Preference Oriented Mining Techniques for

Location based Point Search

S M Nazmuz Sakib
Graduate of BSc in Business Studies
School of Business And Trade
Pilatusstrasse 6003, 6003 Luzern, Switzerland
sakibpedia@gmail.com

Student of Department of Law
Dhaka International University
House # 4, Road # 1, Block - F, Dhaka 1213
sakibpedia@students.diu.ac

Abstract

With the development of internet and wireless technologies, location based search is among the most discussed topic in current era. To address issues of location based search a lot of research has been done but it mainly focused on the specific aspects of the domain like most of the studies focused, on the search of nearby restaurants, shopping malls, hospitals, stores etc., by utilizing location of users as searching criteria. Problem with these studies is that users might not be satisfied by their results and the sole reason behind this might be the absence of user preferences in the search criteria. There exists some studies which focused user preferences along with user location and query time and proposed some frameworks but they are only limited to stores and their research cannot be scaled to other points like schools, hospitals, doctors, petrol pumps, gas station etc. Moreover there exist scalability issues in their recommended algorithms along with some data credibility issues in their public evaluations strategies. Our proposed research is going to present a novel location based searching technique not only for stores but for any point. The presented solution has overcome issues faced in previous research studies and possesses capability to search for “K” nearest points which are most preferable by user, by utilizing searching time as well as query location. Our research has proposed two feedback learning algorithms and one ranking algorithm. To increase the credibility of public evaluation score, system have utilized Google ranking approach while calculating the score of the point. To make user recommendations nonvolatile along with improving recommendations algorithm efficiency, proposed system have introduced item to item collaborative filtering algorithm. Through experimental evaluations on real dataset of yelp.com presented research have shown significant gain in performance and accuracy.

Keywords: Data Mining, Data Mining Techniques, Oriented data mining, Data Science, Data Analytics

Chapter 1: Introduction

This chapter aims to focus on main problems, will discuss about aims and objectives of this research. At the end of this chapter will describe the division of research work.

1.1 Location Based Search Problems

As internet chipped into our lives, it enabled users to search for anything of their desire using

search engines like Google, Yahoo and Bing etc. Searching anything demands some mechanism, to search nearby locations, by considering user's location. With the development of wireless technologies and the popularity of smart devices, like GPS enabled devices, mobile phones and PDA phones, have increased popularity of location based services. Now a days one of the popular topics of location based services is that a mobile user can search anything using mobile gadgets from anywhere at any time. Currently there are number of searching platforms like

POLS [1] which is a complete framework for location based searching, Geolife [2], IPeen [3], MapQuest [4], Google maps [5], PAPAGO [6] and Starbucks [7] having capability of searching nearby locations. Their searching results depend upon user's current query location. With the development of Web2.0 technologies [8], people have made their business information, like business hours, location, availability and products along with their features online. Using point's information like this, 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 that 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 understand and elaborate the importance of this topic, a scenario hops in supposing Jonson as a traveler who is visiting a specific city first time in his life and he feels hungry. He wants to find a restaurant nearby which comes into the price range of his preference and desire at that time, but he has no idea what are the multiple choices he got.

To achieve the required task there are lot of ways , like using Google searching system or by using restaurant recommendation websites but there are some problems in these systems. First of all Jonson will have to provide his current location to that search engine but because of the absence of any landmark, it's difficult to identify his current location. Secondly, suppose if recommender system identifies his position and shows nearby restaurants, even then the quality of restaurant may not be guaranteed. There are some websites [3, 6] 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 Jonson's. Therefore, for best quality location based search, user preference is required with his location.

There exists a research work [1] which covers all the discussed issues but that work is specific to stores only, they utilized qualitative based feedback learning and their CF algorithm is not scalable due to user based collaborative filtering techniques. So the main problem, which this research will address is using user's current

location and query time develop a system framework, which can accurately rank nearby searched points liked by users. Whenever user queries, system will be able to return more accurate results in a short span of time.

A pile of work exists on this topic and existing systems [9, 10, 11] which helped establishing capacity to gather preferred points. These traditional recommender systems still encounter many problems like lack of user location. This problem indicates that system can recommend wrong points to the user because it doesn't have user's current location. For example Jonson, who is currently in Sydney, wants to search restaurant nearer to him but system shows him list of restaurants from New York City because it is unable to locate him. Second problem with the traditional recommenders is that while ranking searched points, systems are not taking user temporal information into account; this indicates that recommender may recommend stores or restaurants which are closed while user wants to find some point which is still opened. Third problem signifies lack of user preference information and give an understanding that recommender system cannot calculate user preferences for searched point. For example, user wants to search for a restaurant with economical tariff while the results shown by the systems don't fall in his desired range.

In addition to this, a number of research studies [2, 9] have been done for location based search approach and are very successful, but the main problem is that, using user's current location and query time, how to rank nearby point. There are platforms which rank points [5, 6] based on distance between user and searched point. Some platforms used public evaluation strategy [3] to rank points but none of them takes user preferences into account. Moreover to increase the accuracy, precision of ranking results two mechanisms are being used which are: relevant feedback [12] and collaborative feedback [10, 11] but none of them have applied both mechanisms with user preferences. There is only one work (POLS) [1] which takes user preference learning, relevant feedback and collaborative feedback learning approaches into account simultaneously but it's specific to the stores only. The study utilized only positive preferences and their CF algorithm is not scalable too due to user based collaborative filtering techniques. None of the above frameworks have taken Google ranking approach into account.

1.2 Aims and Objectives

Our aim in this research is to develop a system or framework which has capability to use user query time, distance and preferences and can have capability to search for any point which is preferred by user. System must have capability to search for any point and must have capability to utilize most credible public evaluation techniques with enhanced ranking approaches. System must be scalable for multiple users and businesses its collaborative filtering techniques recommendations should be non-volatile and scalable.

To fulfill this aim several objectives have to be achieved.

- Collection of data set of points and users.
- Extraction of public evaluation data of points from recommendation websites.
- Development of searching module using ranking algorithm with embedded Google ranking approach.

- Development of feedback learning algorithm to learn user preferences.
- Development of collaborative filtering algorithms to find out similar users and potential preferences of user.
- Measure the effectiveness of optimal solution.

1.3 Research Question

Therefore research question is “How to develop an optimized location based point searching system under distance, time and user preferences constraints and collaborative filtering constraints?”.

1.3.1 Distance Constraint

Distance constraint is applied while matching route to find a point that travel shorter distance from a set of available routes. There exists more than one path to travel from a source to destination. User may want to travel on the shorter path towards their destination.

Figure 1-1 Multiple ways from Source to Destination

1.3.2 Time Constraint

Time is a very important factor which needs to be defined accurately. Users and businesses should declare their schedules. In most of location based searching applications users define their time in which they can travel how much distance to find a specific point. For example user can have different preferences of distance at different slot of time user may travel

12 kilometers at morning to find a gas station but can only travel 3 kilometers in evening. This limited information is a barrier in finding an optimal solution. A time window needs to be declared. We have utilized time constraint to define preferences.

1.3.3 Preference Constraint

Preference is a very important factor which needs to be utilized accurately. Like time

users can have different preferences about points/businesses at different slot of time so based on this criteria our system should be intelligent to filter out those businesses by giving lower ranking score which are not preferred by user. For example user may want big shopping stores in evening and small with lower price in morning. In our system we have utilized this constraint to filter out most preferred businesses.

1.3.4 Collaborative Filtering Constraint

Collaborative filtering is very popular technique to make location based systems intelligent. It's a way to find out relationship between preferences of different users and services. It's very challenging for any system to apply correct collaborative filtering technique based on data model because of different properties in data model every system needs to design and use different technique. Right collaborative filtering approach leads to right recommendation approach. System performance and scalability directly depends upon usage of right CF approach. This research is focused to find out lacks in existing techniques and utilize right CF algorithm to design optimized system.

Other than above constraints there are lot more constraints which are being faced in this research. So research statement belongs to some of the constraints discussed below:

- How we will design our model? What will be the indicators of dependent and independent relationships in our model?
- What type of techniques and tests we are going to use in the modeling of replica design?
- How the model is verified?
- How the system might be implemented in a working application?
- How the feedback of the system might be generated?
- How the convenience sampling might be attained to check the users acceptance for the system?
- How we will be handling reliability, testability of the model?
- What statistical scores will be allotted to each entity at the end of the system in the form of ranks?

1.4 Research Overview

This research is using various complementary approaches. An overview of the research is provided in this section.

Introduction

First chapter has described the current location based searching problems. Moreover we will discuss about important concepts of location based search, difference between simple search and preference oriented search, difference between simple search and location based search, feedback learning approaches. Current research work constraints faced while doing this research. At the end we discussed about research question and objectives we achieved in this research.

Literature Review

Chapter two contains review of massive literature on location based searching techniques; ranking algorithms, feedback learning approaches and collaborative feedback learning techniques to identify the deficiencies in the existing ranking and collaborative filtering approaches and algorithms.

Methodology

Chapter three will explain the framework of our research, stages in research methodology, discussed about main terminologies that have used in this research to formulize the research problem, identify main challenges that have been tackled in this research work.

Components of the POLPS System

Chapter four aims to describe the modules of proposed POLPS system. Ranking algorithm utilizing Google ranking approach while calculating score of each point, feedback learning approach for calculating feedback of point given by user, collaborative filtering techniques, user to user collaborative filtering techniques and item to item collaborative filtering techniques.

Dataset

Chapter five aims to briefly specify the creation of realistic POLPS dataset by identifying the need for developing a dataset of Pakistan city

Lahore. Furthermore, it will explain the process of extracting GPS location longitude and latitude of each point by reading specific location based attributes from GPS log file.

Implementation of the Prototype System

Chapter six aims to present the implementation of the components of POLPS system in form of prototype website. It demonstrated the requirement specification (i.e. stakeholders, functional and non-functional requirements), detailed architecture along with the modules of the prototype system.

Results and Analysis

Chapter seven will analyze the results of the conducted simulation study. Moreover it will

compare proposed prototype system results with existing system to measure the effectiveness of the system.

Conclusion and Future Work

Chapter eight will summarize the conclusion of the whole research and will provide the future directions for research in this domain.

This chapter have discussed about proposed system framework, features of system and constraints that have been faced during research work. In next chapter will go through some of the existing research studies related to this research work, their pros and cons.

Chapter 2: Literature Review

This chapter will discuss some previous studies, which can be categorized into six categories. (1)Global Positioning System, (2) Web 2.0, (3) Location based Search, (4) Ranking Algorithm (5) Feedback Learning System (6) Recommendation System.

2.1 Global Positioning System

GPS is a service provided by Google which is used to measure the distance of one point from other point on earth. Using GPS user can create locations on the map and can trace it using GPS services. In early days system was developed for military usages but in 1980 Google launches for civilians also, because of its usages in every device like mobile, PDA etc. GPS has become very popular now days. This is the reason GPS based applications are very common in popular. Google Maps is one of the most popular applications. Google introduce its programming interface first time in 2005, which led towards another application which is MapQuest which is same as Google Maps. MapQuest [15] is free online web service owned by AOL. GPS revolutionized web applications and introduced another technology known as web 2.0 technology.

2.2 Web 2.0

Web 2.0 is a modern term which describes the modern usages of World Wide and usage of web designing to increase creativity, enhanced

security of information sharing with increased collaboration and improved usage of web. This concept led towards development of modern web based communities and hostage services because it introduced lots of social and networking web sites like Facebook, MySpace, twitter and LinkedIn, sites for video sharing (YouTube, Daily motion ,YTPak) and sites for blogs etc. Although we2.0 introduced the world with a new face of World Wide Web but it does not means there are changes in technical aspects of it. Web2.0 is representation of various terms now used to describe a variety of developments on the web and there is much improvement in the way it is being used. Web2.0 can be characterized as the evolution in the usage of web from active participation, creation and sharing. This methodology was being used but a little until Google and Google spinoff Gmail started using it a few years ago. Google maps utilized a key feature of this technology from showing results into the browser to dragging contents into the browser window. Utilizing partial page rendering feature of this technology the map is continuously refreshed with the server, that why it does not need to reload a page to fetch changes from server. Other than Google maps this technology introduced many new applications on the web. Like Flickr [12] was one of the very early adaptors of this technology, it is a web site that

allows users to store photos share and use their pictures in an interactive way on the web. In addition to social web like Facebook, MySpace and DIGG web2.0 technology also led to other amazing web applications such as Web Ipeen [14] in Taiwan and Zameen.com in Pakistan. Due to lot of studies on this topic there are a lots of such Web applications on Internet which stores content information of different businesses such as store location, characteristics, business hours, public evaluation score, price level, and user commentary, which can be very useful for searching engine to mine and get optimal search result.

