increase the credibility of public evaluation PoI system has extended public evaluation score into two subscores: first is calculated from businesses recommendation websites and second is calculated through the Google search engine (https://www.google.com/business/). Based on NDCG analysis our proposed algorithm allocates 60% weight to public evaluation and 40% weight to the Google search engine score. Assuming user u with searched point p, the scoring function to calculate public evaluation score will be

$$SpEval(u, p) = \alpha \times SEval(p) + \beta \times SGoogle(p) \quad (2)$$

[5] http://www.yelp.com

Here α value is the weight of public evaluation with value 0.6 and β is the weight of Google search engine with value 0.4. All values are normalized in the range of 0 and 1.

Here the score of distance depends on the behavior of the user. The system calculates preferred distance using historical rating logs of the user, which might be preferred distance from the actual user to point.

$$dist\Pr ef(u,p) = \frac{\sum_{j=1}^{n} distActj}{n} \tag{3}$$

Here distAct (u, p) is the actual distance between user u and point p in kilometers, and n is the number of ratings which user u has given. Moreover, it is proposed that point will be marked as not preferred if there exists some other point spatially closer to the user under the same category with same features as of the point being searched by the user e.g. a car oil changing workshop having credit card facility. Hence we designed the distance score SDist (u, p) to assign a lower distance score for those points which are farther and their actual distance is greater than preferred distance. On the contrary, points which are close enough, such that their actual distance is nearly equal or within the average preference distance, will be given the highest score that is 1.

$$Sdist(u,p) = \begin{cases} 1 & \text{if distAct(u, p) = distPref (u, p)} \\ 1 - distAct(u,p)/R & Otherwise \end{cases} \tag{4}$$

distAct (u, p) is the actual distance, distPref (u, p) is the user preferred distance defined as (3) and R is a distance range with user location which is tolerable distance range, this range is predefined by PoI system for each user in kilometer, for example, 2 km. The tolerable range could vary based on query time with distance keyword, for example, 0.5 km and 3 km.

$$Spref(u,p) = \sum_{\forall PPu \in Fs} StrPPu - \sum_{\forall NPu \in Fs} StrNPu \tag{5}$$

User preferred distance can be different in different time slots. For example, the user can prefer spatially nearer restaurants, doctors, shopping malls in the morning office hours due to the shortage of time but may prefer a fancy restaurant at a considerable distance in the free evening hours. Spref (u, p) is the score calculated by matching preferences of user u and the features of point p. The system will calculate this score by subtracting negative feedback from positive feedback. For each user, there is personal preference database which contains his/her positive and negative preferences. Each preference point is assigned a score from 1 to 5 with strength value ranging from 0 to 1. Preferences database contains a mixture of both types, for example, user U1 personal preference database may contain three positive preferences and two negative preferences, a record e.g. {economical, 0.9} in the positive preference means that the user likes and highly prefers economical points. Equation 5 shows the preference matching score, where Fs shows set of features for point p, PPu and NPu are defined as positive and negative preference rated by user u respectively and strength values of positive and negative preferences are represented by StrPPu and StrNPu, respectively. If there are two points like point A and point B, each contains three point features, the point may be a shop or any other business, then using above equation the preference matching scores for point A and point B can be calculated as 0.8 and 0.4 respectively.

B. FEEDBACK BASED ALGORITHM

The system provides a list of ranking points and the user rates those points according to his/her preferences. To gather feedback of a particular point under consideration, the proposed system uses corresponding preference strength values from list {1.0, 0.5, 0, 0.5, 1.0}. Feedback value 1 and 2 means feedback is negative and strength will be 1 and 0.5 respectively. Feedback 3 means neutral feedback and its strength will be 0. Feedback 4 and 5 means positive feedback and strength values will be 0.5 and 1.0 respectively.

For each point, the system records point rating provided by the user into positive databases if its preference value is above 3 and in negative if preference value will be below 3. For example, user, u rated a shopping mall "A" as 5 on Feb 15. The system defined that the corresponding strength value for rating 5 is 1.0. So the system will put all features like cheap rates, parking availability, credit card facility, children friendly, wheelchair accessible, elevators of that shopping mall into positive preferences of user u with strength value 1.0 and dated Feb 15. Rating of point shows that user "u" liked this shopping mall. There may be a chance that user "u" will again go to that shopping mall and again rate it so there should be a mechanism to combine both ratings into a single rating. Our proposed system combines them using rating day. Hence, the strength value for rating 4 is 0.5. As 10 to 20 days are passed since rating day; system found that "cheap rates" attribute is found two times in positive preferences database with strength value 1 and 0.5. To combine strength value StrN of attribute “cheap rates” system will follow the below technique.

