Location Based Context Aware Recommender System Through User-defined Rules

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering

in

Software Engineering

Submitted By

Silky Sharma

(801331027)

Under the supervision of:

Dr. Damandeep Kaur

Assistant Professor, CSED

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

THAPAR UNIVERSITY

PATIALA – 147004

July 2015

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, "Location Based Context Aware Recommender System through User Defined Rules", in partial fulfillment of the requirements for the award of degree of Master of Engineering in *Software Engineering* submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of *Dr. Damandeep Kaur* and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

Signature: Silky Sharma

(Silky Sharma)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

Daman

(Dr. Damandeep Kaur)
Assistant Professor, CSED

Countersigned by

(Dr. Deepak Garg)
Head
Computer Science and Engineering Department
Thapar University
Patiala

(Dr. S. S. Bhatia)
Dean (Academic Affairs)
Thapar University
Patiala

ACKNOWLEDGEMENT

No volume of words is enough to express my gratitude towards my guide **Dr. Damandeep Kaur**, Department of Computer Science and Engineering, Thapar University, Patiala, who has been concerned and has aided for all the materials essential for the preparation of this thesis report. She has helped to explore this vast topic in an organized manner and provided me with all the ideas on how to work towards a research oriented venture.

I am also thankful to **Dr. Deepak Garg**, Head of Department, Computer Science Engineering Department, for the motivation and inspiration that triggered me for the thesis work.

I would also like to thank the staff members and my colleagues who were always there at the need of the hour and provided will all the help and facilities, which I required, for the completion of my thesis work.

Most importantly, I would like to thank my parents and the almighty for showing me the right direction out of the blue, to help me stay calm in the oddest of the times and keep moving even at times when there was no hope.

Silky Sharma

ABSTRACT

Recommender systems are a subclass of information filtering system and are widely used in the e-commerce domain. They filter huge amount of data to provide personalized recommendations on services or products to users. Most of the existing approaches to develop a recommender system do not take into account contextual information such as weather, day, time, distance and location to provide recommendations. This thesis proposes a location based context aware recommender system through user defined rules that uses rules to provide context awareness in the system and a ranking function to generate top-k recommendations. The contextual data is defined by the users and is stored in the form of rules and RuleML is chosen as a rule based language. When an active user needs recommendations about nearby places then contextual data in the user-defined RuleML rules is extracted, evaluated, and top-k recommendations of nearby places based on the ranking function are presented to the user on the Google map.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Expansion	Page no.
GPS	Global Positioning System	4
CARS	Context Aware Recommender Systems	2
Geo SPLIS	Geographical Semantic Personalized Location Information	11
SWRL	Semantic Web Rule Language	11
POIs	Point Of Interests	11
OWL	Web Ontology Language	11
RFID	Radio Frequency Identification	12
RuleML	Rule Markup Language	25

CHAPTER 1
INTRODUCTION

A recommender system is a tool or software that aims to provide personalized recommendations about services or online product to users. The purpose of developing a recommendation system is to reduce information overload by extracting the most relevant information about products or services from a huge data and providing personalized recommendations to the user.

Since the mid-1990s various recommender system techniques have been proposed and using these techniques many recommender systems have been developed for various different applications. There are number of domains where we can use a recommender system. Some of the domains where we can use a recommender system: e-Library, e-Tourism, e-Gov., e-Commerce/ Shopping, e-Business, e-Learning, e-Resource services and e-Group activities. In the recent years, the interest of researchers has drastically increased in recommender system. In the most highly rated internet sites such as YouTube, Yahoo, Amazon.com, Netflix, IMDb, Tripadvisor and Last.fm recommender systems play a very important role. The value of recommendations of a recommender system is shown in the table:

Website	Importance
Amazon	35% sales are from the recommendations
Google News	38% more clicks are generated from the recommendations
Netflix	2/3 of the watched movies are recommended

Table 1.1: Value of recommendations of a recommender system

Recommender system can be developed using a single or combination of more than one approach. There are five different approaches to develop a recommender system which are explained as follows:

1. Content based recommender system [1]: recommender system recommend items to the user that are similar to the ones that user liked in the past. Features associated with different items are compared to recommend similar type of items.
2. Collaborative filtering based recommender system [2]: recommender system recommend items to the active user that other user with similar type of taste liked in the past.
3. Knowledge based recommender system [3]: recommender system recommends items based on the specific domain knowledge about how certain item features meets the user preferences and needs.
4. Demographic recommender system [4]: recommender system recommends items that are based on the demographic information such as age, gender, country, etc.
5. Hybrid recommender system [5]: recommender system recommends items based on the combination of the above mentioned approaches.

1.1 Context aware recommender systems (CARS)

Context is a multifaceted concept and is used across various disciplines. There is different perspective of the concept for each discipline. The definition of Dey et al. in 2001 was probably the first broadly adopted in AI and ubiquitous computing: "*Context can be considered as any piece of information that can be used to characterize a situation of any entity. An entity can be an object, person or place that is considered relevant to the interaction between an application and a user*".

Context aware recommender systems [6] extend the traditional formulation of a recommender system problem by incorporating third dimension to the user-item interactions. The importance of contextual information is realized by many practitioners and researchers in many domains and in the past few years context-awareness has been increasingly explored by many recommender system researchers.

In the traditional recommender systems [7], only two things items and users are considered while providing recommendations to the user and can described by using the rating function S as follows:

S: User × Item → Rating

Adding contextual information to the recommendation process adds one more dimension to the recommendation process and it becomes a three-dimensional recommender system. Now rating function S' depends upon items, users and contexts and is described as follows:

S': User × Item × Context → Rating

The main research challenges [8], issues and directions are classified into the four following categories:

1. Fundamental
2. Algorithms
3. Evaluation
4. Engineering

Most of the existing research on context aware recommender systems is under the algorithm category. Many techniques to develop a recommender system have been proposed that take advantage of contextual data using anyone of the following algorithmic paradigms [9] to develop a context based recommender system:

1. Contextual pre-filtering: In contextual pre-filtering, contextual information is used for data pre-processing. In the process of pre-filtering data in the dataset which is not relevant for the target situation is discarded and we can use remaining data to give recommendations.

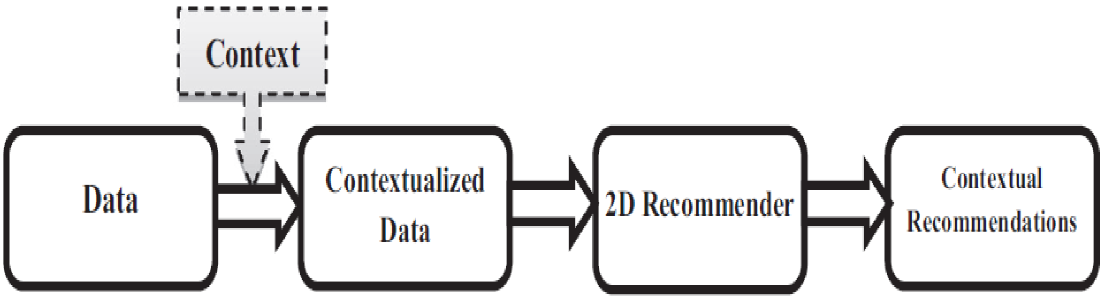


Figure 1.1: Contextual pre-filtering

2. Contextual post-filtering: In contextual post-filtering, contextual information is ignored during the prediction generation. The basic idea behind contextual post-filtering is to firstly apply traditional recommendation algorithms on the domain dataset then contextual data is applied on the predictions generated.