2.3 Location based Search

Location based search is a way for information and entertainment which users can access through mobile devices ,by utilizing the mobile network and utilizing the ability to make use of GPS service which detects the geographical position of the mobile device. Users can utilize location based search in a variety of ways like utilizing in the field of health, work, and personal life. Location based search includes methodologies to find out a location for a person, like discovering the nearby doctor or nearby shopping mall. LBS services can be utilized in the field of investigation to track mobile to track parcel and to find out vehicle using vehicle tracking services. LBS can be utilized to find out personalized weather conditions. In recent years, lots of researches have been done in location based services. In [2] 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 real world, people try to access sequence of location and generate many trajectories in a form of GPS logs, based on these GPS logs, we can build three graphs: a location-location graph, a user- location graph, and a useruser graph. Takeuchi et al. proposed a shop recommendation system based on individual user preferences and need. 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.

In [1] proposed Preference oriented data mining techniques for location based store search which takes user preferences into account for searching nearby store search.

However, the above studies aren't valid for all types of points like searching nearby hospital based on user preferences and none of them

introduce Google ranking algorithm to rank nearby points.

2.4 Ranking Algorithm

Ranking of query results is one of the most fundamental problems in information retrieval. In information retrieval ranking function is a system which ranks information according to its relevance in query. Ranking function is mostly used by search engines.

The articles, topics or items which are relevant to each other are ranked top of the list.

Normally ranking is done by calculating score of each item by using its attributes values. There are varieties of means to evaluate ranking functions but most popular is ranking top

“k” items from some fixed “k” items. Mostly, the computation of ranking functions is understood by following observation about the concept: that only the relative order of the scores matters not their absolute value, the proportion of the top “k” results that are relevant on average over many queries. Many researches introduced different flavors of ranking algorithms in [8] k proposed an approach for web service ranking. In [1] proposed item ranking approach for hotels/stores which ranked stores based of preferences of similar users. One of the most used and popular example of ranking algorithm is page rank. This is an algorithm used by Google search to rank websites in their search engine results. PageRank is a way of measuring the importance of the website pages. Other examples are CRR which was introduced in 2010 which is point wise and pairwise Combined Regression and ranking algorithm. Bayes Rank 2009 which is a list wise method combines Plackett-Luce Model and neural network to reduce the expected Bayes risk, related to NDCG calculations, from the decision-making aspects. In [1] have presented a ranking function which was utilizing public evaluation, distance and user feedback score to calculate score of certain stores, problem with this approach is that their ranking function is only utilizing public evaluation which decreases the credibility of ranking approach. Public evaluations were collected from restaurant recommendation websites which is not credible way to collect data and use it as public feedback. Google has the power to rank businesses by giving some scores based on popularity of that business over the wide area network. Google is using page rank algorithm to calculate rank of each website/business like Facebook has rank of 9, Google itself has rank of 9 and LinkedIn holds a rank total of 8. That rank shows that how much that business is popular and it gives us an abstract

overview of business credibility along with its usage. In our research we are going to improve ranking algorithm by implementing hybrid approach to calculate ranking of points not only stores but approximately all points, by introducing Google ranking approach to calculate score of each point.

2.5 Feedback Learning System

Feedback learning system and collaborative filtering are techniques which normally are used to increase the precision of ranking results; now a day both techniques have been utilized in searching systems. Related to relevance feedback there exists lots of studies in [3, 8] Rui Proposed an approach which is an interactive approach utilized for the retrieval in the content based image retrieval. User can rate any point which has been show in ranking item list. Now a days Relevance feedback based approach is being used to improve the quality of the new query. In [12] proposed a feedback trust model for using feedback effects. There are many feedback based models to increase the accuracy of the algorithm. In [14] authors suggested a new framework using collaborative filtering techniques. Collaborative filtering is used to find out similarity between two users based on their rating results. Those users whose preferences are same are considered as similar users and their preferences are used to rank list of items for the other user. In [1] author has utilized feedback learning algorithms which are learning feedback based on predefined learning mechanism. Problem with existing algorithms is that they are only utilizing quantitative feedback learning approach which is not a credible approach to analyze user feedback while users qualitative feedback can also be considered to analyze either user feedback is positive or negative. More over similar user finding algorithms are only finding similarity based on positive feedback. Negative feedback also can help to find out the deficiencies in search points.

2.6 Recommendation System

Recommender systems [3] normally termed as recommendation engines which works using a specific technique of information filtering system and helps to present things like images, videos, places, books to user which are new and have not rated them before. How recommender systems works they normally compares the items based on specific characteristics and try to predict a rating for an item that most probably user will give to that item . Now based on this information

we can say that main purpose of recommender system is to give suggestion of new things or seek out the usages of some item for specific user by showing that item in recommendation list of that user.

Lots of recommendation systems are working and utilizing collaborative filtering approach. In [10] Horozov et al worked on location based point of interest systems and investigated some issues in these systems. This research has proposed an enhanced CFbased collaborative filtering techniques and following it proposed a restaurant recommender system. In [15] proposed new methodology for online learning of social computing based interest sharing. This research proposed a model for users using internet that allow them to discover their common points of interest like sets of URL frequently searched . In [17] proposed a methodology to find out Service Similarity for Privacy by using location based search queries. 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 [18] X. Zhang and H. Bae presented a survey 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 and techniques of location privacy protection. In this paper, author proposed a taxonomy which was to help researchers to quickly understand existing work, challenges and possible improvements.

In [19] 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 [20, 21] proposed a time-aware recommender system utilizing the dependency network of items. In [22] proposed a recommender method in collaborative tagging system by utilizing time sensitive topic recommendation. In [23] proposed a hybrid recommender system which was based on user-recommender interaction. In [32] proposed web services recommendations system using collaborative filtering techniques and in [33, 34, 35] proposed recommendations system ranking. In [36, 37, 38] proposed recommendation systems for analysis of symbolic data. In [39, 40] proposed recommender systems using IOT from trust aware systems.

In [1] have proposed a recommender system by utilizing user to user collaborative

filtering algorithm which calculates similarity of users and recommend stores to the user but due to lots of comparisons between users and volatile calculations it's not considered as scalable algorithm. 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 point of interest.

This chapter described the literature review of research work find out different domains of location based search, discussed about algorithms utilized to solve different problems of this domain. In next chapter will discuss about architecture of the system framework with complete life cycle of each request and will formulize research problem.

Chapter 3: Methodology

This chapter will briefly discuss about the methodology that is used to develop proposed framework. Here will show and discuss about architecture of POLPS framework with detailed description of each component with life cycle of system and define some terminologies which will

be used during this research work after that will specify main research problem.

Provided user's current location and searching time purpose of this research is to develop a system framework that provide nearby points ranking list which are liked by the user.

Figure 3-1 System Architecture

3.1 Framework

Figure 3.1 shows the proposed POLPS system framework. This research work will extend POLS [1] framework and introduce POLPS that will be a novel data mining approach to efficiently search nearby points which are most preferred by user based on user's current location and query time. POLPS will extend POLS to five points (restaurants, shopping malls, petrol pumps, public transportation and doctors) due to different features and diversified nature of businesses POLS approach was unable scale for multiple businesses. POLPS system will consist of two parts client software, server software. Using client software user will send requests to POLPS server software with embedded location data. Server software will further consists of two layers POLPS middleware and POLPS data storage layer. Middleware will intercept initial requests and filter out points based on user preferences and query time after that middleware will run ranking process to fetch related points, calculate their score and sort points according to score. Point with highest score will be top of the list ranking results will be sent back to the user. POLPS will merge Google ranking approach with existing ranking strategy to calculate user's most preferred points because Google ranking results are most credible. In POLPS research work aims to propose personal preference database to store positive and negative preferences of every user. Using this database, system will apply ranking algorithms to rank nearby points using user's current location and searching time and personal preferences. User will give feedback and personal preference database will be updated using feedback learning algorithm. Every user feedback will be stored in a global preference database where system will apply collaborative filtering algorithm [13, 14] to calculate similar users based on positive feedback. CF results will update personal preferences of user by recommending

points which most probably user find informative. Feedback learning algorithms will learn preferences right after receiving user feedback but collaborative feedback learning will be done periodically in intervals. As per our research knowledge this research will be first work of its kind that will integrate location based search, Google ranking approach, feedback learning strategy and user preferences learning. After that experiments will be carried out to calculate POLPS ranking accuracy and performance improvements.

3.2 Problem Formulation

Definition 1 (Preferences/features) $F = \{f_1, f_2, f_3, \dots, f_n\}$ define list of all features, every user will have certain features.

Definition 2 (Users) $U = \{u_1, u_2, u_3, \dots, u_n\}$ define system users which will use system to search points.

Definition 3 (Similar users) $CF = \{u_1, u_2, u_3, \dots, u_n\}$ define list of all users who are similar to other users due to similarity of preferences.

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

Definition 5 (Distances) $Dist = \{dist_1, dist_2, \dots, dist_n\}$ will define distance between searching user and searched point in kilometers ranking system will use $Dist$ to rank points.

Definition 6 (Time slots) $T = \{t_1, t_2, t_3, \dots, t_n\}$ will define list of time slots in a day.

User's searched query will always lie under certain time slot. Every time slot in a day will be unique because user preferences will be different in different time slot. Research work will filter user preferences according to time slot.

Notations Tables

Notation	Description
F	Collection of preferences
U	Collection of users
CF	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
RL	Rating values used in feedback
RD	Rating dates

Table 3-1 Notations Used in POLPS

Definition 7 (Points) $P = \{p_1, p_2, \dots, p_P\}$ define 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) $RV = \{1, 2, 3, 4, 5\}$ will define list of rating values which user will use to rate points, rating values from 1 to 5 shows the preference level will be from not preferred to preferred. Rating values 1 and 2 will show that point is preferred by user. Rating value 3 will show that user neither liked the point and nor disliked and rating value 4 and 5 will show that point is most valuable and preferred by the user. The not preferred level of 1 is higher than 2. The preferred level of 5 is higher than 4.

Definition 9 (Rating dates) $RD = \{date_1, date_2, \dots, date_{|Date|}\}$ will define list of dates in

which user rated the points. Rating dates will may be different from query date because there may be chance that user will rate point days after query.

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 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 liked by current user. Moreover it is expected that after learning in a short span of time system will return more accurate ranking list. Therefore research question is “How to develop an optimized location based point searching system under distance, time and user preferences constraints?”. To answer this question and to

fulfill aims and objectives mentioned above following challenges need to be tackled.

Challenge 1

- Collection of data set of points and users.
- Extraction of public evaluation data of points from recommendation websites.

Challenge 2

- Development of searching module using ranking algorithm with embedded Google ranking approach.
- Development of feedback learning algorithm to learn user preferences.

- Development of collaborative filtering algorithms to find out similar users and potential preferences of user.

Challenge 3

- Evaluation of results.
 - o Measure the effectiveness of optimal solution.

This chapter discussed about proposed framework its components and discussed about different definitions which have used to define key actors and features of the system. Challenges which need to be solved for implementation of this research work. Next chapter will discuss about different components which have been used to build a system framework.

Chapter 4: Components of POLPS

This chapter aims to discuss about the different components used to build POLPS framework. It will start from illustration of ranking algorithm, and then will describe feedback learning process after that will discuss about collaborative filtering techniques that have utilized to design a system framework.

4.1 Ranking Algorithm

Ranking algorithm purpose will be to provide a precise and accurate list

of ranked points. It will prioritize points using public evaluation, user's preferences and current location. In previous research studies [2, 9, 10] researchers only discussed one or two factors to rank points and none of the research study considered user's personal preferences to find out the point which is preferred by user. So whenever user will query before applying the ranking algorithm, system will filter all those points that are closed and will filter by comparing their query time and business hours. System will give score to the remaining points by using scoring function. 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 score of point p will be defined as

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

So Equation 1 will be the sum of three scores which will be public evaluation, the distance score and preference matching score. Here α , β and γ will be weights of all three scores such that 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 yelp. To increase the credibility of public evaluation POLPS have extended public evaluation score into two sub scores: one calculated from businesses recommendation websites and other calculated by Google search engine. So based on NDCG analysis system has given 60% weight to public evolution and 40% weight to Google search engine score. Assuming user u with searched point p scoring function to calculate public evolution score will be.