StrN=(0.5*20-1)/ (20-1+10-1))+(1.0*10-1/20-1+10-1))=0.66

As the human preferences may change with the passage of time preference level of a particular point is gradually decremented. If the users’ preference is high, the effect of decay will be reduced and preferences will remain active. In other words, preferences will stay active if the user rates points for longer time. If a user stops rating a point, decay weight will be decreased to 0 and that preference will become inactive. The system applies decay function and based on how many numbers of days have passed; the system decreases each active preference strength value. The system will start calculation from day first. The decay function is defined as

D(StrN, dN, i) = StrN − eg(i)×(dN−c2i)

Where g(i)=1/1+clog10i

Here StrN is normalized preference strength value of a point which is currently active and dN −c2i is a controlled variable used to control the distance between the number of days which have been passed and weight of preference of the user. It is denoted by a parabolic curve and y-axis. Here Dn is the number of passed days, the system used it to set the weight of preference at y-axis, -c2i is a factor which is used to optimize the distance between the passed number of days and weight of user preference. Moreover, c is a gradient coefficient which is predefined with value 4. More the distance smaller will be the value of -c2i. In Equation 6, g(i) is denoted by the gradient rate.

By taking all coefficients of Equation 6 into account, the system will find out any active preference with a negative strength which means that rating is passed for a long time. So the system will start an updating process and will set that preference to inactive in the user preference database.

C. COLLABORATIVE FILTERING BASED LEARNING

To learn the preferences of users globally this work proposes Collaborative Filtering (CF) based learning algorithm. The algorithm learns the preferences of a user with respect to other similar users. The system aims to have the capability to recommend PoI’s by utilizing personal and potential preferences of the user. To implement this algorithm system has stored reviews of users into rating logs; this rating is submitted by all users. By following review log, instead of analyzing user to user preferences between subject user and other users [1], the system analyzes the similarity of preferences between the item and other items, the item here is PoI. For items with highly similar preferences sets, the system transfers those similar items into the recommendation of all those users who have rated the current item.

Item to item collaborative technique [30] is beneficial in terms of both performance and accuracy. Users-to-user collaborative techniques are beneficial when datasets contain more points than users. In the case of this system, the dataset contains 366715 users and 61184 thousand points. In the case of such systems, user-to-user CF techniques tend to be costly and are less scalable due to N*N comparisons between users, and their recommendations will be non-static.

This work employs Item-to-Item collaborative filtering technique to find out the similarity between a user’s preferences. To find out the most similar match, Algorithm 1 first creates a table of similar items just by finding those items which have been rated by both users. The system builds a point to point matrix by iterating over all item pairs and computes a similarity metric for every pair.

```
For each user U who rated point Pi
    For each point, Pj rated by user Ui
        Record that a User Ui rated Pi and Pj
    For each point Pj
        Compute the similarity between Pi and Pj
    Find out most similar user Uj of Ui
        Recommend points of Uj to Ui
```

ALGORITHM 1 Item to Item Collaborative Filtering

However, there could be many point pairs with no common users. To resolve this issue, we have computed the similarity between two points by using cosine similarity [12, 31]. Suppose A and B are two points whose similarity needs to be calculated, to add it to the preferences of the user who also rated A; proposed algorithm used following cosine similarity technique to calculate similarity score of A with B.

Similarity (A, B) = cos(A, B) = A.B/||A||*||B||

Here |A| and |B| are magnitudes of positive preferences of both vectors. PoI’s having highest similarity are considered as most similar points and system adds such points into the preferences of all those users who rated current point.

V. COMPLEXITY ANALYSIS

This section presents time complexity analysis of the proposed technique and explores best, average and worst cases. The time complexity of proposed technique primarily depends upon three factors: i) number of users “UN” using the system, ii) number of points/businesses “PN” in the database and iii) number of similar users “SU”. The last factor is most important as CF based Item to Item recommendation algorithm uses similar users to recommend points/businesses. To analyze the time complexity of the proposed system big O notation is used as below.

Best case: O(UN*PN)

System latency increases with the increase in number of users and points. As the number of users increases system performs more calculations combined with the overhead for feedback learning. Similarly, the system needs to perform more comparisons due to increase in the number of points. Comparisons are performed on properties/attributes of the points and preferences of the user, e.g. a user has a preference that in evening he/she prefers to visit malls which have shops selling clothes/garments on clearance sale along with the availability of elevators and car parking etc. However, if a particular mall possesses attributes that are “clearance sale on garments” and “availability of elevators” but car parking is not available then the system will remove that mall by comparing user preferences with mall attributes.