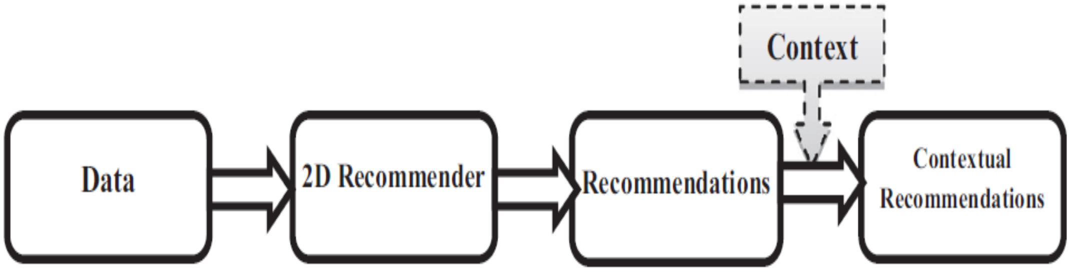


Figure 1.2: Contextual post-filtering

3. Contextual modeling: In contextual modeling, context aware information is applied directly to the dataset along with the information about users and items.

Figure 1.3: Contextual modeling

1.2 Location based search

With the advancement in location-acquisition technology [10], the location-based applications have gained popularity in this century. Today almost every mobile phone is Global positioning System (GPS) enabled [11] which help in acquiring the user location information. The location-based service provides the useful information to a mobile user. Huge amount of trajectory data is being generated at a very high rate. There are two different types of trajectory data. The first is the sequence of time-stamped locations which are generated by the GPS receivers and mobile devices such as mobile phones at a very high sampling rate. The sequence of locations with semantics [12] is a second type of trajectory where all the recorded locations are important. Geo-tagged photographs are the one such example of such a trajectory which is taken by a traveler in a trip. The location-based social network services provide these trajectories such as FourSquare (foursquare.com). There are number of web applications on the internet with copious amount of store information such as business hours, store location, public evaluation score, user commentary, and price level, which is useful for search engines to mine and get useful information. Location based recommender should be secured [13] so that access to user's personal information is possible in a secured way.

Most of the traditional recommender systems still have three main problems [14]:

1. The lack of location information: this problem occurs in recommender systems when they are not capable of taking location information of a user. Thus incapable of providing recommendations of nearby places to the user.
2. The lack of temporal information: this problem occurs in recommender systems that are not capable of undertaking query time information into account and hence recommender cannot predict whether the store is opened or not.
3. The lack of preference information: this problem occurs in recommender systems when they cannot predict the preference of a store for user due to no lack of user preference information.

In addition number of research and studies have been proposed and done for location-based search approaches [15], but in the recommender systems our main concern is to rank nearby stores based on mobile user location.

1.3 Rule-based systems

A rule-based system in computer science is used to store and manipulate knowledge in order to interpret information in a useful way. Rules of a rule-based system are used most often in artificial intelligence research and applications. In a rule-based system a classic example is the domain specific expert system that uses rules to make choices or deductions. Rules are of the form

IF some condition ***THEN*** some action

Conventional computer programs make use of data structures, crisp reasoning strategies and well structured algorithms to find solutions. The general association of causes with effects is described by a rule. Associations like ***if-then*** of imperative or conventional programs are hard-coded into a control flow whereas associations of a rule-based systems are soft-coded as a set of rules known as rulebase. Importance of using rules in a rule-based system is described in following points:

1. Rules provide interoperability across various platforms.
2. Rules are soft-coded not hard-coded like in imperative programs.
3. Rules provide data independence because rules are formally a text.
4. Knowledge obtained from rules can be used to query a domain dataset.

A rule in a rulebase is a conditional statement that links given conditions to outcomes or actions. A rule-based system consists of a collection [16] of if-then rules that are stored in a rulebase, an interpreter that controls the application of the rules, and a bunch of facts. Conditional statements are formulated using if-then rule statements that comprise the complete knowledge base. A single if-then rule statement assumes the form ‘if a is X then b is Y’ and the if- part of the rule ‘a is X’ is called the premise or antecedent while the then-part of the rule ‘b is Y’ is called the conclusion or consequent.

There are two types of inference engines in a rule based systems: backward chaining and forward chaining systems. Initial facts are processed first in a forward chaining system, and keep on using the rules to draw new conclusions given those facts. The hypothesis (or goals/ solution) that we are trying to reach are processed first in backing chaining and keep on looking for new rules that would allow us to conclude that hypothesis. New sub goals are set for validations as the processing progresses. Backward chaining systems are goal driven while forward chaining systems are primarily data-driven. Forward chaining is useful especially in situations where data is expensive to collect and it is also less in quantity. However, special care should be taken while the construction of these rules. Pre-conditions should be precisely specified when different rules should fire. We should use backward chaining where quantity of data is very large. Also in situations where there is lack of knowledge what the conclusion would be or if there is specific hypothesis to test, it is inefficient to use forward chaining.

1.4 Importance of tags in recommender systems

Today items of interest in the web can be organized and retrieved by using a popular means known as user-contributed tags [17]. Social tagging not only plays an increasingly important role on social web platforms such as Flicker and Delicious but also on large scale e-commerce sites such as Amazon. The main purpose of such a recommender system is to provide personalized buying proposals to its users.

Many different proposals have been made in recent years to exploit social web based tagging [18] information in order to build more effective or accurate recommender systems. The tagging data can be viewed as additional information about the items to be recommended or can also be viewed as a data to identify similar users. Recent research has shown that “attaching feelings to tags” by a user is experienced as a valuable means to show which features of an item a user particularly dislike or like. By using this approach, users can not only add tags to an item as it is seen in web 2.0 applications, but can also attach affect or preferences to the tag itself. For example, whether they liked or disliked a certain actor in a movie, if they have liked or disliked then up to what extent.

Tagging information can be exploited in different ways in the recommendation process. Tags can be seen as an item description and be used by any traditional recommender system for example it can be used by a content-based recommender system. An item can have many features and tag data can be used to identify which features a particular user dislike or like about the item. The relationship between tags and users can be used in collaborative filtering systems to find similar users in a neighborhood. The main goal of many different tag based recommender systems is to exploit the existing interactions among items, users and tags can be used to improve effectiveness of the recommender systems. In the most of the existing approaches to tag-enhanced or tag-based collaborative item recommendation, the main assumption of the recommender systems is that preference information provided by the different user community is available for the items.