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

Here α value is weight of public evaluation with value 0.6 and β is weight of Google search engine with value 0.4. Whole score is normalized from 0 to 1. To calculate score of distance first we have to

understand its concept. Concept of distance is based on user behavior. So system calculates preferred distance using rating historical logs of user, which might be preferred distance from the actual user to point.

$$distPref(u, p) = \frac{\sum_{j=1}^n distAct_j}{n} \quad (3)$$

□

Here $distAct(u, p)$ is the actual distance between user u and point p in kilo-meter, and n is the number of ratings which user u have given. Moreover, by imaging the concept that point will be marked as not preferred if there exists some other point closer to user under same category with same features as of user is searching like an oil exchange station having credit card facility so we designed the distance score $SDist(u, p)$. To get a lower distance score for those points that are further we have designed equation 4 if the distance is over the average preference distance in equation shown in equation 3. On the contrary, points which are close enough that their actual distance is nearly equal or within the average preference distance system will give them highest score 1.

$$SDist(u, p) = \begin{cases} 1 & \text{if } distAct(u, p) \leq distPref(u, p) \\ 1 - \frac{distAct(u, p) - distPref(u, p)}{R} & \text{otherwise} \end{cases} \quad (4)$$

□

where $distAct(u, p)$ 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 POLPS framework for each user in kilometer, for example, 2 km. The tolerable range could be changes based on query time with distance keyword, for example, 0.5 and 3 km.

$$Spref(u, p) = \begin{cases} StrPPu & \text{if } distAct(u, p) \leq distPref(u, p) \\ StrNPU & \text{otherwise} \end{cases} \quad (5)$$

□

Here notable thing is that user preferred distance can be different in different time slots. For example user can prefer closer and cheap restaurants, doctors, shopping malls in the morning due to shortage of time but may prefer high- class and costly restaurants in the evening although restaurants are far. Here $SPref(u, p)$ is the score calculated by matching preferences of user u and the features of point p . will calculate this score by subtracting negative feedback from positive feedback. For each mobile user, there is personal preference database which contains positive and negative preferences. Each record in positive and negative preference database has a preference point its score from 1 to 5 with strength value ranges from 0 to 1. Preferences database contains mixture of both types for example user U_1 personal preference database contains three positive preferences and two negative preferences, a record {cheap, 0.9} in the positive preference means that the user like and prefers cheaper points with great intense. Equation no 5 shows the preference matching score, where F_s shows set of features for point p , PP_u and NPU defined as positive and negative preference rated by user u , respectively and strength values of positive and negative preferences are represented by $StrPP_u$ and $StrNPU$, respectively. If there are two points represented by point A and point B and each contain three point features point may be a store then using above equation the preference matching scores for point A and point B can be calculated as 0.8 and -0.4 respectively.

4.2 Feedback based Algorithm

System will give list of ranked points and user will rate points according to their preferences. To understand the algorithm and to indicate preference level, let's suppose there are five rating values from 1 to 5. Higher value of rating will show that user liked the point more. System will use this algorithm to automatically learn positive and negative preferences and updating the personal preferences of the user.

Using rating values from list {1, 2, 3, 4, 5}, system will learn either user has liked the point or disliked. If rating value is 4 or 5 system will understand that user liked the point, if rating value will be 1 or 2 it means point is disliked by the user and rating value 3 means ordinary. Considering feedback of the points, system will use 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 its strength will be 0. Feedback 4 and 5 means positive feedback strength values will be 0.5 and 1.0 respectively.

For the points, system will store point rated by user into positive databases if its preference value will be 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. System defined that the corresponding strength value for rating 5 is 1.0. So system will put all features like cheap, parking, credit card allowed, good for children, wheelchair accessible, elevators of that shopping mall into positive preferences of user u with strength value 1.0 and dated Feb

10. Rating of point shows that user “u” liked this shopping mall. There may be chance that user “u” will again go to that shopping mall and again rate it so there should be mechanism to combine both rating into single rating. This system combine them using rating day. Hence, strength value for rating 4 is 0.5. Days passed 10 and 20 from rating day respectively system found that cheap attribute is found two times in positive preferences database with strength value 1 and 0.5. To combine strength value StrN of attribute cheap system will follow below technique.

$$\text{StrN} = (0.5 * 20^{-1}) / (20^{-1} + 10^{-1}) + (1.0 * 10^{-1} / 20^{-1} + 10^{-1}) = 0.66.$$

So according to human nature, by passage of time user preferences will be changed and with the passage of time preference level will be decreased by the number of days have passed. If user will prefer more effect of decay will be decreased and preferences will be remain active. In other words, preferences will be set active if user will rate points for longer time. If user will stop rating point, decay weight will be decreased to 0 and that preference will become inactive. System applied decay function and based on the number of days have passed system decrease each active preference strength value. System will start calculation from day first. So actual decay function can be defined as

$$D(\text{StrN}, dN, i) = \text{StrN} - e^{g(i) \times (dN - c2i)} \quad (6)$$

$$\text{Where } g(i) = 1/1 + \log_{10} i$$

Here StrN is normalized preference strength value of point which is currently active and dN - c2i is control variable, used to control the distance between number of days which have been passed and weight of preference of user it is denoted by parabolic curve and yaxis. Here Dn is number of passed days, system used it to get weight of preference at yaxis, -c2i is factor which is used to optimize the distance between 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 4 g(i) is denoted by gradient rate. By taking all coefficients of equation 4 into account, system will find out any active preference with negative strength which means that rating is past for a long time. So system will start an updating process and will set to inactive in the user preference database.

4.3 Collaborative Filtering CF-based Learning

To learn preferences of users globally we have proposed CF based collaborative filtering algorithm. Algorithm is to learn preferences of a user with respect to other similar user. POLPS have capability to recommend businesses by utilizing personal preferences and potential preferences of the user. To implement this algorithm system has stored reviews of users into rating logs this rating is made by all users. By following review log, instead of analyzing preferences between user and other users, this system analyzed similarity of preferences between item and other items. For the items who have high similar preference sets, system transferred those similar items into the recommendation of all those users who have rated current item.

So to analyze the similarity between preference sets rather than matching the users to similar users, system matched similarity between items to item [14]. In Item to item collaborative filtering technique this algorithm matched each of the user’s rated items to similar items, and then combines those similar items into recommendation list.

Item to item collaborative techniques are beneficial if in terms of both performance and accuracy. Users to user collaborative techniques are beneficial when data sets contain more points than users. In case of POLPS dataset contains 366 thousand users and 61 thousand points. So User to user CF Techniques will be costly and less scalable due to

N*N comparisons between users and their recommendations will be non-static. That's why we have used Item to Item collaborative techniques to find out similarity between users.

How system implemented this algorithm by finding similar matches for a given item so to find out the most similar match, the algorithm first created a table of similar items just by

```

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

```

Figure 4-1 Item to Item Collaborative Filtering Algorithm

It's possible to compute the similarity between two points system have followed cosine similarity [25].

Suppose A and B are two points whose similarity needs to be calculated to add it into the preferences of the user who also rated A. So algorithm used following cosine similarity technique to calculate similarity score of A with B.

$$\text{Similarity}(A,B)=\cos(A,B)=A.B/\|A\|*\|B\|$$

Here |A| and |B| are magnitudes of positive preferences of both vectors.

finding those items which have been rated by both users. So system build a point to point matrix by iterating over all item pairs and computed a similarity matrix for every pair. But there are many point pairs with no common users. System has followed following algorithm to implement item to item collaborative filtering technique.

Point having highest similarity will be considered as most similar point and system will add that point into the preferences of all those users who rated current point.

This chapter have discussed about different components of proposed framework now next chapter will discuss about the datasets used to evaluate system framework. Data set collection techniques with data model building techniques.

Chapter 5: Dataset

This chapter will briefly discuss the creation of realistic POLPS Dataset. Start from illustration of the need of creating a dataset for different points. Afterwards will describe the process of developing a dataset i.e. calculation of GPS location for each point, required equipment and software's, preprocessing of the collected dataset, features and attributes of the dataset etc.

5.1 Need of developing POLPS Dataset

To practically evaluate the proposed preference oriented location based search system, realistic POLPS dataset was collected of different

cities of world including Pakistan specially Lahore. Real data from Pakistan was collected to make system analysis more credible. And to run proposed system in location environment it mandatory to include data from that environment. Why selected dataset from Lahore region is very special question which may come from reviewers. Now are going to justify this question in three points

First its urban area and majority of the people in Pakistan are migrating from rural areas to cities because of exponential growth in population and continuously decreasing land occupancy. According to the UN estimates fifty

percent of Pakistan population has become urbanized. Urban development and planning are a magnanimous challenge for the government of Pakistan. Urban cities development plays an important role in development of country. Cities describe the economic structure, nature of society and culture of the country. According to the author [26] city determines the country place in the global system.

Second because of terrorist activities in Pakistan special in Karachi lots of business men moved their businesses from their cities to Lahore. Due to handsome development within city from last 5 to 6 years Lahore has become an eye catching place for people of Pakistan. Ratio of restaurants, schools, universities, parks, hospitals and even business points has been increased in this city.

People from local cities are moving in this city to make their future bright. Lot of business opportunities have been created in this city from last 10 years. So it was very easy for us to collect data of different points. In our research there was a requirement to collect data which our system can utilize to give more feasible results. So Lahore was most suitable location for us to collect data of different points including doctors, hospitals, shopping malls, restaurants, schools, universities, parks, gas stations etc.

Third this research is done in an institute which is located in Lahore region our research is applied research so it was easy for us to practically test our results. It was easy to gather test clients to generate requests to our system.

Figure 5-1 Lahore Map

5.2 Dataset Creation Process

In this work respondent used following accessories and apps to test POLPS system which are described as follows:

Mobile phone having GPS Feature each respondent have a mobile phone having GPS feature. GPS is a space based satellite navigation system that presents location and time related information in all types of weather conditions. Before starting the trip, respondents enable their GPS option which get connected to the satellite in a few minutes and start giving you location information.

GPS Logger Application To record trajectory i.e. travel attributes latitude, longitude and time between two coordinates , this work uses an Android app “GPS Logger” [27]. This app works efficiently without having internet connectivity .Application consumes less battery as compared to other logging apps. This app record the GPS coordinates at fixed time interval specified by respondents. Before starting trip respondent enable the start logging option and after reaching its destination it disable logging option. App

records the whole travel route coordinates along with time. The recorded file is stored in the SD card folder. This app runs in the background, after start logging, respondent can use their mobile functions easily. This app stores files in GPX or KML format. This is a very user convenient app by allowing user to email a file, ftp upload and can upload traces on open street map.

POI Conversion Tool Recorded GPS traces are in different format depending on manufacturer (i.e. mobile phone manufacture), which can't be directly used for further processing (i.e. searching module).Our retrieved GPS log files are in .GPX format, hence, converted it into another file format by using conversion tools. This work selected the CSV format to create our dataset files. Creating a file in CSV has many advantages, it can be read easily by any spreadsheet program, can export to any other application easily [28]. For converting GPX file into CSV format this work uses POI convertor. POI convertor is an easy to use tool. No hard and fast rules are required to convert GPX file into CSV file. By uploading a file it's easily converted the file into comma delimited CSV format.

Figure 5-2 System Flow of Dataset Creation

5.3 Attributes of POLPS Client Request File

A file consists of set of records, where each record is composed of fields, each containing a particular piece of information[29]. In our dataset each file has the following records:

Client Id Client id is the name of the client, which is numeric in our dataset. Each trajectory file or each recorded trip has a unique id. Recorded trajectory is saved with the id of client. First column of CSV file is filled up with client id.

GPS Coordinates Each file has a list of GPS coordinates i.e. latitude and longitude.

Latitude specifies the north south position of a point on the Earth's surface. It's an angle ranging from 0° at Equator to 90° (North or South) at the

poles [80], while longitude specifies the east west position of a point on east surface. After a fixed interval the logger logs the coordinates of the location.

Time Each record contains time along with latitude and longitude. A fixed time interval is set by the respondent. In this work, fixed the time interval by 1 minute, i.e. after every 1 minute logger will record the coordinates along with time. Starting time will be received at the start of the trip.