Worst case: O(UN*PN+CFN)

System latency increases with the increase in UN and PN. Increase in the number of users and points results in more comparisons and system has to find similar users and recommendation of those points to the users which he/she has not rated in the past. So, main issues that affect time complexity of the system are UN, PN, and CFN.

Average case: O(UN*PN+CFN)

Time complexity of the system is same for both average and worst cases because the system has to perform additional comparisons to find out similar users. Time complexity of the proposed system and POLS [1] is same.

VI. EXPERIMENTAL DESIGN

This section elaborates experimental design for comparing the proposed system recommendation capability with POLS [1] recommendations. Nevertheless, both proposed technique and POLS have similar key purposes i.e. ranking and prediction of PoI’s. Prediction accuracy is used to measure the recommendation quality provided by the system. To evaluate and validate the proposed approach, we have organized a set of experiments to address the following questions.

1) Does Google ranking affect the credibility of public evaluation?
2) Does proximity of location indicate similarity in points?
3) Is proposed “item to item collaborative filtering” technique better than existing “user-to-user collaborative filtering” approach?

Experiments are conducted using Oracle and Java developer tool. Oracle database is used for storage of points data coupled with its user feedback and evaluation results. All experiments are performed on a desktop computer with the configuration as Intel(R) Core(TM) i5-4200U having CPU 1.60GhZ, 4GB RAM, and Microsoft Windows 7 operating system.

A. DATASET PREPARATION

To evaluate the proposed system for multiple points we used two datasets namely Yelp dataset and UCI Machine learning repository. The dataset was obtained from a recommender containing 366715 users, 61184 points of 783 different categories (hotels, hospitals, shops etc.) with 889964 attributes and a total of 1566000 reviews. On average each point contained 15 attributes. Initially, data was prepared in CSV format and was later mapped to relational objects for computational purposes.

Table 2 shows some of the important attributes of points with their category. Each point has an ID, name, latitude, longitude to obtain its physical location, category, and schedule of each point for each day.

TABLE 2

Sample points under different categories

Point ID	Point Name	Latitude	Longitude	Category	Start Time	End Time	Business Days
134983	Restaurant and Bar	18.948657	-99.235361	Restaurant	11:00	23:30	7 days a week
134986	Restaurant Las Mananitas	18.928798	-99.239513	Restaurant	11:00	23:00	7 days a week
134999	Kiku Cuernavaca	18.915421	-99.184871	Restaurant	12:00	23:30	7 days a week
135041	Luna Café	22.15106	-100.977659	Restaurant	07:30	22:00	7 days a week
135072	Sushi Itto	22.149192	-101.002936	Restaurant	19:00	23:00	7 days a week
135123	Smart Hotel Lahore	18.875011	-99.2353499	Hotel	08:00	02:00	7 days a week
135124	Dr. Khurshid Alam	31.4665451	74.2737073	Doctor's Clinic	17:00	20:00	Thursday, Friday, Saturday
135125	Dr. Muhammad Fiaz	31.5141163	74.349148	Doctor's Clinic	18:00	22:00	Monday, Tuesday, Wednesday, Thursday, Friday
135126	Dr. Suleman Elahi	31.5752026	74.377818	Doctor's Clinic	15:00	20:00	Monday, Tuesday, Wednesday, Thursday, Friday
135127	Dr. Ayesha Sheikh	31.4805698	74.3853803	Doctor's Clinic	17:00	22:00	Monday, Thursday, Friday
135129	Dr. Ayaz Munir	30.2023601	71.4414767	Doctor's Clinic	14:30	20:00	Monday, Tuesday, Wednesday, Thursday, Friday

TABLE 3

List of sample users

User ID	Name	Latitude	Longitude	User Type
1003	Naeem Qasim	22.119847	-100.946527	Student
1004	Hamid Wyne	18.867543	-99.183	Professional
1005	Naseem Ali	22.183477	-100.95989	Student
1006	Ijaz Ahmed	22.158474	-100.983	Student
1007	Musa Naeem	22.118464	-100.938256	Professional
1008	Saqib Rasool	22.159427	-100.923811	Student
1009	Raheem Obaid	22.190889	-100.990448	Student
1010	Umer Saleem	23.724972	-100.998669	Professional