1.5 Ranking Algorithm

A ranking function [19] is a function that can be used in the process of information retrieval. The ranking function is used by various search engines to rank items according to their relevance to a search query. Relevant items once identified by the search engine, out of those relevant items the task to determining most relevant items is to be done. This task is typically done by assigning a numerical value to each item. Ranking functions can be evaluated by variety of ways. One of the simplest ways is to determine the first k top ranked results [20] for some value of k. The value of k can be user defined or pre-defined based on that top-k items are presented to the user.

CHAPTETR 2
LITERATURE SURVEY

2.1 Context-aware recommender systems

The CARS [21] has been formed as a result of recognition of contextual information in recommender systems. Different approaches to develop a context aware recommender system have been proposed by Adomavicius and Tuzhilin which are categorized into post-filtering, pre-filtering and contextual modeling.

The concept of context was first introduced by Schilit, Admas, and Want [22] and has significantly evolved since then. It is widely accepted that there can be diverse kinds of contextual information which is classified into three main categories: device context, user context and physical context depending on the perspective adopted: application or user side. The interpretation of context varies between one application to another application according to the architecture and domain of the application so that a definitive definition can be given of this concept. For example, Adomavicius and Tuzhilin [23] described interpretations of context across different fields concerned with recommender system such as ubiquitous systems and e-commerce. Lee, Kaoli, and Huang [24] proposed the social tag-based collaborative filtering approach as part of a smart TV system. This approach considers both device and user context as a context information. The recommendation in this approach are only computed on the user preferences and re-ranked accordingly. In this location depends and defined on the appropriateness of multimedia content for a particular place in which user is accessing the system. Crowd and time are considered as recurrent kinds of contextual information. A genetic algorithm for mobile-based advertising based on collaborative filtering approach is presented by Dao, Jeong, and Ahn [25]. Context awareness is addressed in this approach by calculating similarity between contexts of different users. In this approach context is the combination of motivation and time factors whereas user location is individually considered. In detail time is the time and day a user might go to visit a place. In this they have made an assumption that a user visit a particular place may depends upon the period of the day, here visiting time is not computed as the time in minutes and hours but as a class that has different periods such as: morning, lunch, afternoon and so on.

In Table 2.1 a comparative analysis of CARS is reviewed in the literature. The following factors are considered in comparative analysis: (1) kind of recommender system, (2) filtering technique, (3) context-aware approach, and (4) context dimensions.

Sr. No	**Work**	**Kind of recommender system** **KB CFB CB**	**Context-aware approach** **PrF PoF CM**	**Context dimension**
1	(Lee et al., 2014)	X (user-based)	X	Location type, mobile device, crowd and network conditions
2	SMARTMUSEUM (Ruotsalo et al., 2013)	X X(item-based)	X	Out-door scenario: location, in-door scenario: companion and motivation
3	iTravel (Yang & Hwang, 2013)	X (user-based)	X	Location
4	SigTur/E destination (Moreno, Valls, Isern, Marin, & Borràs, 2013)	X(user-based) X	X	Location

5	Turist@ (Batet, Moreno, Sánchez, Isern, & Valls, 2012)	X(user-based) X	X	Location
6	Fang et al. (2012)	X	X	Location
7	SR-REJA-3D (Noguera et al., 2012)	X X(user-based)	X X	Location
8	Route recommendation system (Tsai & Chung, 2012)	X(item-based)	X	Location, time and crowd
9	iConAwa (Yılmaz and Erdur, 2012)	N/A- Expert System using Rule-based reasoning	X	Location
10	Group corrections-driven filtering (Blanco-Fernández et al., 2011)	X X(user-based) X	X X	Time
11	Wang and Wu (2011)	X (user-based)	X	Location

Table 2.1: Comparison of proposals for CARS (KB = knowledge-based, CFB = collaborative filtering-based, CB = content-based; PoF = post-filtering, PrF = pre-filtering, CM = contextual modeling, and Context dimensions) [27]

In [26] proposed a data mining based approach named as Preference oriented location based search (POLS) to efficiently search k nearby stores that are preferred by the user based on the user's location, query time and preference. In POLS, two preference learning algorithms are proposed that automatically learns user preference and ranking function is also proposed to rank the nearby stores based on the user's location, query time and preference. In [27] proposed a context aware recommender system called RecomMetz that recommends movie showtimes based on three different kinds of contextual information crowd, location, and time. Items to be recommended in this recommender system have a composite structure (movie_theater + showtime + movie). It is based on semantic web technologies and implements an ontology based context modeling approach. RecomMetz supports by multi-platform mobile user.

2.2 Context awareness through rules

In [28] Geographical Semantic Personalized Location Information/ Geo SPLIS provide user the capability to add their own rules. Rules contain the contextual information and rules are defined using RuleML. Contextual information defined in the rules is used to recommend nearby locations to the user based on user location. In the Geo SPLIS work, user preferences in the form tags are not considered. We are using the basic concept of rules and tag preferences of the user to recommend nearby places based on the user location. A situation aware service recommender is proposed by ciaramella, Lazzerini, Cimino and Marcelloni [29]. Each condition is determined by using rules in SWRL format based on the contextual information obtained from each rule and based on that a set of related tasks is recommended. Kebler, Wosniok and Raubal [30] combined data from rules in SWRL format and sensors with ontologies which is used to provide personalized recommendations to a user for surf spots. Sem-Fit is another example of this area by García-Crespo [31], which uses rules defined in fuzzy logic to give recommendations of hotels to a user. The rules are updated based on the customer evaluation of the returning results to provide better results next time. A tourist informative service is proposed by Serrano, Bravo, and Hervás [32] which combines RDF data gathered from sources such as foaf profile that uses rules defined in SWRL format to recommend places of interest (POIs) related to a user profile. A service for smart devices, work done by Furno and Zimeo [33] to propose a model by extending the OWL-S ontology for context aware recommendations service composition. Many researchers also make use of data from various social networking sites Facebook, Twitter, foaf etc. to achieve or recommend better recommendations. A person matching application is proposed by Woensel, Casteleyn, and Troyer [34] based on the data retrieved from a foaf profile and by using identification techniques such as Bluetooth and RFID etc. Photo Map developed by Viana, Gensel, Filho, Villanova-Oliver, & Martin [35] uses SWRL rules to attach social and physical context to photo shots for example who was there and where the photo was taken.

2.3 Geo-recommender system

The ability of a recommender system to recommend possible places of interest to a user by taking into account both the geographical location [36] of the places and current geographical location of the user. According to this definition of geo-recommender system [37], we can take into consideration dynamic nature of user's location. This allows mobility, a concept generated from mobile computing. This is how mobile computing and recommender system converge in geo-recommendation systems research.