To explain the format of file let's take an example, if a respondent start their trip at 09.00 am, logger will save origin location coordinates along with 09.00 am in the following format.

Latitude	Longitude	Time
31.46805	74.24655	09.00

Table 5-1 Sample File Format

After every one minute logger will save the respective location coordinate along with time. In this way whole trajectory is recorded until respondent end their trip. Final trajectory files contain location coordinates (i.e. longitude and latitude), travel time between two geo coordinates. Final trajectory contains 1000 records of the whole route.

Sample of the final trajectory file is shown below:

Client Id	Time	Longitude	Latitude
8754	08.30	74.3559	31.5260
8754	08.31	74.3556	31.5246
8754	08.32	74.3553	31.5237
8754	08.33	74.3551	31.5232
8754	08.34	74.3530	31.5214
8754	08.35	74.3512	31.5209
8754	08.36	74.3497	31.5208

Table 5-2 Sample of Final Client File

At the end of this chapter this work has achieved its fourth objective that was: “*Create a dataset for evaluating proposed system in realistic scenario*”. Lahore GPS ride trajectory dataset is created for evaluating the proposed system in realistic scenario. Dataset is gathered by routine commuters which have a high demand of ridesharing. Dataset is created through an unbiased tool GPS Logger, generated dataset semantic is easily understandable, hence, can be utilized efficiently in future, format of the dataset preserves the syntactical integrity of the elements

of the dataset, therefore, it can be analyzed effectively.

5.4 Attributes of POLPS Businesses File

Other than client business dataset have collected list of different points under different categories. Like client attributes businesses so called points in this research also have GPS locations attached with their attributes and unlike client businesses have their attributes so called properties and features. Each point has different

nature so their attributes are different as compared to points under different categories. This research have utilized real time dataset obtained from yelp.

5.4.1 Attributes of Yelp Dataset

Yelp [30] founded in 2004 is an American multinational corporation headquartered in San Francisco which focus on development of yelp.com solution utilizing mobile apps, collect reviews about businesses and publish those reviews for analysis. By 2016 yelp has 135 million monthly visitors and about 95 million reviews about local businesses.

Along with yelp data set system also utilized dataset from UCI Machine learning repository of Donald Bren School of Information and Computer Sciences University of California, Irvine, this dataset was obtained from a recommender containing 366715 users , 61184 points of 783 different categories with 889964 attributes on average each point contains 15 attributes and with 1566000 reviews.

Initial shape of data was in the form of CSV files to make it useful for computations mapped it from CSV to relational objects shown in figure 5.3 below

Figure 5-3 Relational Model of Yelp [30] Dataset

Point Id	Point Name	Country	Latitude	Longitude	URL	Point Type
134983	Restaurant and Bar	Mexico	18.948657	-99.235361	carlosandcharlies.com	Restaurant
134986	Restaurant Las Mananitas	Mexico	18.928798	-99.239513	lasmananitas.com.mx	Restaurant
134987	Chilis Cuernavaca	Mexico	18.932725	-99.225211	chilis.com.mx	Restaurant
134999	Kiku Cuernavaca	Mexico	18.915421	-99.184871	kikucuernavaca.com.mx	Restaurant
135041	Luna Café	Mexico	22.15106	-100.977659	lunacafe.com.mx	Restaurant

135072	Sushi Itto	Mexico	22.149192	-101.002936	sushi-itto.com.mx	Restaurant
135123	Smart Hotel Lahore	Pakistan	18.875011	-99.2353499	smarthotellahore.com/	Hotel
135124	DR Khurshid	Pakistan	31.4665451	74.2737073	entlahore.com/	Doctor
135125	Dr.Muhammad Fiaz	Pakistan	31.5141163	74.349148	entlahore.com/	Doctor
135126	Dr. Solomon Elahi	Pakistan	31.5752026	74.377818	health.hamariweb.com/Lahore	Doctor
135127	Dr. Aysha Sheihk	Pakistan	31.4805698	74.3853803	health.hamariweb.com/Lahore	Doctor
135128	Dr. S Tahira Izhar	Pakistan	31.5381496	74.3488148	health.hamariweb.com/Lahore	Doctor
135129	Dr. Badar Munir	Pakistan	30.2023601	71.4414767	hhealth.hamariweb.com/multan/	Doctor

Table 5-3 Sample Points data model

Each point has its name, country, latitude longitude to recognize its physical location, type of point, public evaluation score and its business website URL.

Point Id	Start Time	End Time	Days
132561	00:00	23:30	Monday, Tuesday, Wednesday, Thursday, Friday
132562	00:00	23:30	Monday, Tuesday, Wednesday, Thursday, Friday
132563	00:00	23:30	Monday, Tuesday, Wednesday, Thursday, Friday
132564	00:00	23:30	Sunday
132565	07:00	18:00	Monday, Tuesday, Wednesday, Thursday, Friday
132566	07:30	19:00	Monday, Tuesday, Wednesday, Thursday, Friday
132566	07:30	19:00	Sunday

Table 5-4 Point wise Opening and Closing Timings

5.5 Calculation of Google Score

To increase credibility of public evaluation score proposed work have utilized website URL and applied Google ranking algorithm to calculate its popularity on internet our system has added this score with public evaluation score during calculation of ranking of points. To generate data set of users we have taken 20 members of our laboratory to participate in our

experiment. Each user has been given predefined latitude and longitude and each have been given a time slot from predefined timeslots. Then have instructed participants to query POLPS system during different timeslots like from 10:00AM to 12:00PM, 12:00PM to 3:00PM, 3:00PM to 7:00PM, 7:00PM to 10:00 PM and 10:00PM to 12:00AM.

User Id	Name	Latitude	Longitude	User Type
1003	Naeem	22.119847	-100.946527	Student
1004	Hamid	18.867543	-99.183	Professional
1005	Naseem	22.183477	-100.95989	Student
1006	Ijaz	22.158474	-100.983	Student
1007	Mical	22.118464	-100.938256	Professional
1008	Saqib	22.159427	-100.923811	Student
1009	Raheem	22.190889	-100.990448	Student
1010	Ubaid	23.724972	-100.998669	Professional
1011	Kashif	28.813348	-99.152856	Student
1012	Pietersen	22.174624	-100.993873	Professional
1013	Jessica	18.952615	-99.201616	Professional
1014	Noor Ullah	18.878189	-99.222969	Student

Table 5-5 Dataset of Users with GPS Coordinates

So in other words each user queried POLPS system 5 times in a day and continued working for 20 days. For a while we have assumed that each user is under predefined timeslot and location. System applied ranking approach to calculate and given list of top 7 points under searched category. Now participants are requested to give their feedback against each populated point based on their personal preferences under current conditions.

SRNO	Point Name	Score	Point Features	Distance
1	Dr. Aysha Sheikh	-1138.31	Radiologist, Pediatrician	8.547
2	Dr. Solomon Elahi	-1138.71	Radiologist, Pathologist	8.550
3	Dr. Muhammad Fiaz	-1139.15	Radiologist	8.551
4	Dr. S Tahira Izhar	-1139.44	ENT Specialist	8.551
5	Dr. Khurshid	-1139.97	ENT Specialist	8.555
6	Dr. Babar Munir	-1165.31	Pathologist	8.745
7	Dr. Prof. Imran Iqbal	-1165.37	Radiologist, Pediatrician	8.747

Table 5-6 Representing Sample Outcome from POLPS Framework

Table 5.6 showing the result of top doctors near to a user, distance is the distance between user u and point p in meters.

SRNO	Point Name	Score	Point Features	Distance
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 Del Virrey	0.61	Maintenance Staff	11.978
7	Luna Café	0.61	Parking	11.978

Table 5-7 Ranked List of Top Rated Points by POLPS Framework

Above Table showing list of top restaurants near to user u point features are after distance. During the experiment system perceived the rating behavior of user who rate

points based on public evaluation (Public evaluation +Google evaluation), distance of point p from user u and by user preferences.

To analyze performance of POLPS, system enlarged number of points and users by merging yelp users and points which contains real information like features, public evaluation reviews and business hours. For each point this research has calculated Google score if possible. The average number of preferences of user is 6. To generate more and more ratings of points under different categories we have designed a

simulation of POLPS in which created dummy users each user is assigned a location from different random locations and query time from predefined query slots. Every user has its predefined preferences. As compared to POLS our system is quite capable to search points other than stores and restaurants so our research have configured users to search for different points like doctors, hospitals, shopping malls, schools, oil pumps, parks, gas stations etc. user may search by specifying some keywords which will be name of category like hospitals, parks, schools etc. when user send query to POLPS. System will take location information, temporal information and preferences of user into account and is recommended by the ranking algorithm of POLS. After finding results from recommender system every user rate points by giving them rating from 1-5. POLPS feedback learning system learned user ratings and stored into either positive or negative databases based on the value given by the user. 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 point from user. Every user was simulated to query POLPS 40 times in the experiment.

By using different techniques laboratory users, yelp user reviews as feedback and simulation we have constructed three experimental models of dataset. Credibility of public evaluation score generated by our system was ensured by getting data from most credible business site yelp and by considering Google score. POLS framework has only utilized laboratory users, and different recommendation sites work.

This chapter has discussed about data collection methodologies with attributes. Next chapter will describe implementation strategy of above POLPS components. There will show about different technologies that have been utilized to implemented different phases of POLPS framework and implementation life cycle with detailed description of each cycle.

Chapter 6: Implementation of the Prototype System

This chapter aims to focus on components of the proposed POLPS system that have been described in chapter four for the development of a prototype system. This will explain the design, working scenario, functional and non-functional requirements of the prototype system. Architecture of the proposed prototype and different modules of the architecture will be discussed in detail.

6.1 Prototype

To test the viability of POLPS system and to study the merits of the ranking algorithm, feedback learning algorithm and collaborative filtering techniques, developed a prototype website named POLPS system, which will be easily accessible for both passenger and ride giver through the mobile phones to be connected everywhere. The reason for developing a website instead of developing a mobile app is that: mobile application is needed to develop for different mobile phone operating systems. Passengers used different mobile phone models having different operating systems. For example Samsung and Qmobile have ANDROID operating system, I phone has IOS operating system, blackberry has blackberry operating system, Nokia has Symbian operating system etc. Therefore developing an app compatible with all

these operating systems is a tedious work. Hence we developed a website that can be easily accessible from mobile phones, laptops and desktop computers etc.

6.2 Design of the Proposed Prototype

From the found deficiencies through literature review, we are motivated to design a prototype for location based point searching system based on multiple constraints by using open source java, JSF technologies. Our aim is to develop a more convenient, flexible and beneficial dynamic ride sharing system for visitors of Lahore city or any other city. We selected Oracle award winning tool J Developer and ADF for implementation, the rationale behind this tool selection is its features: such as its enhanced UML diagrams designing techniques without writing much code for Use case, class, sequence diagrams, relational database designing techniques, separating business logic from user interface and controller layer and providing multiple views or options to developer for designing and implementation. We have used JSF technology with enhanced ADF Faces API for developing our prototype rational behind choosing JSF instead of other J2EE framework is its very mature framework it contains embedded

html5 and AJAX enabled components with touch support functionality.

6.3 Working Scenario of Proposed Prototype System

To practically evaluate the system on realistic dataset this research have utilized collected dataset of Lahore city for doctors, schools, hospitals, parks, shopping malls with UCI machine learning dataset and yelp as described in chapter 6. As road network is divided into grid cells.

6.4 Requirement Specification of the Prototype

To avoid problems of ambiguities along the development process, defining the requirement in a formal way is necessary. In this section, we describe stakeholders, functional and non-functional requirements of the proposed prototype.

6.4.1 Stakeholders

An important aspect of software development is identifying system stakeholders. Our system stakeholders are general users, who want to search for businesses and users of business men who want to expose their businesses to internet and wants that their businesses will become popular based on public and user feedback. Our target group is Lahore city users and businesses. Moreover we have extended our research to any location so any user who wants to search for specific point/ business in the Lahore city or from any location is our target user

6.4.2 Functional Requirements

Specific statements of service that defines how the system should react to particular inputs and how it should behave in particular situations is called functional requirement. Functional requirements of our proposed prototype website are as follows:

1. *Whenever user submits a request, system should quickly provide an optimal list of points.*

Motivation After submitting request for a business or point, user want to receive the result immediately without wasting time. Therefore, an efficient searching mechanism is required which matches request with the required business by fulfilling all participant constraints instantly. The proposed prototype will implement the searching mechanism for fast retrieval of points / businesses by accomplishing constraints. System will filter out all those points which are not according to preferences of user.