TABLE 4

Top ranked restaurants with respect to user location

S. No	Point Name	Score	Point Features	Distance (km)
1	Sirloin stockade	1.41	Spicy, Maintenance Staff	11.981
2	Restaurant Bar Coty	1.01	Parking	11.142
3	Chaires	1.01	Maintenance Staff	11.982
4	Crudlia	0.61	Parking	11.978
5	Puesto De Tacus	0.61	Spicy	11.978
6	La Posada	0.61	Maintenance Staff	11.978
7	Luna Cafe	0.61	Parking	11.978

G

B. CALCULATION OF GOOGLE SCORE

To increase the credibility of public evaluation score we utilized URL's of points websites and applied Google ranking algorithm to calculate their popularity on the web. This score is added to public evaluation score during calculation of ranking of points. To generate a dataset of users, 20 members of our computer science laboratory participated in the experiment. All participants were male and mean age was 24 years. Each user was allocated a predefined latitude and longitude of real locations extracted from the data set and was also allocated a predefined time slot so as to create a real-world scenario of different people querying the system from different locations at different times of the day. Participants were instructed to query the system during different time slots e.g. from 10:00 AM to 12:00 PM, 12:00 PM to 3:00 PM, 3:00 PM to 7:00 PM etc.

Manuscript received January 5, 2026
Manuscript revised January 20, 2026
https://doi.org/10.22937/IJCSNS.2026.26.1.10

Table 3 is showing a list of sample users using the system. Each user queried system 5 times in a day and continued doing so for 20 days. It is assumed that each user is under predefined timeslot and location. The system applies ranking algorithm and returns back a list of top 7 points under the searched category. Afterward, participants give their feedback against each ranked point based on their personal preferences under certain conditions. Table 4 is presenting the result of top-ranked restaurants that are spatially near to a user "u". Distance attribute is the distance in kilometers between user u and point p. We increased the number of points and users by merging Yelp users and points which contain actual information such as features, public evaluation reviews and business hours. We enriched the dataset by calculating

Google score for each point. Average number of preferences of a user is 6. To generate more and more ratings of points under different categories we designed a simulation of our PoI system in which users are created for experimental purposes, each user is assigned a location from different random locations and query time is assigned from predefined query slots. Moreover, every user has his/her own predefined preferences.

As compared to POLS [1] our system is scalable and capable to search points other than only shopping stores and restaurants. We have configured users to search for different points like doctors' clinics, hospitals, shopping malls, schools, petrol/gas filling stations, recreational parks etc. A user may search by specifying keywords which could be a name of any category like hospitals, parks, schools etc. When user queries our system, it takes location information, temporal information and preferences of user into account for ranking and retrieval purposes. After retrieving results from recommender system every user rates points by giving them a rating from 1-5. Proposed feedback learning algorithm learns user ratings and stores them in either positive or negative database based on the value given by the user. The system has configured users to rate points randomly by taking random number value with range 1-5 by considering public evaluation score, Google ranking score and distance of a point from the user. Every user was simulated to query system 40 times during the experiment.

By using different techniques i.e. laboratory users, yelp user's actual reviews and simulation this work has constructed three experimental models. Credibility of public evaluation score generated by our system is ensured by gathering data from most credible business site YELP and by considering Google score. Whereas, POLS [1] framework has only utilized laboratory users in their work.

VII. EXPERIMENTAL EVALUATION

The proposed system is evaluated by using ROC curve and NDCG performance measures. Area under the receiver operating characteristic (ROC) is a metric to find the ratio between false positive rate (FPR) and true positive rates (TPR) in user preferences provided by the system. Furthermore, normalized discounted cumulative gain (NDCG) is utilized to measure the accuracy of ranking algorithm while evaluating the effectiveness of PoI framework. Whenever a user sends a query to the system, system returns a list of ranked points in descending order. A user may evaluate each point in the ranked list in the form of feedback. Gain is calculated cumulatively from the top of the result list to the bottom of the result list. If any point has previously higher rank and user assigns it a low score, then relevance score will be low. NDCG score represents the ranking algorithm's accuracy.

To find out the accuracy of the system; that how system correctly calculated positive points among all positive points, here positive means: points that are fulfilling criteria of user, and how system incorrectly showed points into ranking list which were not fulfilling the criteria of user means: that a user wanted to find out shopping malls that are open between 10 PM to 12 AM but system incorrectly showed those shopping malls which were closed. This behavior is evaluated using ROC curve by drawing graphical plot by the fraction of TPR and the fraction of FPR. ROC is also termed as relative operating characteristic curve because it shows a comparison between two operating characteristics TPR and FPR.