Some of the existing proposals supporting this are discussed below. A recommender system is proposed by Fang et al. [38] that are used for assisting in-door shopping. Due to the certain constraints of current geographical positioning systems for indoor usage a received signal strength pattern mining positioning method was proposed. The user's location can be obtained by a location server from the real time information sent by their mobile devices by measuring the strength of the signal received from nearby stations. The similarity between the real-time received signal strength information and predefined received signal strength patterns that are stored in a database in which received signal strength patterns represents the location of various stores. In a nutshell, location of the nearest store can be calculated as the location of each user. A location aware mobile recommendation system is proposed by Yang, Cheng, and Dia [39] for location based advertising or m-commerce. This system uses agent based architecture to catalog vendors' information such as WebPages links and physical addresses that need to be maintained by the vendors themselves. The recommendations of vendors' WebPages are recommended to users by doing a match between geographical location locations traced by the GPS receivers of the users' mobile devices with the physical addresses in the vendor catalog. In this context, the distance between each vendor's location and a user's location is calculated by using the formula of Euclidean distance metric. In addition to this, user profiles can be obtained through implicit preferences from visiting webpage histories in the recommendation process. A mobile recommender system for tourism activities known as tourist@ [40] is proposed by Batet, Valls, Moreno, Sanchez, and Isern. This system makes use of GPS receivers of mobile devices to track the location of users. The aim of this system is to provide pro-active recommendations along with on-demand recommendations that rely on an agent based architecture. In detail, an agent running in user mobile device can suggest an activity automatically to the user when the user is near the interesting activity. In Table 2.2, comparison between different geo-recommender systems is presented reviewed in the literature. The following factors are considered in the comparative analysis: (1) geo-recommendation technique, (2) geo-localization technology.

Sr. No.	Work	Geo-localization technology	Geo-recommendation technique
1	SMARTMUSEUM (Ruotsalo et al., 2013)	GPS receivers of mobile devices	Fixed-distance bounding box
2	SigTur/E-destination (Moreno et al., 2013)	Static location information	JST topology suite API-based
3	Fang et al. (2012)	Radio signal receiver of mobile phones	Received Signal Strength (RSS)-based positioning method
4	SR-REJA-3D (Noguera et al., 2012)	GPS receivers of mobile devices	Euclidean distance
5	iConAwa (Yılmaz and Erdur, 2012)	GPS receivers of mobile devices	Haversine formula
6	Route recommendation system (Tsai and Chung, 2012)	RFID readers	N/A

7	Wang and Wu (2011)	RFID reader-equipped mobile devices	N/A
8	Yang et al. (2008)	GPS receivers of mobile devices	Euclidean distance

Table 2.2: Geo-recommendation proposals comparison [27]

2.4 Tag-based recommender system

Tagging information can be exploited in different ways in the recommendation process. Tags can be seen as an item description and be used by any traditional recommender system for example it can be used by a content-based recommender system. An item can have many features and tag data can be used to identify which features a particular user dislike or like about the item. The relationship between tags and users can be used in collaborative filtering systems to find similar users in a neighborhood. The main goal of many different tag based recommender systems is to exploit the existing interactions among items, users and tags can be used to improve effectiveness of the recommender systems. In the most of the existing approaches to tag-enhanced or tag-based collaborative item recommendation, the main assumption of the recommender systems is that preference information provided by the different user community is available for the items.

In [41], proposed a recommender system that uses item specific tag preferences. The predictive accuracy of recommender system is enhanced by assigning preferences to tags. In this recommender system, user can attach tags to the items and can also give rating to each tag. User can also give overall rating to the item. The cosine-tag algorithm is used to calculate similarity among items.

A tag based recommender system whose goal is to leverage information the users' preference for individual tags to predict more precise recommendations. In the "tagommender" [42] user's attitude towards different tags is analyzed. In the movie domain the task is to determine of and up to what extent user bob likes movies that are attached with the tags "animated." After that rating prediction for item is based on the mean of the inferred user tag preferences to that item. It is analyzed that more precise recommendations can be made when user based tag preferences are taken into account.

CHAPTER 3
PROBLEM STATEMENT

Recommendation process in most of the recommender systems is considered to be 2-dimensional: Users × Items. Recommender systems have proved to be a useful tool in coping with information overload. It is very important to use correct approach to build a recommender system and it depends on the type of application. However, most recommender systems focus on recommending the most relevant items to users without taking into account any additional contextual information. Most existing information retrieval systems base their retrieval decisions solely on queries collections, whereas information about search context is often ignored. By adding contextual information to the recommendation process makes a recommender system 3-dimensional: Users × Items × Contexts.

3.1 Gap analysis

It is analyzed that most of the existing context aware recommender systems do not provide users with the flexibility to customize the contextual information because most of the contextual information is either pre-defined or it is difficult to change the contextual information as user needs changes. So sometimes a user may get recommendations that are neither valid nor interesting from user point of interest because those recommendations do not match his/her context. Also existing offering strategies are not interoperable among different systems.

Most of the recommender systems do not consider user location while providing recommendations. Due to lack of location awareness in the recommender systems users may get recommendations that are not within their locality. Also there are many recommender systems that do not consider time information about the various stores while giving recommendations so a user may get recommendation of a store that might be closed at that time.

Recommender systems increasingly use contextual and demographical data as a basis for recommendations. Users, however, often feel uncomfortable providing such information. In a privacy-minded design of recommenders [27], users are free to decide for themselves what data they want to disclose about themselves. But this decision is often complex and burdensome, because the consequences of disclosing personal information are uncertain or even unknown. Although a number of researchers have tried to analyze and facilitate such information disclosure decisions, their research results are fragmented, and they often do not hold up well across studies.

3.2 Problem statement

In the thesis, the main problem being addressed is formulated as follows:

Given a user location and contextual information in the form of rules, our goal is to develop a system that provides ranking list of nearby places of a particular domain. The ranking function proposed to rank nearby places includes four things: user distance preference, rule priority, public evaluation and user tag based preference to a place. The ranking function ranks various nearby places and Top-K places are selected from the ranking list.

CHAPTER 4

PROPOSED SOLUTION

4.1 Proposed solution

The proposed model recommends nearby places to the user based on the user-defined rules and the active user current location. While providing recommendations to the user nearby places are ranked according to the ranking function and only top-k recommendations are presented to the user on the Google map.

The proposed model of our recommender system is shown in figure 4.1 with the workflow of the model.