2. *After providing points matches, system should process the feedback provided by user in terms of the incoming request efficiently.*

Motivation after receiving points system will receive feedback from user in terms of score ranges from 1 to 5 for each point. System will process feedback provided by user will identify positive and negative feedback and will store that feedback as positive and negative preferences of user. Here system will use feedback learning approaches.

3. *System should give recommendation to user for points of certain category by learning similarity of users' preferences with respect to other user's preferences.*

Motivation Every user wants to that system should recommend some of the points which should be displayed to him no matter he/she has rated them or not. So system should have capability to find out those points which should be shown as recommended points. System will utilize collaborative filtering techniques to first find out most similar users and then recommend points rated by that similar user to target user. Here system will utilize positive preferences of users to calculate recommended points.

6.4.3 Non Functional Requirements

System quality related statements that define the constraints on the services offered by the system are called nonfunctional requirements [81]. The statements that will help understanding the pattern of the development process of the proposed prototype is given below:

1. *The preferences oriented location based searching prototype website shall be easy to use and should not require special computer skills.*

Motivation Website is used by many people nowadays without any technical skills. The proposed prototype shall be easy to use as it will stand as other web applications. The electronic forms will have labels to guide users on what and where to fill data. Interfaces will also be user friendly.

2. *The preference oriented location based searching prototype website should be portable.*

Motivation Website should be easily accessible from many different devices. For example, it should be accessible through desktop computers,

mobiles, tablets etc. Website should be compatible with all browsers.

6.5 Developing the Prototype

The development process of the proposed prototype is divided into different modules where functionality of each module of the proposed prototype is described below:

Server Module

Figure 6-1 Architecture of the Prototype

6.6 Modules of the Prototype

System prototype is divided into client and server modules. The prototype website contains two client modules and for each client module there exist several server modules performing different functionalities. Client side modules will run based on the request of the user. But server side has two types of modules module

1 will activate when user will submit his request on the other hand module 2 will activate when system will receive feedback from user some module 2 will partially run based on user request and partially run on background.

6.6.1 Client Module 1

Request for a top ranked points under specified category

This web page allows users to request for a point by searching using category list of values. System will identify that category with location of user using GPS system techniques. We have developed this page using JSF technology with AJAX enabled components. When user will submit search button system will send that request to server side module 1 for finding top points under that category. This page contains one List of values for choosing a category for searching for example using this list of values user can search nearest and best restaurants or shopping malls. This page also contains a side table to show top searched points with different features like point name, point distance from user, available features etc. after seeing list user will evaluate available list points with respect to its preferences. Evaluations will be done in the forms of feedback score. Main page contains list for feedback score so which user can select one from available scores. Scores range is from 1 to 5. After pressing submit feedback button client module 1 will handover that list to server module 2 for further evaluations. Implementation of client module is done using different layers

- Writing business layer for fetching list of values data

```
class GoogleGeoCode
    Initialize Geo_Code_Server to Google Maps API
    declare and define main(arguments:args)
        Initialize code to "holiday inn downtown shanghai"
        Initialize response with Return getLocation(arguments:code)
        Initialize result array with Return parseLocation(code)
        Print latitude
        Print longitude

        Declare getLocation(arguments:code)
        Declare buildurl(arguments:code)
        Declare parseLocation(arguments:response)
        Declare getOrdinate(arguments:response)
    end function main
end class GoogleGeoCode
```

Figure 6-2 Implementation of GPS Coordinate Calculator

Implementing Ranking algorithm: After finding the coordinates of user we have sent a request to business tier to find out the unique id in the system which will be used to identify each user in the POLPS database and then passed whole request to ranking

- Writing business components for displaying data □
- Writing business components for submitting feedback □
- Designing of Java server faces page.

6.6.2 Server Module 1

On clicking submit button, the request is sent to the server where server side module is divided into different sub modules which are illustrated below:

- 1.Address conversion into GPS coordinates.
- 2.Filtering points based on searching time
- 3.Implementing the Ranking algorithm.
- 4.Displaying the ranking result on JSF page.

Address conversion into GPS coordinates: user request for a point by specifying the category of businesses and address to the client module. However, businesses in dataset contain GPS coordinates of the location. Therefore, to calculate the distance between point and user we need to convert the address into location coordinates i.e. latitude and longitude. We implement Google Geocode API (Application programming interface) to convert the address into latitude and longitude. We have implemented this technique using java programming language. Figure 6.2 is showing algorithm we implemented to get coordinated by passing normal address.

function of business tier. Ranking function received four parameters point category,

latitude, longitude of user and user id. Here is implementation of the ranking function which will calculate score of each point and return the list of top rated points.

Ranking function is combination of ranking algorithm and several supporting functions.

- Log query
- Distance calculator
- Feature extractor
- Public evaluation score calculator o Google score calculator o Public score calculator
- Distance score calculator
- Preference score calculator
- Total score calculator

```

Initialize RankedPoints List to rankedPoints
Try to
  if location_lat is not null and location_long is not null
    declare and initialize lat1 to Double value of location_lat
    declare and initialize lng1 to Double value of location_long
    declare and initialize queryid to return value placeQuery(arguments:userid,pointType,lat1,lng1)
    print queryid
    declare and initialize PointsViewImpl object points to return value getPoints
    set points.bind_PointType to Number object
    execute points query
    declare and initialize iter with points rowSetIterator
  While iter has next
    declare and initialize PointsViewRowImpl object row to iter next value
    declare and initialize RankedPoints object rp
    set rp point name to row name
    declare and initialize PointCategoryViewRowImpl object categoryRow to row point category view
    if categoryRow is not null
      set rp category to category Name of categoryRow
    end if
    else
      set rp category to "Unkown"
    end else
    parse rp queryid to integer
    declare and initialize DistanceCalculator object cal
    declare and initialize lat2 with value of row's latitude
    declare and initialize lng2 with value of row's longitude

    set rp distance to return result of distFrom(arguments:lat1,lng1,lat2,lng2)

    set rp pointFeatures to features of row PointFeautresView
    set rp pointId to row pointId after parsing to integer

    declare and initialize publicEvScore : call calculatePublicEvaluation(arguments:publicEvaluation ,googleRanking)
    declare and initialize distanceScore : call calculateDistanceEvaltion(arguments: userid,row pointId , rp distance)
    declare and initialize preferenceScore: call calcaltePreferenceScore(userid, row pointId )
    set rp score: call calculateTotalScore(publicEvScore, distanceScore,preferenceScore)
    set rp to userid
    add rp to rankedPoints
  end while
end if
catch exceptions

```

Figure 6-3 Implementation of POLPS Ranking Algorithm

Calculating Distance Between two Points: to calculate score of points based on distance between them first system need to know what is the distance between requester and the point so we have implemented a distance calculator function. This function has capability to receive GPS coordinated of both user and point one by one and calculate the distance between both in

```

class DistanceCalculator
  declare and define constructor
  declare
    declare and initialize earthRadius to 6371000
    declare and initialize dLat with lat2-lat1
    declare and initialize dLng with lng2-lng1
    declare and initialize a with sine(dLat/2) * sine(dLat/2)+cos(lat1)*cos(lat2)*sine(dLng/2)* sine(dLng/2)
    declare and initialize c with 2*tan(squareroot(a),squareroot(1-a))
    declare and initialize dist with earthRadius*c
    print distance dist/1000
    return dist/1000
  end funtion distanceCalculator

  declare and define main
    initialize DistanceCalculator object cal
    declare and initialize distance with return call DistanceCalculator(74.2583053,31.4334333,74.2105955,31.4029787)
    print distance dist/1000
    declare and initialize time t with current System time
    print t
  end funtion main
end class DistanceCalculator

```

Figure 6-4 Implementation of Distance Calculator Between two Points

terms of kilometers. Distance calculator calculates distance based on the radius of the earth its default result was in the form of meters but in user need it in the form of kilometers because we normally habitual of calculating distance in kilometers rather than meters. Here is code implemented using java language.

Implementing Google Ranking Algorithm:

Google ranking algorithm is a unique function used by Google search engine to calculate score of businesses based on its popularity on the internet it will use URL of the business to calculate the ranking score. Here main algorithm which is calculating score or rank of specified point is page rank. We have calculated rank of each point who has URL and available on the

internet. Score calculated using Google page rank has very high credibility as compared to the public evaluation score calculated from recommendation websites. So what we have done is added this score with public evaluation score and calculates a final score which is also referred to as public evaluation score. Here is java implementation of Google ranking algorithm.

```
class GooglePageRankUtil
    declare and define main (arguments: args)
        declare and initialize GooglePageRankUtil object pagerankTool
        print pagerankTool page rank
    end Function main
    declare and define getRank(arguments: pageAddress)
        declare and initialize result
        declare and initialize JenkinsHash jenkinsHash object
        declare and initialize hash with pageAddress and information hash
        declare and initialize url with url + hash+pageAddress
        print url
        Try to
            declare and initialize URLConnection conn with openConnection
            declare and initialize BufferedReader br with inputStream
            declare input
            while input = br.readLine is not null
                set result to substring of input last index
            catch exceptions
        end function getRank
end class GooglePageRankUtil
```

Figure 6-5 Implementation of Google Page Ranking Algorithm

Display the Ranking Results on User Interface:

After implementation of searching and ranking techniques POLPS ranking function prepare a list of all point which should be displayed to user system give them order according to their score. Whole work is done by business tier of POLPS.

Now business tier will give these results to controller tier to filter out top scores according to threshold and show prepare GUI to display. We have implemented this layer using ADF Controller and ADF UI layer components using JSF as UI technology.

. A snapshot of the output of the server module 1 is given in Figure 6.6.

SRNO	Point Name	Score	Point Features	Distance	Category	Rate	Relevance
1	Dr. Aysha Shehik	-1138.3100000000000	Radiologist, Pediatrician	8547	Unknown	5	0
2	Dr. Solomon Elahi	-1138.7100000000000	Radiologist, pathologist	8550	Unknown	5	0
3	Dr. Muhammad Fiaz	-1139.1533333333333	Radiologist	8551	Unknown	5	0
4	Dr. S Tahira Izhaz	-1139.4433333333333	ENT Specialist	8551	Unknown	5	0
5	DR Khurshid	-1139.9766666666666	ENT Specialist	8555	Unknown	5	0
6	Dr. Badar Munir	-1165.3100000000000	pathologist	8745	Unknown	5	0
7	Dr. Prof. Imran Iqbal	-1165.3766666666666	Radiologist, Pediatrician	8747	Unknown	5	0

Figure 6-6 Snapshot of the Output

6.6.3 Client Module 2

Feedback Learning:

After finding list of top points searched under some category user will evaluate those points. We have implemented this functionality on the same page which showing ranking results. A list of values has been given adjustment to each point showing number from 1 to 5 which user can

choose to give feedback. Feedback as we discussed previously is to learn preferences of user and to evaluate system accuracy. We have implemented this functionality using java, JSF, html and CSS languages.

SRNO	Point Name	Score	Point Features	Distance	Category	Rate	Relevance
1	Dr. Aysha Shehik	-1138.3100000000000	Radiologist, Pediatrician	8547	Unknown	5	0
2	Dr. Solomon Elahi	-1138.7100000000000	Radiologist, pathologist	8550	Unknown	4	0
3	Dr. Muhammad Fiaz	-1139.1533333333333	Radiologist	8551	Unknown	3	0
4	Dr. S Tahira Izhaz	-1139.4433333333333	ENT Specialist	8551	Unknown	2	0
5	DR Khurshid	-1139.9766666666666	ENT Specialist	8555	Unknown	1	0
6	Dr. Badar Munir	-1165.3100000000000	pathologist	8745	Unknown	4	0
7	Dr. Prof. Imran Iqbal	-1165.3766666666666	Radiologist, Pediatrician	8747	Unknown	4	0

Figure 6-7 Snapshot of the Feedback Submission Web Page

On submitting form system pops up many server modules.