A. IMPACT OF RANKING FACTORS WITHOUT USING GOOGLE RANKING APPROACH

This section attempts to find out the best combination of following three primary ranking factors utilized for calculating the score of points: (i) weight of public evaluation denoted by α (ii) weight of point's distance from user's location denoted by β and (iii) weight of user's preferences about point denoted by γ. These weights have been elaborated in Equation 1 of Section 4. We carried out an experiment by taking into account users feedback on top ranking 7 points and measured using NDCG. Figure 2 shows that by just considering public evaluation system have shown worst ranking accuracy. If the system considered only distance or personal preferences, then system ranking accuracy is less than 0.76. However, when these two factors are considered together instead of anyone then system gave better output. System NDCG comparison has a higher accuracy of 0.91 when all three parameters are considered together but with double emphasis on personal preferences factor. It is found that system has higher accuracy of 0.97

when considered 1:2:2 as the weighted combination of three ranking factors.

These results showed that a point which has a higher public evaluation does not mean that this point will be preferred by the user. So, we choose a weighted combination of 1:2:2 and normalized all weights to 1 for following experiments as α is set to 0.2, β is set to 0.4 and γ is set to 0.4 so that α: β: γ=1:2:2 and α+ β+ γ=1.

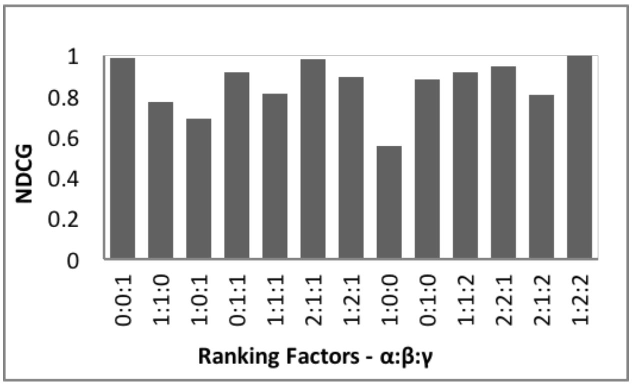

Figure 2: NDCG score without using Google ranking

B. IMPACT OF RANKING FACTORS USING GOOGLE RANKING APPROACH

To enhance the credibility of public evaluation, Google ranking weights are added to public evaluations.

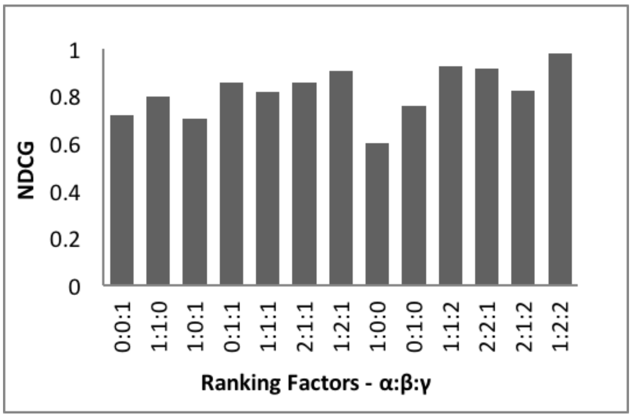

Figure 3: NDCG score with Google ranking

Hence, this approach divided the public evaluation ranking into two categories: Google ranking scores and public evaluation scores. Two weights are assigned to both ranking approaches α:β=1, where α=0.6 weight of public evaluations and β=0.4 weight of Google evaluation. Results of the experiment are shown in Figure 3. It can be seen that system showed improvement in accuracy of ranking results.

C. IMPACT OF GOOGLE RANKING FACTOR AND PUBLIC EVALUATION FACTOR ON RANKING EFFICIENCY

While calculating the ranking score for each searched point, our system divided public evaluation into two categories: public evaluation and Google ranking based evaluation. Each evaluation technique is assigned a weight and final score is calculated by multiplying each evaluation score with a respective weight value. What should be the optimal weight for each category is based on continuous feedback effect analysis by using NDCG evaluation measure. Analysis started by first assigning public evaluation α =0.1 and Google score β=0.9 by ensuring α+ β=1. We increased evaluation score by 0.1 and decreased Google score by 0.1 to analyze the effect of public evaluation on ranking results and user feedback. It is observed that depending more on Google ranking factor was not satisfying user and the same was for when public score was given more than 60% weight. System showed stable behavior with increased ranking efficiency with α=0.6 and β =0.4 values. Figure 4 is showing NDCG accuracy.