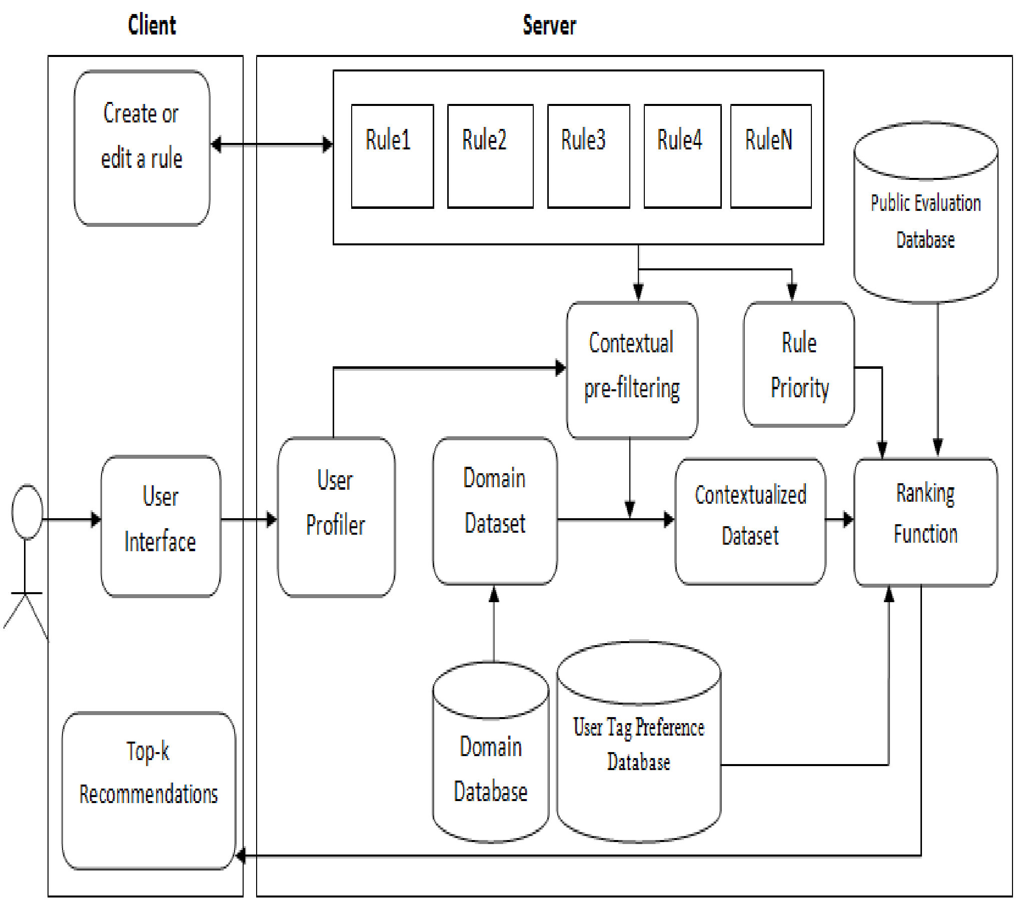


Figure 4.1: Proposed Recommendation Model

4.2 Algorithm

Input: I= {L, K, DomainDB} is the set consists of user location L= {Long, Lat}, k decides the value of K in Top-K items, and DomainDB is the domain database.

Output: Top-K recommendations are presented on the Google map.

for all rules R_i ε R do

1. XML_Reader ← get_rule(R_i)
2. Context_array[C_j] ←XML_Reader

for all C_j ε Context_array[C_j]

3. Pre_contextualized_DB ← DomainDB.getTuples(C_j)

End for

4.Rank the Pre_contextualized_DB data set using the ranking function

$R(u, p) = R_{Dist}(u, p) + \text{Rule priority} + R_{User\text{-}pref}(u, p) + Pub\_Eval(p)$

For all p ε Top-k places for rule R_i in R(u, p)

5.Google-map ← place (p)

End for

End for

Where

R= {R_1, R_2, R_3, R_4, R_N} where N is the number of rules

Context_array[C_j] = {$context_1$, $context_2$, $context_3$, $context_M$} where M is the number of contexts defined corresponding to rule R_i.

Table 4.1: Algorithm

4.3 Data Flow Diagram

The data flow diagram that shows how a user going to interact with our proposed model is shown in figure 4.2. The user needs to create an account by filling a registration form. At the time of registration a directory is created for the user. In the system the user can create, edit or delete rules. The user can click on show places button to get the recommendations of nearby places from the user location.

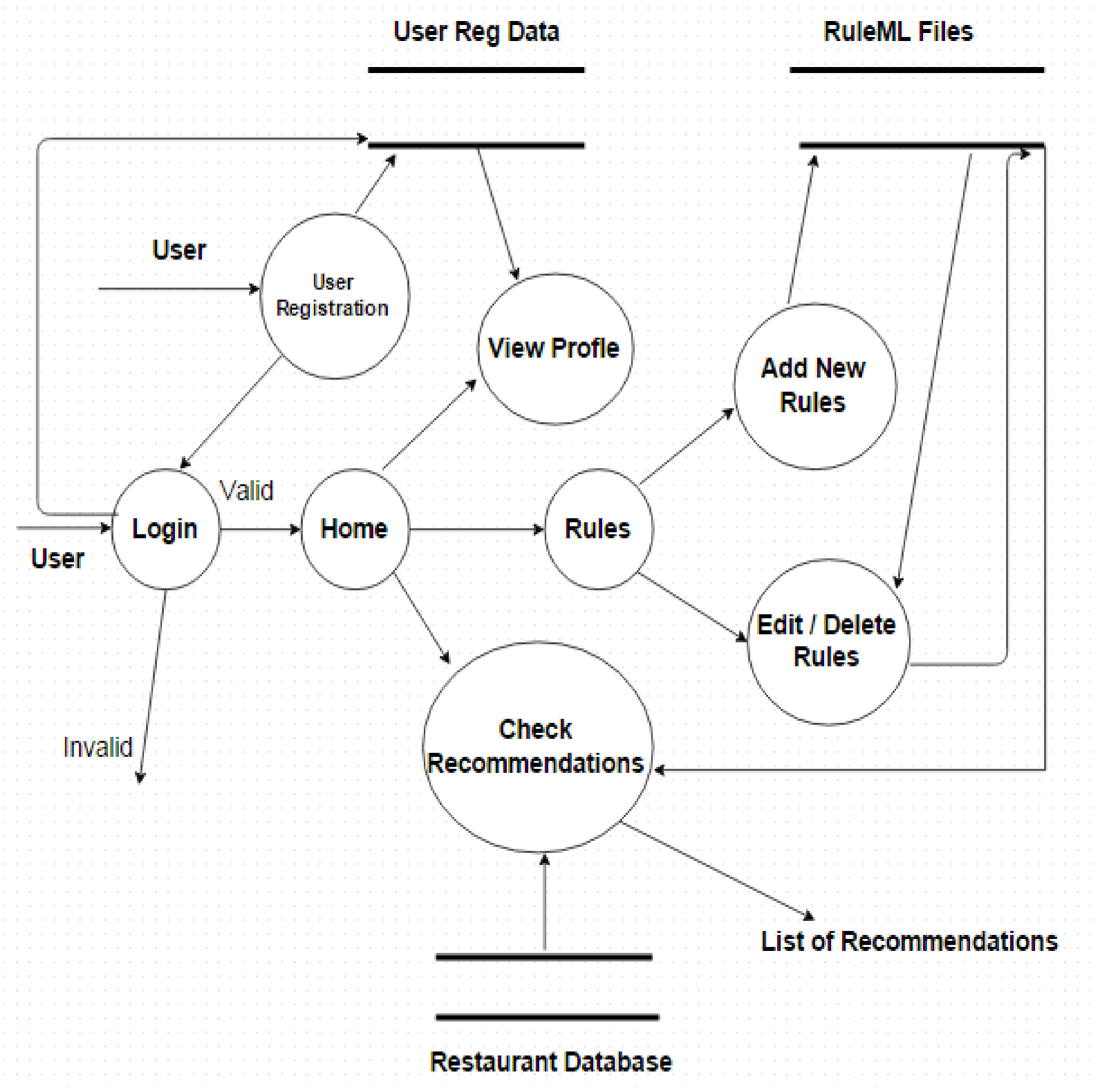


Figure 4.2: Data flow diagram of the proposed work

4.4 Overview of the proposed model

4.4.1 User profiler

The user profiler receives information sent by the user interface. The user interface sends active user current location, current weather status and time information. All the information received by the user profiler is used in the contextual pre-filtering process.