6.6.4 Server Module 2

Server module 2 is divided into the following sub modules which are written below:

- 1) Feedback submission from client module 2.
- 2) Learning positive and negative feedback.
- 3) Decay feedback weight after passing time.
- 4) Finding similarity between preferences using collaborative filtering techniques.
- 5) Recommendation of businesses to user based on similarity.

Feedback Submission from Client Module 2: submitted feedback from user will be received by feedback submission module. UI layer bean will receive feedback in the form of array and that score will be attached to each point. Function called from UI layer will pass that feedback score to business tier for further processing. We have implemented this functionality using java, JSF based components. Figure 6.8 is showing implementation of algorithm.

```

declare and define function submitFeedback(arguments: userid,feedback)
    declare and initialize procedure with "begin pkg_feedback_learning_algo.submit_feedback_proc(?,?);end;"
    declare and initialize message
    Try to
        declare and initialize callableStatement stmt with DBTransaction callableStatement
        declare and initialize connection conn with stmt.connection
        set stmt array to 1 and getArray(arguments:conn,feedback)
        set stmt registerOutParameter
        execute stmt
        set message stmt.getString(arguments:2)
        close stmt
        call learnPreferences(arguments:userid)
    catch exceptions
end function submitFeedback
declare and define function getArray(arguments: conn,feedback)
    declare and initialize array
    Try to
        declare and initialize ArrayDescriptor desc with call createDescriptor(arguments: "POLS.FEEDBACK_TAB",conn)
        set array to call ARRAY(desc,conn,call getCollection(arguments: conn,feedback))
    catch exceptions
    return array
declare and define getCollection(arguments: conn,feedback)
    declare and initialize object inputList object call object(feedback.size)
    Try to
        declare and initialize StructDescriptor structDescriptor with call createDescriptor("POLS.FEEDBACK",conn)
        for all feedback.size
            declare and initialize RankedPoints p with feedback
            print objects ,userid,pointId,rating,rankNo,queryId,distance
            declare and initialize object objcr obj with new object(arguments: userid,pointId,rating,rankNo,queryId,distance)
            declare and initialize STRUCT object struct to new STRUCT(arguments structDescriptor,conn,obj)
            inputList=struct
        end loop for
    catch exceptions
end function getCollection

```

Figure 6-8 Implementation of Feedback Submission Function

Learning Positive and Negative Feedback: feedback submission module will hand over its feedback array to database procedure which we have written using PLSQL language. The procedure has implemented feedback learning algorithm which will learn positive and negative feedback. To calculate positive and negative feedback we have pre-defined each feedback score whether it is positive or negative. Feedback

learning algorithm will learn each feedback by look at the list we pre- defined. We have defined that feedback from 1 to 2 will be learned as negative feedback 3 will be learned as neutral feedback and 4 to 5 will be learned as positive feedback. After learning each point feedback algorithm will store each feedback in positive and negative feedback databases. Figure 6.9 is showing implemented algorithm.


```

PROCEDURE SUBMIT_FEEDBACK_PROC(P_FEEDBACK IN POLS.FEEDBACK_TAB,
                              P_OUT      OUT VARCHAR2) IS
    V_POINTID      NUMBER;
    V_USERID       NUMBER;
    V_RATING       NUMBER;
    V_RANKNO       NUMBER;
    V_STRENGTH_VALUE NUMBER;
    V_QUERYID      NUMBER;
    V_DISTANCE     NUMBER;
    V_SEQ          NUMBER;
BEGIN
    IF P_FEEDBACK.COUNT > 0 THEN
        FOR V_INDEX IN 1 .. P_FEEDBACK.COUNT LOOP
            V_USERID := P_FEEDBACK(V_INDEX).USERID;
            V_POINTID := P_FEEDBACK(V_INDEX).POINTID;
            V_RATING := P_FEEDBACK(V_INDEX).RATING;
            V_RANKNO := P_FEEDBACK(V_INDEX).rank_no;
            V_QUERYID := P_FEEDBACK(V_INDEX).QUERYID;
            V_DISTANCE := P_FEEDBACK(V_INDEX).DISTANCE;
            SELECT P.STRENGTH
            INTO V_STRENGTH_VALUE
            FROM POLS.PREFERENCES_MAPPING P
            WHERE P.VALUE = V_RATING;
            FOR I IN (SELECT F.FEATURE_ID
                     FROM POLS.POINT_FEATURES F
                     WHERE F.POINT_ID = V_POINTID) LOOP
                SELECT POLS.RANKING_SEQ.NEXTVAL INTO V_SEQ FROM DUAL;
                INSERT INTO POLS.RANKINGRESULTS
                (RESULT_ID,POINT_ID,RANK_NO,PREFERENCE_VALUE,FEEDBACK_DATE,RESULT_DATE,QUERY_ID,DISTANCE,FEATURE_ID,STRENGTH_VALUE)
                VALUES
                (V_SEQ,V_POINTID,V_RANKNO,V_RATING,SYSDATE,SYSDATE,V_QUERYID,V_DISTANCE,I.FEATURE_ID,V_STRENGTH_VALUE);
            END LOOP;
        END LOOP;
        COMMIT;
        P_OUT:='SUCCESS';
    END IF;
    EXCEPTION
    WHEN OTHERS THEN
        P_OUT:='Fail';
END SUBMIT_FEEDBACK_PROC;

```

Figure 6-9 Feedback Learning Algorithm

Collaborative Feedback Learning: This module will run in background of each module. It will find out the similarity between different users and recommend points to the users which they have not rates but due to similarity between users system find it suitable to be shown to the requester. Here we have implemented two algorithms one recommend based on user similarity and other recommend based on item similarity. Implementation strategy of both algorithms is simple we have written two procedures in database. One will calculate similarity based on user to user comparison other based on items to item comparisons. Figure 6.10 shows implementation of user to user and Figure 6.11 shows implementation of item to item collaborative filtering algorithm.

Recommendation of Businesses to User based on Similarity. Recommendation is part of collaborative filtering algorithm but recommendation come after running collaborative filtering algorithms like user to user recommendation and item to item recommendation.

Purpose of this function was to recommend those businesses or points to user U_1 who have been rated by some other user U_2 who is most similar to user U_1 . Recommendation function has utilized results of CF Item to item and user to user algorithms. We have implemented this functionality using PLSQL language same utilized in CF algorithms.

Figure 6.12 shows implementation details of recommendation function.

```

PROCEDURE PROC_FIND_SIMILAR_PREFERENCES(P_USERID IN VARCHAR2,
                                         P_OUT    OUT VARCHAR2) IS

    P_NU      NUMBER; -- NUMBER OF POSITIVE FEATURES OF USER U
    P_NV      NUMBER; -- NUMBER OF POSITIVE FEATURES OF V
    P_LOG      NUMBER; --NATURAL LOG VALUE
    P_SIGMA    NUMBER; --SIGMA CALCULATION
    P_U_F_COUNT NUMBER; --POSITIVE PREFERENCE COUNT FOR USER U
    P_V_F_COUNT NUMBER; --POSITIVE PREFERENCE COUNT FOR USER V
    P_MIN      NUMBER; -- MINIMUM COUNT BETWEEN USER U AND USER V
    P_SUM      NUMBER; --SUMMATION OF COMMON FEATURES OF USER U AND USER V

    P_SIMILARITY NUMBER; --SIMILARITY OF USER U AND USER V;
    P_MOST_SIM_USER VARCHAR2(30);

BEGIN
    --FIRST FIND ALL POSITIVE PREFERENCES OF USER P_USERID
    FOR I IN (SELECT DISTINCT Q.USERID
              FROM POLS.RANKINGRESULTS R, POLS.QUERY Q WHERE R.QUERY_ID = Q.QUERYID AND R.PREFERENCE_VALUE > 3
              AND Q.USERID = P_USERID) LOOP
        --CALCULATE TOTAL NUMBER OF ITEMS OF USER U
        SELECT COUNT(FEATURE_ID)
        INTO P_NU
        FROM (SELECT USERID, FEATURE_ID, COUNT(FEATURE_ID)
              FROM (SELECT R.POINT_ID,R.FEATURE_ID,R.PREFERENCE_VALUE,R.STRENGTH_VALUE,Q.USERID
                        FROM POLS.RANKINGRESULTS R, POLS.QUERY Q
                        WHERE R.QUERY_ID = Q.QUERYID AND R.PREFERENCE_VALUE > 3 AND Q.USERID = I.USERID) U
              GROUP BY U.FEATURE_ID, U.USERID ORDER BY U.USERID);

        FOR J IN (SELECT DISTINCT Q.USERID
                  FROM POLS.RANKINGRESULTS R, POLS.QUERY Q
                  WHERE R.QUERY_ID = Q.QUERYID) LOOP
            P_SUM := 0;
            IF J.USERID <> I.USERID THEN
                --CALCULATE TOTAL NUMBER OF POSITIVE ITEMS OF USER V
                FOR K IN (SELECT DISTINCT R.FEATURE_ID
                          FROM POLS.RANKINGRESULTS R, POLS.QUERY Q WHERE R.QUERY_ID = Q.QUERYID AND R.PREFERENCE_VALUE > 3
                          AND Q.USERID = I.USERID
                          INTERSECT
                          SELECT DISTINCT R.FEATURE_ID
                          FROM POLS.RANKINGRESULTS R, POLS.QUERY Q WHERE R.QUERY_ID = Q.QUERYID AND R.PREFERENCE_VALUE > 3
                          AND Q.USERID = J.USERID) LOOP
                    --NOW CALCULATE PREFERENCE COUNT OF EACH PREFERENCE ITEM FOR EACH USER AND TAKE MIN
                    SELECT UV.TOTAL_COUNT
                    INTO P_U_F_COUNT
                    FROM POLS.USER_POSITIVE_FEATURES_VIEW UV WHERE UV.USERID = I.USERID AND UV.FEATURE_ID = K.FEATURE_ID;

                    SELECT UV.TOTAL_COUNT
                    INTO P_V_F_COUNT
                    FROM POLS.USER_POSITIVE_FEATURES_VIEW UV WHERE UV.USERID = J.USERID AND UV.FEATURE_ID = K.FEATURE_ID;
                    IF P_U_F_COUNT < P_V_F_COUNT THEN
                        P_MIN := P_U_F_COUNT;
                    END IF;
                    IF P_V_F_COUNT < P_U_F_COUNT THEN
                        P_MIN := P_V_F_COUNT;
                    END IF;
                    IF P_V_F_COUNT = P_U_F_COUNT THEN
                        P_MIN := P_V_F_COUNT;
                    END IF;
                    P_SUM := P_SUM + P_MIN;
                END LOOP;
                P_SIMILARITY := P_SUM / P_SIGMA;
                --SAVE IN TEMPORARY TABLE
                INSERT INTO POLS.T_SIMILARITY__WEIGHT
                VALUES(J.USERID, P_SIMILARITY);
            END IF;
        END LOOP;
        --FIND MOST SIMILAR USER
        SELECT T.USERID
        INTO P_MOST_SIM_USER
        FROM POLS.T_SIMILARITY__WEIGHT T WHERE T.SIMILARITY_VALUE =
        (SELECT MAX(SIMILARITY_VALUE) FROM POLS.T_SIMILARITY__WEIGHT);
        PKG_CF_ALGORITHM.PROC_RECOMMEND_FEATURES(P_USERID,
                                                P_MOST_SIM_USER,
                                                P_OUT);
    END LOOP;
EXCEPTION
    WHEN OTHERS THEN
        P_OUT := 'SORRY PROBLEM IN CALCULATING SIMILARITY OF USERS ' ||
        SQLERRM;
END PROC_FIND_SIMILAR_PREFERENCES;