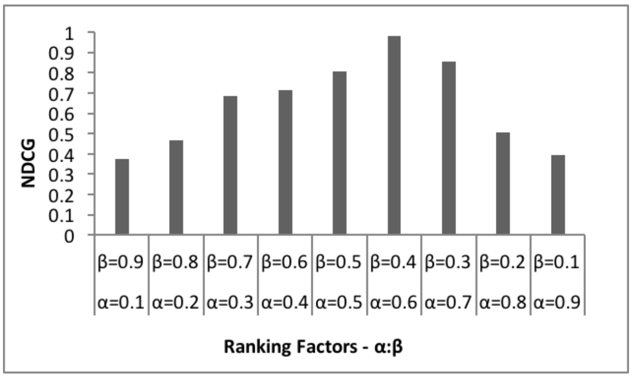

Figure 4: NDCG comparison between α and β

D. IMPACT OF NOISE RATING ON RANKING ACCURACY

To find out the behavior of system on different density of noise we simulated noise rate to analyze the noise rating of the system. To measure the accuracy of system it is evaluated using both NDCG and ROC area as shown in Figures 5, 6 and 7 respectively. We found out that NDCG and ROC area showed a continuous similar behavior for noise ratings. Accuracy of the system increases with a decrease in noise rating and decreases with increase in noise rating.

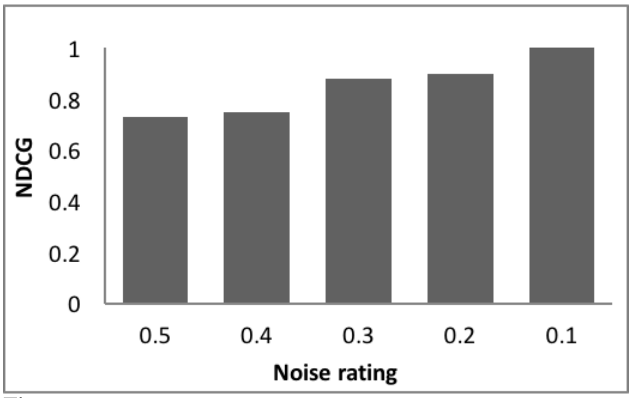

Figure 5: NDCG comparison between noise and accuracy of system

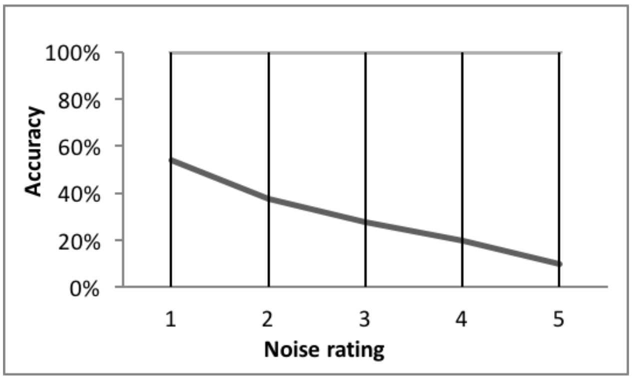

Figure 6: Comparison between noise and accuracy of system using ROC analysis of TPR rate

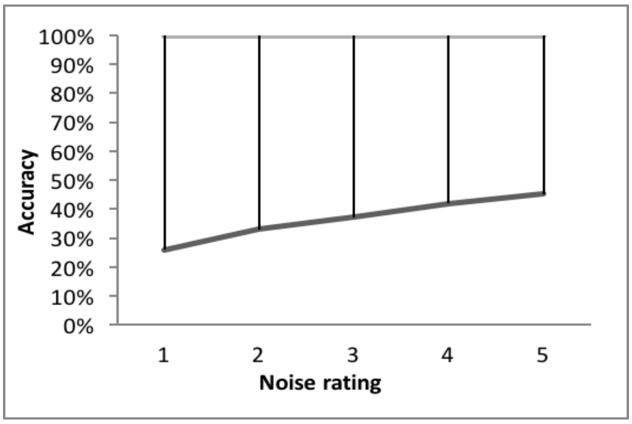

Figure 7: Comparison between noise and accuracy of system using ROC analysis of FPR rate

E. COMPARISON OF USER TO USER CF TECHNIQUE WITH ITEM TO ITEM CF TECHNIQUE

To analyze the performance of both techniques, in terms of time taken to find out similar users, we performed simulation experiments. Simulation is carried out on data of 0.389 million users with 1.5 million reviews for 61 thousand PoI's. Both algorithms were executed on the same number of users and points for four times and a comparison of both methods is shown in Figure 8. It can be observed that Item to Item technique takes comparably less number of minutes to find out similar users and to recommend points to a similar user.