4.4.2 User defined rules

User can create a new rule through a user defined web form. Data collected from using a web form is transformed into RuleML rules. XML writer is used to transform collected data into RuleML rules. After that, transformed RuleML rule is stored in the specified directory of the user which is created for the user at the time of registration. The user specified directory contains a set of rules R= $\{R_1, R_2, R_3, R_4, R_N\}$ where N is the number of rules defined by the user. $C_{USER}= \{C_1, C_2, C_3, C_4, C_N\}$ is the set of contextual data corresponding to each rule. For example $C_1= \{context_1, context_2, context_3, context_M\}$ where M is the number of contexts defined corresponding to rule R_1.

4.4.3 Contextual filtering

Contextual filtering is requires contextual data. When an active user needs recommendations of the nearby places then contextual information contained in the user defined rules is extracted using XML reader. The extracted contextual information is then applied on the domain dataset to filter out the places that do not satisfy the criteria defined in the user rules.

4.4.4 Location related pre-filtering

The calculation of the distance between the user current location and nearby places is a complex task because calculated distance should have to be optimal for a given transport facility. Euclidean distance between two points does not give optimal solution. Google Distance Matrix API is used in our work to calculate the distance between user location and various nearby places. Google Distance Matrix API allows calculation of travel distance and travel time between locations based on the recommended routes. It supports many modes of transportation including driving, cycling, walking, and transit. The calculated distance between user and various nearby places is used during the process of contextual pre-filtering. All the places that have distance more than the distance that user wants to travel are discarded in the resultant set.

4.4.5 Time related pre-filtering

The following equation calculates the time required to reach the restaurant T_P:

$T_P= P_{CT}[i] – (current\_time + P_U[i])$

Where $P_{CT}[i]$ is the closing time of the restaurant and $P_U[i]$ is the time required by the user to reach the restaurant. Now if $T_P \leq 0$ then the restaurant is discarded form the dataset

4.4.6 Ranking function

Contextualized dataset is ranked using a ranking function. Ranking function is used to rank the nearby places after the dataset is contextualized. The ranking function R (u, p) used in the system to rank nearby places is defined as follows:

$$R(u, p) = RDist(u, p) + \text{Rule priority} + R_{User\text{-}pref}(u, p) + Pub\_Eval(p) \quad (1)$$

Pub_Eval (p) can be calculated based on the different ratings given by the users to place p. Average of all the ratings is calculated and average score is normalized between 0 and 1.

To calculate R_{Dist} (u, p) we use the user preference database. The user preference database that contains the historical log of the places visited by the user, we calculate D_{Pref} (u, p) the average distance that user has preferred to a place in the following equation:

$$D_{Pref}(u, p) = \frac{\sum_{i=1}^{t} D\_Act\_i(u,\ p)}{t} \quad (2)$$

Where $D\_Act\_i(u,\ p)$ is the actual distance between the place p and locations of the user u and t is the number of times user visited the place p. The main concept behind the calculation of distance score is that father place might not preferred if there is closer place with the similar features is available.

$$R_{Dist}(u, p) = \begin{cases} 1, & D_{Act\,(u,\ p)} \leq Dpref(u,p) \\ 1 - \frac{D_{Act\,(u,\ p)}}{\max distance}, & otherwise \end{cases} \quad (3)$$

Where $D_{Act\,(u,\ p)}$ the actual distance between user u and place location p. $Dpref(u,p)$ is the distance defined in equation (2). By default value of max distance is 5km but it can be changed on the bases of max distance defined by the user in the rules.

Rule priority is an integer that is given to rule at the time of rule creation that shows the importance of a rule. Lower the integer value higher will be its priority. There is a threshold value of 20 to limit the value of priority.

$R_{User\text{-}pref}$ (u, p) is the preference matching score between the preferences of the user tags and tags [17] attached to a place p. A set positive tags P_tag = {tag1, tag2, tag3, …, tag_P} where p is the number of positive tags and a set of negative tags N_tag = {tag1, tag2, tag3, …, tag_K} where k is the number of negative tags maintained by the user u in the user preference database. A user can give rating value to each store. User can rate both negative and positive tags. The rating scale is 1 to 5 means form lowest to the highest preferred. Table 4.2 and 4.3 shows the positive and negative tags for user u. For example, Service (5) is the positive tag for user u represents that the user u prefers places where ambience is extremely good and Crowd(5) is the negative tag for user u represents that user u has a strong dislike for the crowded places.

Positive Tags	Negative Tags
Service(5) Ambience(4) Clam(5) Buffet(4) Spicy(3)	Crowd(5) Parking(3) Cheap(4)

Table 4.2: Positive and negative tags for a user u

Place X	Place Y
Service Clam Cheap Spicy	Ambience Parking Spicy

Table 4.3: Tags corresponding to place A and place B

The tag based preference matching between place X and place Y is calculated by using the formula:

$$R_{User\text{-}pref}(u, p) = \frac{\sum_{PTR\ \varepsilon\ T_{mp}} PTR_u - \sum_{NTR\ \varepsilon\ T_{np}} NTR_u}{\text{P\_tag} + \text{N\_tag}} \quad (4)$$

Where PTR_u is the rating value corresponding to positive tag and NTR_u is the rating value corresponding to negative tag for the user u. T_{mp} is the set of tags attached to a place p and matched with user positive tag preference set P_tag. T_{np} is the set of tags attached to a place p and matched with user negative tag preference set N_tag. The tag based preference matching scores for place X and Y can be calculated as follows:

$R_{User\text{-}pref}(u, X) = [(5 + 5 + 3) – (4)]/10 = .9$

$R_{User\text{-}pref}(u, Y) = [(4 + 3) – (3)]/10 = .4$

This shows that user u has more preference to place X than place Y.

4.4.7 Present Top-K places on the Google map

The ranking function consists of four parameters and calculation of weight of these four parameters is done in equation (1), (2), (3), and (4). Weights of all these four equations are added to get the overall weight. Top-K places in the ranking process are shown on the Google map. To present the Top-K places we need to convert addresses of various places into geographical coordinates and then show those places on the Google map via Google Places API.

4.4.8 Geocoding API

Geocoding is the process of converting addresses into the geographical coordinates such as longitude and latitude. Top-K places addresses are converted into geographical coordinates using Google Geocoding API. In our work we have used dynamic Geocoding. The Google Maps API provides a Geocoder class for Geocoding dynamically from user input. To dynamically convert addresses of various places after the ranking process JavaScript is used. JavaScript dynamically converts addresses into geographical coordinates i.e. latitude and longitude.