```

Figure 6-10 Implementation of User to User Collaborative Filtering Algorithm

Implementation of item to item collaborative filtering was done by applying algorithm described in figure 4.1 of chapter 4.

```

PROCEDURE PROC_ITEM_ITEM_RECOMENDATION(P_POINT_TYPE IN VARCHAR2,
P_OUT OUT VARCHAR2) IS
IS_ALREADY NUMBER := 0;
MAIN_POINT_PREFERENCE number := 0;
COSINE_SIMILARITY NUMBER := 0;
DOT_PRODUCT NUMBER := 0;
MODE_OF_POINTS NUMBER := 0;
MAX_SIMILAR_POINT NUMBER;
max_spp number;
max_sps number;
BEGIN
--fetch all those points under category
FOR I IN (SELECT P.POINTID
FROM POLS.POINTS P
WHERE P.POINT_TYPE = P_POINT_TYPE) LOOP
--fetch all those users who rated current point
FOR J IN (SELECT PP.USERID,
PP.POINT_ID,
PP.PREFERENCE_VALUE,
PP.STRENGTH_VALUE
FROM POLS.PERSONALPREFERENCES PP
WHERE PP.USERID IN
(SELECT TP.USERID
FROM POLS.PERSONALPREFERENCES TP
WHERE TP.POINT_ID = I.POINTID)
AND PP.PREFERENCE_VALUE > 2) LOOP

IF J.POINT_ID <> I.POINTID THEN
IS_ALREADY := 0;
SELECT COUNT(*)
INTO IS_ALREADY
FROM POLS.T_USER_WISE_POINTS_UNION
WHERE USER_POINT = J.POINT_ID;
IF IS_ALREADY = 0 THEN
INSERT INTO POLS.T_USER_WISE_POINTS_UNION
(USER_POINT, USER_POINT_PREFERENCE, USER_POINT_STRENGTH)
VALUES
(J.POINT_ID, J.PREFERENCE_VALUE, J.STRENGTH_VALUE);
END IF;
IS_ALREADY := 0;
END IF;
IF J.POINT_ID = I.POINTID THEN
MAIN_POINT_PREFERENCE := J.PREFERENCE_VALUE;
END IF;
END LOOP;

--NOW CALCULATE SIMILARITY OF EACH POINT P WITH ANY OTHER POINTS RECOMENDED BY OTHER USERS
--USING COSINE SIMILARITY
FOR J IN (SELECT TU.USER_POINT,
TU.USER_POINT_PREFERENCE,
TU.USER_POINT_STRENGTH
FROM POLS.T_USER_WISE_POINTS_UNION TU) LOOP

DOT_PRODUCT := MAIN_POINT_PREFERENCE * J.USER_POINT_PREFERENCE;
MODE_OF_POINTS := SQRT(J.USER_POINT_PREFERENCE *
J.USER_POINT_PREFERENCE) *
SQRT(MAIN_POINT_PREFERENCE *
MAIN_POINT_PREFERENCE);
IF MODE_OF_POINTS <> 0 THEN
COSINE_SIMILARITY := DOT_PRODUCT / MODE_OF_POINTS;
UPDATE POLS.T_USER_WISE_POINTS_UNION
SET SIMILARITY_VALUE = COSINE_SIMILARITY WHERE USER_POINT = J.USER_POINT;
END IF;
IF MODE_OF_POINTS = 0 THEN
UPDATE POLS.T_USER_WISE_POINTS_UNION
SET SIMILARITY_VALUE = 0
WHERE USER_POINT = J.USER_POINT;
END IF;
END LOOP;
SELECT USER_POINT
INTO MAX_SIMILAR_POINT
FROM POLS.T_USER_WISE_POINTS_UNION TU
WHERE TU.SIMILARITY_VALUE =
(SELECT MAX(SIMILARITY_VALUE)
FROM POLS.T_USER_WISE_POINTS_UNION);
FOR J IN (SELECT TP.USERID FROM POLS.PERSONALPREFERENCES TP
WHERE TP.POINT_ID = I.POINTID AND TP.PREFERENCE_VALUE > 2
MINUS
SELECT TTP.USERID FROM POLS.PERSONALPREFERENCES TTP
WHERE TTP.POINT_ID = MAX_SIMILAR_POINT AND TTP.PREFERENCE_VALUE > 2) LOOP
SELECT UU.USER_POINT_PREFERENCE, UU.USER_POINT_STRENGTH
INTO max_spp, max_sps
FROM POLS.T_USER_WISE_POINTS_UNION UU WHERE UU.USER_POINT = MAX_SIMILAR_POINT;

INSERT INTO POLS.PERSONALPREFERENCES
(userid, point_id, preference_value, strength_value)
VALUES
(j.userid, MAX_SIMILAR_POINT, max_spp, max_sps);
END LOOP;
COMMIT;
END LOOP;
P_OUT := 'DATA SAVED SUCCESSFULLY';
EXCEPTION
WHEN OTHERS THEN
P_OUT := 'Problem while item to item collaborative filtering ' ||
sqlerrm;
rollback;
END;

```

Figure 6-11 Implementation of Item to Item Collaborative Filtering Algorithm

```

PROCEDURE PROC_RECOMMEND_FEATURES(P_USERID IN VARCHAR2,
                                  P_SIMI_USER IN VARCHAR2,
                                  P_OUT OUT VARCHAR2) IS
    P_COUNT NUMBER;
BEGIN
    --FIND ALL THOSE POSITIVE FEATURES OF SIMILAR USER WHICH ARE NOT BEING USED IN P_USERID
    FOR I IN (SELECT DISTINCT R.FEATURE_ID
              FROM POLS.RANKINGRESULTS R, POLS.QUERY Q
              WHERE R.QUERY_ID = Q.QUERYID
              AND R.PREFERENCE_VALUE > 3
              AND Q.USERID = P_SIMI_USER
              MINUS
              SELECT DISTINCT R.FEATURE_ID
              FROM POLS.RANKINGRESULTS R, POLS.QUERY Q
              WHERE R.QUERY_ID = Q.QUERYID
              AND R.PREFERENCE_VALUE > 3
              AND Q.USERID = P_USERID) LOOP
        -- GET PREFERENCE VALUE AND CHECK IF THAT FEATURE EXIST IN USER PERSONAL PREFERENCE DATABASE
        P_COUNT := 0;
        FOR J IN (SELECT R.POINT_ID,
                        R.FEATURE_ID,
                        R.PREFERENCE_VALUE,
                        R.STRENGTH_VALUE,
                        R.FEEDBACK_DATE
                  FROM POLS.RANKINGRESULTS R, POLS.QUERY Q
                  WHERE Q.QUERYID = R.QUERY_ID
                  AND Q.USERID = P_SIMI_USER
                  AND R.FEATURE_ID = I.FEATURE_ID) LOOP
            SELECT COUNT(*)
            INTO P_COUNT
            FROM POLS.PERSONALPREFERENCES P
            WHERE P.USERID = P_USERID
            AND P.POINT_ID = J.POINT_ID
            AND P.FEATURE_ID = J.FEATURE_ID;
            IF P_COUNT = 0 THEN
                FOR J IN (SELECT R.POINT_ID,
                                R.FEATURE_ID,
                                R.PREFERENCE_VALUE,
                                R.STRENGTH_VALUE,
                                R.FEEDBACK_DATE
                          FROM POLS.RANKINGRESULTS R, POLS.QUERY Q
                          WHERE Q.QUERYID = R.QUERY_ID
                          AND Q.USERID = P_SIMI_USER
                          AND R.FEATURE_ID = I.FEATURE_ID) LOOP
                    SELECT COUNT(*)
                    INTO P_COUNT
                    FROM POLS.PERSONALPREFERENCES P
                    WHERE P.USERID = P_USERID
                    AND P.POINT_ID = J.POINT_ID
                    AND P.FEATURE_ID = J.FEATURE_ID;
                    IF P_COUNT = 0 THEN
                        INSERT INTO POLS.PERSONALPREFERENCES PP
                        (USERID,
                         POINT_ID,
                         FEATURE_ID,
                         PREFERENCE_VALUE,
                         STRENGTH_VALUE,
                         RATING_DATE)
                        VALUES
                        (P_USERID,
                         J.POINT_ID,
                         J.FEATURE_ID,
                         J.PREFERENCE_VALUE,
                         J.STRENGTH_VALUE,
                         SYSDATE);
                    END IF;
                END LOOP;
            END IF;
        END LOOP;
        COMMIT;
        P_OUT := 'DATA HAS BEEN SAVED SUCCESSFULLY';
    EXCEPTION
        WHEN OTHERS THEN
            P_OUT := 'PROBLEM RECOMMENDING FEATURE' || SQLERRM;
    END PROC_RECOMMEND_FEATURES;

```

Figure 6-12 Implementation of Recommendation Algorithm

In sum, we conclude that by implementing multiple API's a prototype preference oriented

location based point searching system. We have utilized different languages and platforms for

implementation of different algorithms. All features and algorithms have been integrated with each other to form a complete lifecycle of proposed framework. In next chapter we will evaluate our proposed system using different

evaluation techniques. This chapter have discussed about implementation of each algorithm utilized in POLPS framework. Next chapter will discuss about experimental results and their analysis.

Chapter 7: Results and Analysis

This chapter aims to do complexity analysis of our proposed framework with evaluation of experimental results using different evaluation techniques.

7.1 Complexity Analysis

Here will discuss about best, worst and averages analysis of proposed framework. POLPS system complexity depends upon three factors number of users U_N using the system, number of points/businesses P_N in the system databases and number of similar users CF_N this factor is important because CF based Item to Item recommendation algorithm will use similar users to recommend points/businesses. To analyze time complexity of system we will use big o notations and will be analyzed as below.

Best case:

$$O(U_N * P_N)$$

System latency is increased with increase in both number of users and number points. More number of users system performed more calculations and overhead of time is increased like more calculations in feedback learning. In the same way system has to perform more comparisons due to increase in number points. Comparisons are done with properties/attributes of the points and preferences of the user like user has a preference that in evening he prefers to visit malls which have shops selling garments on sale with availability of elevators and parking and mall attributes are cheap cloths ,sales garments, elevators available, parking not available. So system will remove that mall by comparing user preferences with mall attributes.

Worst case:

$$O(U_N * P_N + CF_N)$$

System latency is increased with increase in U_N and P_N . More number of users and Points more comparisons system has to perform with additional cost of finding similar users and recommendation of those points to the users which he/she have not rated in past. So main issues that affect time complexity of the system are U_N , P_N and CF_N Average case:

$$O(U_N * P_N + CF_N)$$

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

7.2 Experimental Evaluation

We have evaluated POLPS system using different measurement. We have used the area under receiver operating characteristic normally called ROC curve or ROC Area and for accuracy measurement of ranking algorithm, for evaluating the effectiveness of POLPS framework we have utilized normalized discounted cumulative gain NDCG.

NDCG have been used to find out the gain. Whenever user sent query to POLPS system, system gave him the list of points sorted by ranking order from top to bottom. User evaluated each point in the ranking list in the form of feedback. We found out gain which is calculated cumulatively from the top of the result list to the bottom of the result list. If point has been given higher rank and user gave it low score then relevance score will be low. NDCG score shows the ranking algorithm 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: user wanted to find out open shopping malls from

10PM to 12AM but system incorrectly shown those shopping malls which were closed.

We measured this behavior by 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.

7.2.1 Impact of Different Weights for POLPS

During calculation of final ranking list of points we have assigned different weights to different factors. In POLS research, they have utilized three main factors public evaluation, distance between points and user to user preferences. So they have assigned different weights to each of factors that are weight α , β and γ respectively.

Symbol α is assigned to weight of public evaluation β assigned to distance and γ assigned to user preferences respectively. To find out the effect of each weight on the ranking accuracy of the system, we vary the proportions of all weights performed simulation by simulate users rating on top 7 points in each result list and measured result of this experiment by drawing NDCG graph in below figure.

7.2.2 Impact of Different Weights without using Google Ranking Approach

To find out best possible combination of weights of three main ranking issues utilized

while calculating the score of points, that is weight of public evaluation denoted by α , weight of point distance denoted by β and weight of user preferences about point denoted by γ weights have been described in equation no 1 of section 4. We have done experiment by taking users feedback on top 7 points collected their feedback and measured using NDCG. By seeing figure 7.1 we noticed that by just considering public evaluation system have shown worst accuracy of ranking lists. If system considered only distance or personal preferences POLPS ranking accuracy was less than 0.76. When considered two ranking issues instead of one system gave better output as compared to previous one. System NDCG comparison has a higher accuracy of 0.91 when considered all three parameters but with double emphasis on personal preferences. So finally we found that system has higher accuracy of 0.97 when considered 1:2:2 as weighted combination of three ranking issues. These results showed that a point which have higher public evaluation does not means that this point will be preferred by user but on opposite of public evaluation point distance and user personal preferences are more important ranking issues. So finally we summarize our weighted combination of 1:2:2 and normalized all weights to 1 by following experiments α is set as 0.2, β is set as 0.4 and γ is set as 0.4 so that $\alpha: \beta: \gamma = 1:2:2$ and $\alpha + \beta + \gamma = 1$

Figure 7-1 NDCG Score without using Google Ranking

7.2.3 Impact of Different weights using Google Ranking Approach

In POLPS to enhance the credibility of public evaluation we have changed ranking algorithms by adding Google ranking approach. So in this approach we divided public evaluation ranking issue further into two categories Score of

Google and score of public evaluations. So we gave two weights to both ranking issues α : $\beta=1$ where $\alpha=0.6$ weight of public evaluations and $\beta=0.4$ weight of Google evaluation. Again we simulate our experiment results have been shown in Figure 7.2. We found that system showed improvement in accuracy of ranking results.