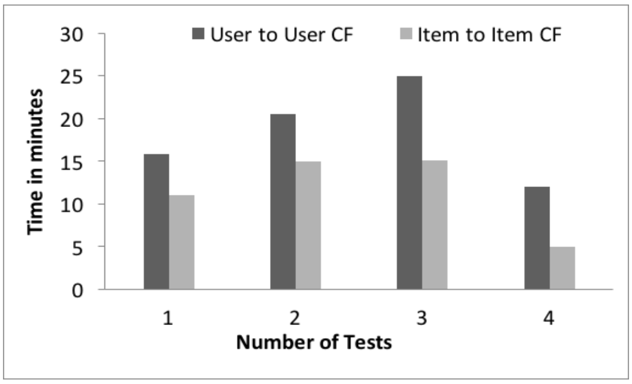

Figure 8: Comparison of "User to User CF" technique with "Item to Item CF" technique

VIII. CONCLUSIONS AND FUTURE WORK

This work presented a hybrid preference oriented collaborative filtering technique for context-aware PoI searching problem. Feedback learning, collaborative filtering and Google page rank are symbiotically harnessed to achieve enhanced predictive accuracy and performance of the proposed system. Unlike the standard context-aware recommender systems, the proposed system exploits user preferences, current time and physical location to return the most favorable K-nearest points. The propose of this research was to develop, test and validate a variant of collaborative filtering towards evaluating PoI search queries. To the best of our knowledge, none of the existing approaches provides a holistic solution where the criteria mentioned earlier are considered simultaneously for answering PoI search queries. This is the novel aspect of the proposed hybrid approach presented in this paper. To evaluate the predictive accuracy, real usage data from yelp.com, comprising 366,715 users, 783 product categories, 889,964 product attributes and a total of 1,566,000 product reviews were analyzed. The performance of the proposed system is compared with that of the existing POLS system.

The evaluation criteria were based on user-to-user and item-to-item recommendation tasks. The proposed hybrid system effectively learned these recommendations by synergistically utilizing feedback learning algorithm and improved CF algorithm. The experimentation revealed that the proposed system outperforms the existing POLS system under varying configurations. The resulting recommendations of favorable location results were eventually sorted using the Google page rank algorithm. Popularity index of nearby users is considered in case of missing data. This feature has substantially improved the predictive accuracy for under certain and missing data situations. The predictive accuracy of traditional recommendation systems suffers largely due to missing and incomplete data. The experiments revealed that the proposed system exhibits best predictive performance with low computation cost. In future research, we will focus on enhancement of the proposed hybrid system by incorporating negative feedback and qualitative learning using modern text analytics approaches.

REFERENCES:

[1] J. S. F. Tan, E. H. C. Lu, and V. S. Tseng, "Preference-oriented mining techniques for location-based store search", Knowledge and Information Systems, vol. 34, pp. 147-169, 2013.

[2] Y. Zheng, Y. Chen, X. Xie, and W.-Y. Ma, "GeoLife2.0: a location-based social networking service", in Mobile Data Management: Systems, Services and Middleware, 2009, pp. 357-358.

[3] I. Toma, Y. Ding, K. Chalermsook, E. Simperl, and D. Fensel, "Utilizing Web2. 0 in web service ranking", in Third International Conference on Digital Society, 2009, pp. 174-179.

[4] N. J. Belkin, "Helping people find what they don't know", Communications of the ACM, vol. 43, pp. 58-61, 2000.

[5] T. Horozov, N. Narasimhan, and V. Vasudevan, "Using location for personalized POI recommendations in mobile environments", in International symposium on Applications and the Internet, 2006, pp. 6 -12.

[6] R. Jin, L. Si, C. Zhai, and J. Callan, "Collaborative filtering with decoupled models for preferences and ratings", in Proceedings of the twelfth international conference on Information and knowledge management, 2003, pp. 309-316.

[7] R. Logesh, V. Subramaniyaswamy, V. Vijayakumar, and X. Li, "Efficient User Profiling Based Intelligent Travel Recommender System for Individual and Group of Users", Mobile Networks and Applications, pp. 1-16, 2018.

[8] C. C. Tuan, C. F. Hung, and Z. H. Wu, "Collaborative location recommendations with dynamic time periods", Pervasive and Mobile Computing, vol. 35, pp. 1-14, 2017.

[9] F. Zhao, F. Yan, H. Jin, L. T. Yang, and C. Yu, "Personalized mobile searching approach based on combining content-based filtering and collaborative filtering", IEEE Systems Journal, vol. 11, pp. 324-332, 2017.