4.4.9 Google Places API

The dynamically converted addresses into geographical coordinates are shown on the Google map. To show places on the Google map Google map API is required. Top-k places after the ranking process are presented on the Google map to the user based on the user current location.

CHAPTER 5
IMPLEMENTATION AND RESULTS

5.1 Testbed

Rules are created Using Rule Markup Language abbreviated as RuleML. To create rules using RuleML basic knowledge of RuleML is required and also how RuleML uses XML schema is thorough understanding is required. Thorough study of various API's provided by the Google such as Distance matrix API, Geocoding API, and Google Place API is required. MySQL server is used to store domain database. Restaurant dataset is considered as a domain dataset. To develop a web application JSP is chosen. NetBeans IDE is required to develop JSP web pages.

5.2 Prerequisite

The following are the prerequisite required for the development of the system:

1. To collect user contextual data a user friendly web form is needed for that JSP is used to create a web form.
2. Distance matrix API is required to calculate the distance between user location and various nearby places.
3. XML writer is required for the creation and XML reader is required to read a rule.
4. Geocoding API is required to convert address of a place into geographical coordinates such as longitude and latitude.
5. Google Place API is required show places on the Google map.
6. Domain database is stored and retrieved using MySQL server.

5.3 Tools and API required

The following are the various tools and API's required for the development of the system:

1. NetBeans IDE 8.0.2
2. MySQL
3. XML reader and writer
4. Geocoding API
5. Distance Matrix API

5.4 Use Scenario

In this section, consider the following use scenario explained to present our contextual pre-filtering paradigm used in our recommendation process, followed by ranking of nearby places on the basis of our ranking function in equation (1). The Top-k places after the ranking process are presented on the Google map.

Although the proposed recommender system can be used to recommend various type of nearby places. In this scenario, we recommend nearby restaurants and ongoing offers to the user. So in the use scenario restaurant domain is considered. In our day to day life, people need information about restaurants. The information presented to user must be based on user interest and also need to match with the user context.

When an active the user needs recommendations about nearby restaurants at that time the user specifies how many recommendations he wants on the bases of that value of k is decided but by default value of k is set to 10 in the system. The maximum value of k is 20.

People have preferences in everyday's life, such as "if day is Monday and weather is cloudy then I would like to go to a coffee shop" or "if it is Friday, time is between 6 p.m. -10 p.m. and distance is less than 5 km then I would like to go to a Dominos". These types of preferences are not completely random. These preferences present strong daily patterns.

In order to collect contextualized information a rule based language is needed and RuleML is chosen for this purpose. A user friendly web form is shown in figure 5.1 which the user can use to add their preferences. Every time the user logs into the system and clicks on show recommendations button, system gets his/her context (weather, time, location, and day), system evaluates his/her rule based preferences and finally presents Top-k personalized recommendations based on the ranking function on the Google Map.

Contextual information changes as the user mood changes. To reflect changes in the contextual information a user friendly form to update or delete rules is shown in figure 5.2. The User can edit or delete a rule stored in the specified directory for the user.

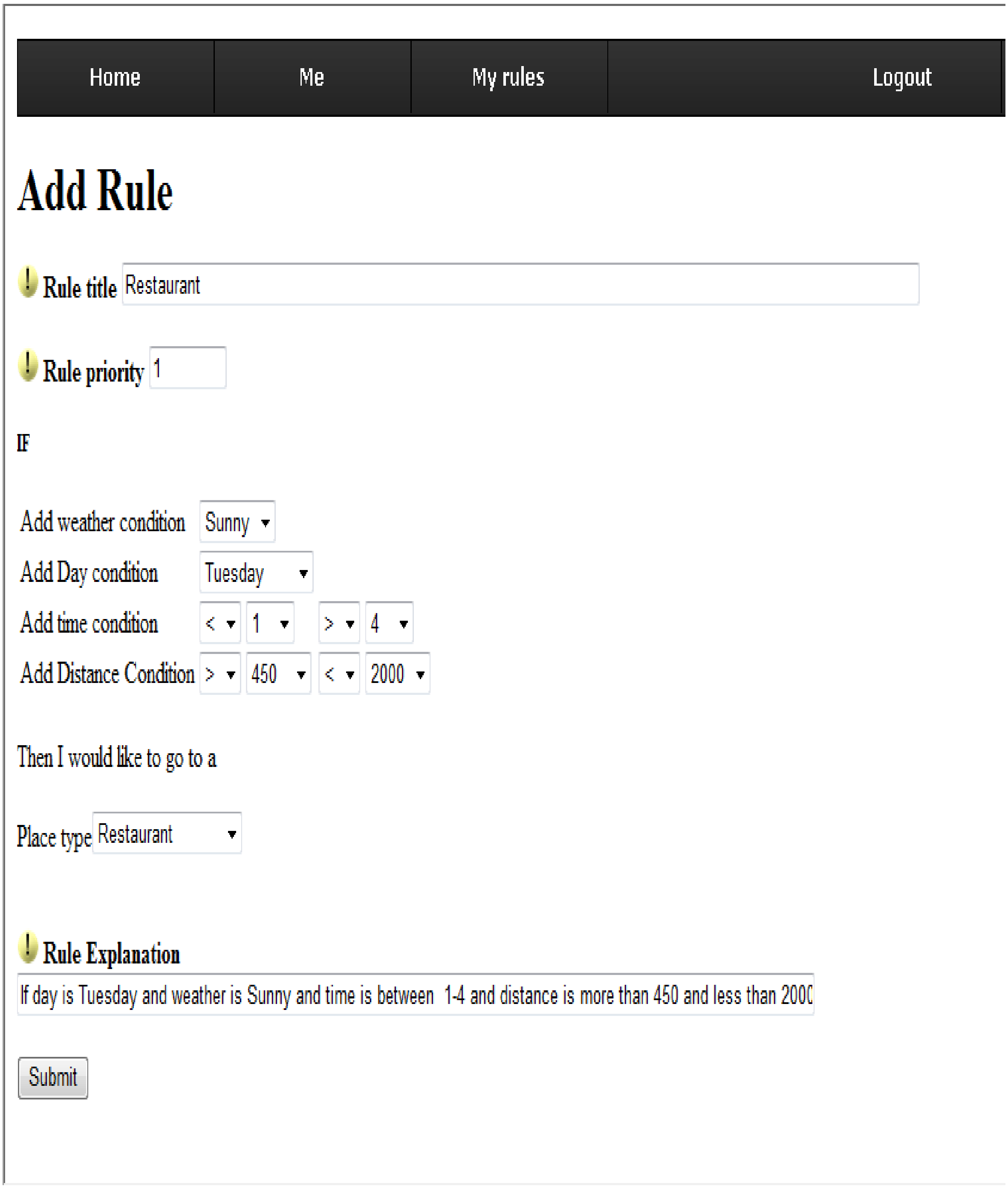


Figure 5.1: A web form to add user based preferences

The user needs to provide rule priority of the rule. Various contextual data such as day, weather, time, and distance. The place type in the current scenario is restaurant. When the user has filled all the information rule explanation is automatically generated. Once the user clicks on the submit button, the contextual information provided by the user is converted into a RuleML rule.