Figure 7-2 NDCG Score with Google Ranking

7.2.4 Impact of Google Weight and Public Evaluation Weight on Ranking Efficiency

While calculating ranking score for each searched point system divided public evaluation into two sub evaluation categories one is same public evaluation and other one is Google based evaluation. Each evaluation technique has been assigned a weight. Final score is calculated by multiplying each evaluation score with respective weight value. What should be the best value of weight for each strategy is decided by continuous feedback effect analysis by using NDCG graphs.

Analysis started by first assigning public evaluation $\alpha=0.1$ and Google score $\beta=0.9$ by ensuring $\alpha+\beta=1$. We increased evaluation score by 0.1 and decreased Google score by 0.1 to analyze effect of public evaluation on ranking results and user feedback. We analyzed that depending more on Google power was not satisfying user and same when public score was given more than 60% weightage shown in figure 7.3. System was showing stable behavior with increased ranking efficiency with $\alpha=0.6$ and $\beta=0.4$.

Figure 7-3 NDCG Comparison between α and β

7.2.5 Impact of Various Noise Rating Rates on Ranking Accuracy

To find out the behavior of system on different density of noise this experiment was performed on POLPS simulator. We simulate a noise rating rate for the simulation user to see the noise rating of the POLPS. To measure the accuracy of system we draw two graphs NDCG and ROC area. We varied noise rating from 0.1 to 0.5. We find out that NDCG and ROC area showed a continuous similar behavior on noise. Accuracy of system increases with decrease in noise rating and decreases with increase in noise rating. The lower rating noise rate the higher ranking accuracy shown in figure 7.4, 7.5 and 7.6 respectively.

Figure 7-4 NDCG Comparison between Noise and Accuracy of System

Figure 7-5 Comparison between Noise and Accuracy of System using ROC Analysis of TPR Rate

Figure 7-6 Comparison between Noise and Accuracy of System using ROC Analysis of FPR Rate

7.2.6 Collaborative Filtering Algorithm Performance Comparison

To analyze performance of both techniques in terms of time taken by both techniques to find out similar users we performed simulation experiments. To make it more real we performed simulation on 0.389 million users with

1.5 million reviews of 61 thousand points. We run both algorithms on same number of users and points for 5 times and has drawn a comparison graph both techniques shown in figure 7.7. We can see that Item to Item technique takes comparably a smaller number of minutes to find out similar users and to recommend points to similar user.

Figure 7-7 Comparison between Performance of User to User CF Technique and Item to Item CF Technique

Chapter 8: Conclusion

The main research question which were addressed and resolved are:

- How we will design our model? What will be the indicators of dependent and independent relationships in our model?
- What type of techniques and tests we are going to use in the modeling of replica design?
- How the model is verified?
- How the system might be implemented in a working application?
- How the feedback of the system might be generated?
- How the convenience sampling might be attained to check the user's acceptance for the system?
- What statistical scores will be allotted to each entity at the end of the system in the form of ranks?
- How we will be handling reliability, testability of the model?

We started by answering every research question which been resolved in our work. While designing the model we focused to the comparison with earlier models. While answering above question our research has proposed a new framework for searching points and named it as "preference oriented Location based point search (POLPS)". System possesses the capability to integrate user location, query time, and user preference.

According to our knowledge, this is a unique work which is simultaneously taking: location-based searching techniques with ranking techniques, preferences learning techniques and collaborative filtering technique with utilization of item to item collaborative filtering techniques, into account.

To analyze the performance of POLPS, we have done different types of experiments by directly receiving feedback from users; running simulations also compared performance of POLPS with existing POLS system. Through experimental results we have shown that our proposed system outperforms as compared to other existing approaches by testing under different environmental conditions. Under different location and temporal information, system accurately learned preferences of user by utilizing feedback learning algorithms and CF-based preference learning algorithm with enhanced item to item collaborative filtering and recommendation capabilities. At the end, system ranked point list by using proposed hybrid Ranking Algorithm with utilization of Google ranking approach. We have designed system ranking strategy to find out popular and nearby points for user when there is lack of initial data. We have resolved new item problem by using searched point properties as the features. Cold start or lacks of initial data are problems that often occurred in the traditional systems. By doing experiments, we have shown that our proposed system showed best performance and achieved high quality with low computation cost.

In future research we will focus on enhancement of system capability to incorporate negative feedback for recommendation and introduce qualitative feedback learning with quantitative feedback learning because using quantitative feedback to calculate feedback of user is not a very secure way because there may be chance that user unable to give right score to some point. Qualitative feedback helps in different ways one qualitative helps researcher to analyze feedback at an individual level. It can be an added dimension for quantitative answers, second using quantitative feedback with qualitative feedback give u a complete picture of the user feedback. It also helps to reduce inconsistency from feedback.

References

1. J. Tan, E. Lu and V. Tseng, "Preference-oriented mining techniques for locationbased store search", *Knowledge and Information Systems*, vol. 34, no. 1, pp. 147-169, 2012.
2. Zeng Y, Chen Y and Xie X, "Geolife 2.0: a location based social networking service", In 10th international conference on mobile data management, Taipei, Taiwan, 2009, pp 357-359.
3. iPeen(2014, August, 16). Ipeen[online]. Available: <http://www.ipeen.com.tw/>
4. AOL(2000). MapQuest[online]. Available: <http://www.mapquest.com/>
5. Google(2004, October). GoogleMaps[online]. Available: <https://www.maps.google.com>
6. PAPAGO!Inc(2001). PAPAGO[online]. Available: <http://www.papago.com.tw/>
7. Howard Schultz(1987). Starbucks[online]. Available: <http://www.starbucks.com/>
8. Tuma I Ding Y and Calermsook K, "Utilizing web 2.0 in web service ranking", In 3rd international conference on digital society, Cancun, Mexico, 2009, pp 174179.
9. Belkin NJ, "Helping people find what they don't know", *Communications of the Association for Computing Machinery* 43(8), New York, NY, USA, 2000, pp 58-61.
10. Horozov T, Narasimhan N and Vasudevan V, "Using location for personalized PoI recommendations in mobile environments", In 6th international symposium on applications on internet, Phoenix, AZ, 2006, pp 124-129.
11. Jin R, Si L, Zhai C et al, "Collaborative filtering with decoupled models for preferences and ratings", In 12th international conference on Information and knowledge management, New York, NY, USA, 2003, pp 106-309.
12. Yong Rui, T. Huang, M. Ortega et al, "Relevance feedback: a power tool for interactive content-based image retrieval", *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 5, pp. 644-655, 1998.
13. Lio NN, Yang Q, "Eigen Rank: a ranking oriented approach to collaborative filtering", In 31th International ACM SIGIR Conference on research and development in Information retrieval, New York, NY, USA, 2008, pp 83-90.
14. SARWAR, B. M., KARYPIS et al, "Item-based collaborative filtering recommendation algorithms", In 10th International World Wide Web Conference, ACM Press, New York, NY, USA, 2001, pp. 285-295.
15. D. Muhlestein and S. Lim, "Online learning with social computing based interest sharing", *Knowledge and Information Systems*, vol. 26, no. 1, pp. 31-58, 2009.
16. Cao, Z. Wu, Y. Wang and Y. Zhuang, "Hybrid Collaborative Filtering algorithm for bidirectional Web service recommendation", *Knowledge and Information Systems*, vol. 36, no. 3, pp. 607-627, 2012.
17. R. Dewri and R. Thurimella, "Exploiting Service Similarity for Privacy in Location-Based Search Queries", *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 2, pp. 374-383, 2014.
18. X. Zhang and H. Bae, "Location Positioning and Privacy Preservation Methods in Location-based Service", *IJSIA*, vol. 9, no. 4, pp. 41-52, 2015.
19. W. Zhang, S. Guo and S. Zhang, "Personalized manufacturing service recommendation using semantics-

- based collaborative filtering", *Concurrent Engineering*, vol. 23, no. 2, pp. 166-179, 2015.
20. G. Gültekin and O. Bayat, "Smart Location-Based Mobile Shopping Android Application", *JCC*, vol. 02, no. 08, pp. 54-63, 2014.
 21. S. Daneshmand, A. Javari, S. Abtahi and M. Jalili, "A Time-Aware Recommender System Based on Dependency Network of Items", *The Computer Journal*, vol. 58, no. 9, pp. 1955-1966, 2014.
 22. 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.
 23. H. Zhang, F. Min, X. He and Y. Xu, "A Hybrid Recommender System Based on User-Recommender Interaction", *Mathematical Problems in Engineering*, vol. 2015, pp. 1-11, 2015.
 24. X. Chen, Z. Zheng, Q. Yu and M. Lyu, "Web Service Recommendation via Exploiting Location and QoS Information", *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 7, pp. 1913-1924, 2014.
 25. Y. Zhu, J. Huang, Z. Zhang, Q. Zhang, T. Zhou and Y. Ahn, "Geography and Similarity of Regional Cuisines in China", *PLoS ONE*, vol. 8, no. 11, p. e79161, 2013.
 26. B. S. J. Burki(2013). Rising Lahore and recovering Pakistan[online]. Available: <http://tribune.com.pk/story/583094/rising-lahore-and-recovering-pakistan/>.
 27. GPS Logger forAndroid(2013).GPS Logger[online]. Available: <https://play.google.com/store/apps/details?id=com.mendhak.gpslogger>
 28. CSV - What, Why and How(2012).[online]. Available:<http://www.shoppingcart-migration.com/blog/61-must-know-tips/5985-csv-what-why-and-how-> .
 29. Record(2012)[online]. Available:<http://www.techterms.com/definition/record>.
 30. YELP(2004). YELP[online]. Available: <http://www.yelp.com>.
 31. S. YUAN, W. CHEN and S. FU, "User behavior similarity analysis of location based social network", *Journal of Computer Applications*, vol. 32, no. 2, pp. 322325, 2013.
 32. S. Elfirdoussi, Z. Jarir and M. Quafafou, "Ranking Web Services using Web Service Popularity Score", *International Journal of Information Technology and Web Engineering*, vol. 9, no. 2, pp. 78-89, 2014.
 33. P. Bedi and R. Sharma, "Trust based recommender system using ant colony for trust computation", *Expert Systems with Applications*, vol. 39, no. 1, pp. 11831190, 2012.
 34. Mohan, "RESOURCE SELECTION IN GRID ENVIRONMENT BASED ON TRUST EVALUATION USING FEEDBACK AND PERFORMANCE", *American Journal of Applied Sciences*, vol. 10, no. 8, pp. 924-930, 2013.
 35. M. Hamid, M. Naser, M. Hasan and H. Mahmud, "A cohesion-based friendrecommendation system", *Soc. Netw. Anal. Min.*, vol. 4, no. 1, 2014.
 36. B. Yurekli Yilmazel and C. Kaleli, "Robustness analysis of arbitrarily distributed data-based recommendation methods", *Expert Systems with Applications*, vol. 44, pp. 217-229, 2016.
 37. B. Leite Dantas Bezerra and F. Tenorio de Carvalho, "Symbolic data analysis tools for recommendation systems", *Knowledge and Information Systems*, vol. 26, no. 3, pp. 385-418, 2010.
 38. A. Anamika and K. Tyagi, "Secure Approach for Location Aided Routing in Mobile Ad Hoc Network", *International Journal of Computer Applications*, vol. 101, no. 8, pp. 24-27, 2014.
 39. N. Arora and A. Jangra, "GLAAR: Geographic Location Aware Adaptive

Routing in Mobile Ad Hoc Networks (MANETs)", International Journal of Computer Applications, vol. 50, no. 22, pp. 17-22, 2012.

40. W. Yuan, D. Guan, L. Shu and J. Niu, "Recommender Searching Mechanism

for Trust-Aware Recommender Systems in Internet of Things", Automatika Journal, vol. 54, no. 4, 2013.