[10] Y. Ying, L. Chen, and G. Chen, "A temporal-aware POI recommendation system using context-aware tensor decomposition and weighted HITS", Neurocomputing, vol. 242, pp. 195-205, 2017.

[11] D. Lian, Y. Ge, F. Zhang, N. J. Yuan, X. Xie, T. Zhou, "Scalable Content-Aware Collaborative Filtering for Location Recommendation", IEEE Transactions on Knowledge and Data Engineering, pp. 1122-1135, 2018.

[12] J. Zeng, X. He, Y. Li, J. Wen, and W. Zhou, "A Point-of-Interest recommendation method using user similarity", in Web Intelligence, 2018, pp. 105-112.

[13] Y. Rui, T. S. Huang, M. Ortega, and S. Mehrotra, "Relevance feedback: a power tool for interactive content-based image retrieval", IEEE Transactions on circuits and systems for video technology, vol. 8, pp. 644-655, 1998.

[14] L. Cai, J. Xu, J. Liu, and T. Pei, "Integrating spatial and temporal contexts into a factorization model for POI recommendation", International Journal of Geographical Information Science, vol. 32, pp. 524-546, 2018.

[15] L. Zhu, C. Xu, J. Guan, and H. Zhang, "SEM-PPA: a semantical pattern and preference-aware service mining method for personalized point of interest recommendation", Journal of Network and Computer Applications, vol. 82, pp. 35-46, 2017.

[16] T. De Pessemier, S. Dooms, and L. Martens, "Context-aware recommendations through context and activity recognition in a mobile environment", Multimedia Tools and Applications, vol. 72, pp. 2925-2948, 2014.

[17] N. N. Liu and Q. Yang, "Eigenrank: a ranking-oriented approach to collaborative filtering", in Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, 2008, pp. 83-90.

[18] Z. Bahramian, R. Ali Abbaspour, and C. Claramunt, "A Cold Start Context-Aware Recommender System for Tour Planning Using Artificial Neural Network and Case Based Reasoning", Mobile Information Systems, vol. 2017, pp. 1-18, 2017.

[19] I. Ben Sassi, S. Mellouli, and S. Ben Yahia, "Context-aware recommender systems in mobile environment", Information Systems, vol. 72, pp. 27-61, 2017.

[20] D. Muhlestein and S. Lim, "Online learning with social computing based interest sharing", Knowledge and Information Systems, vol. 26, pp. 31-58, 2011.

[21] R. Dewri and R. Thurimella, "Exploiting service similarity for privacy in location-based search queries", IEEE Transactions on Parallel and Distributed Systems, vol. 25, pp. 374-383, 2014.

[22] X. Zhang and H. Y. Bae, "Location positioning and privacy preservation methods in location-based service", International Journal of Security and Its Applications, vol. 9, pp. 41-52, 2015.

[23] W. Zhang, S. Guo, and S. Zhang, "Personalized manufacturing service recommendation using semantics-based collaborative filtering", Concurrent Engineering, vol. 23, pp. 166-179, 2015.

[24] G. Gültekin and O. Bayat, "Smart location-based mobile shopping Android application", Journal of Computer and Communications, vol. 2, pp. 54-63, 2014.
[25] S. M. Daneshmand, A. Javari, S. E. Abtahi, and M. Jalili, "A time-aware recommender system based on dependency network of items", The Computer Journal, vol. 58, pp. 1955-1966, 2015.
[26] Y. Teng, X. Lin, L. He, and J. Song, "A Novel Recommender Method in Collaborative Tagging System Based on Time Sensitive Topic Recommendation", Energy Procedia, vol. 13, pp. 4375-4382, 2011.
[27] H. R. Zhang, F. Min, X. He, and Y. Y. Xu, "A hybrid recommender system based on user-recommender interaction", Mathematical Problems in Engineering, vol. 2015, pp. 1-11 2015.
[28] F. Santos, A. Almeida, C. Martins, R. Gonçalves, and J. Martins, "Using POI functionality and accessibility levels for delivering personalized tourism recommendations", Computers, Environment and Urban Systems, 2017.
[29] C. Cheng, H. Yang, I. King, and M. R. Lyu, "A unified point-of-interest recommendation framework in location-based social networks", ACM Transactions on Intelligent Systems and Technology (TIST), vol. 8, p. 1-10, 2016.
[30] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms", in Proceedings of the 10th international conference on World Wide Web, 2001, pp. 285-295.
[31] Y. X. Zhu, J. Huang, Z. K. Zhang, Q. M. Zhang, T. Zhou, and Y. Y. Ahn, "Geography and similarity of regional cuisines in China", PloS one, vol. 8, p. e79161, 2013.