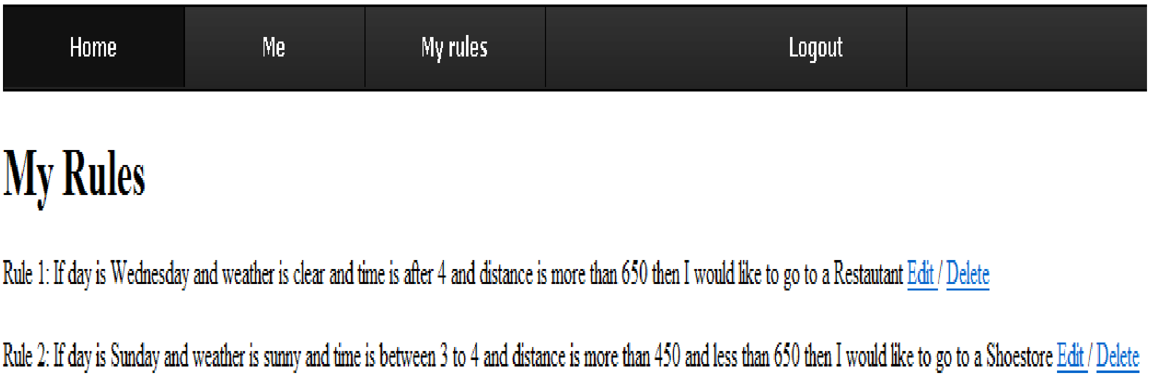


Figure 5.2: A web form to edit or delete a rule

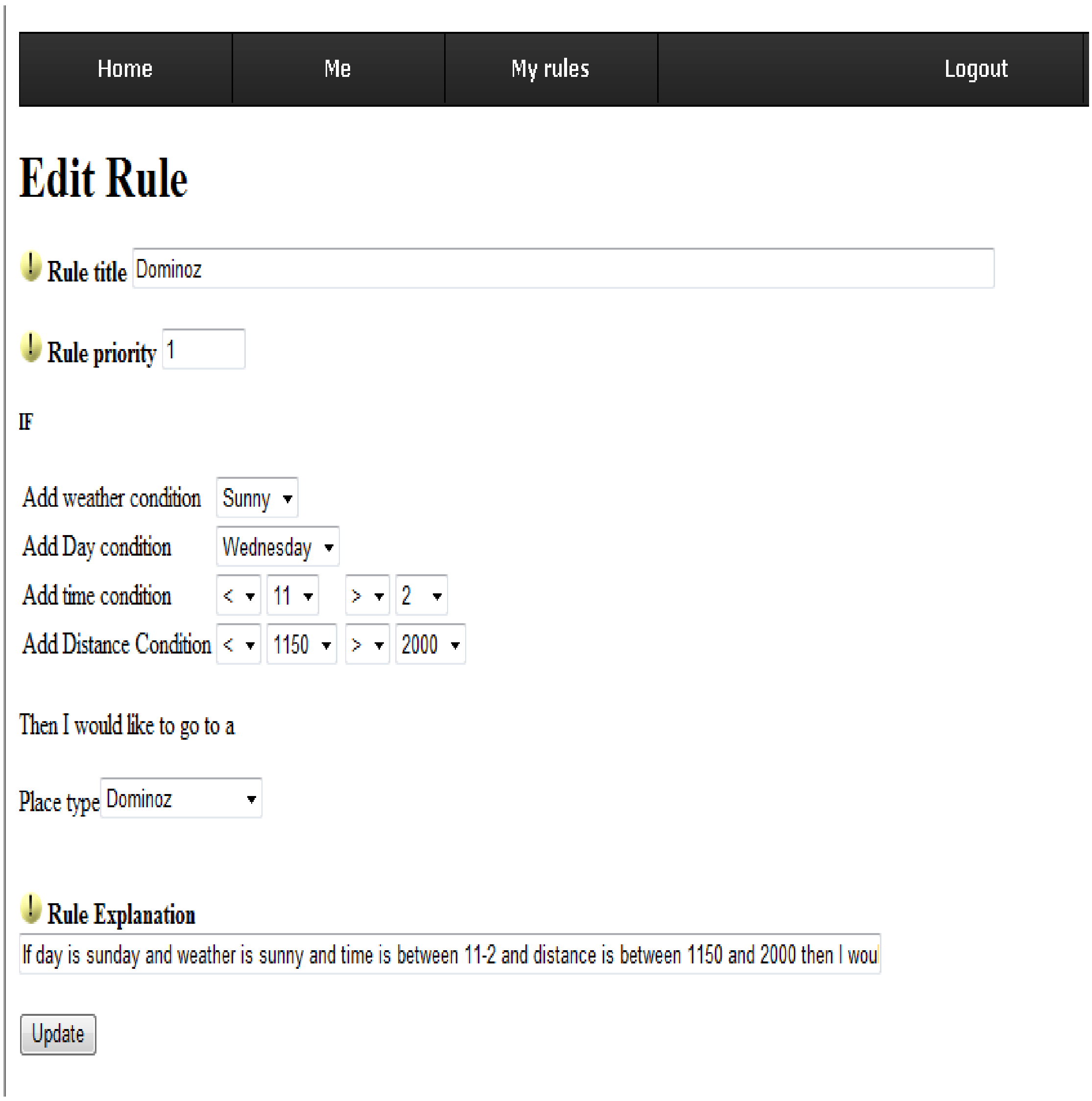


Figure 5.3: Edit a rule

5.4.1 RuleML Rule

RuleML uses XML based schema and provide interoperatability on the web among various systems. Contextual information collected via web form is used by the RuleML to create RuleML rule. Suppose a user defined rule "*If weather is sunny, day is Wednesday, time is between 5 p. m.-10 p.m. and distance should be less than 5 km then I would like to have pizza*". The User set of contextual data for this particular rule is {weather, day, time, distance, food_type}. RuleML rule created for the above rule using a XML writer is shown in table 5.2. Once the rule is created by the XML writer, the rule is stored in the specified directory created for the user at the time of registration. Rules stored in directories are shown in figure 5.3.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<RuleMl>
<Atom>
<Rel>Place</Rel>
<Slot><Ind>Type</Ind><Ind>Restaurant</Ind> </Slot>
<Slot><Ind>Food_Type</Ind><Ind>Pizza</Ind> </Slot>
</Atom>
<Atom>
<Rel>User</Rel>
<Slot><Ind>Day</Ind><Ind>Wednesday</Ind> </Slot>
<Slot><Ind>Weather</Ind><Ind>Sunny</Ind> </Slot>
<Slot><Ind>Time</Ind><Ind>more</Ind><Ind>5 </Ind>
<Ind>Time</Ind><Ind>less</Ind><Ind>10</Ind> </Slot>
<Slot><Ind>Distance</Ind><Ind>less</Ind><Ind>5
</Ind></Slot>
</Atom>
</RuleMl>
```

Table 5.1: RuleML rule

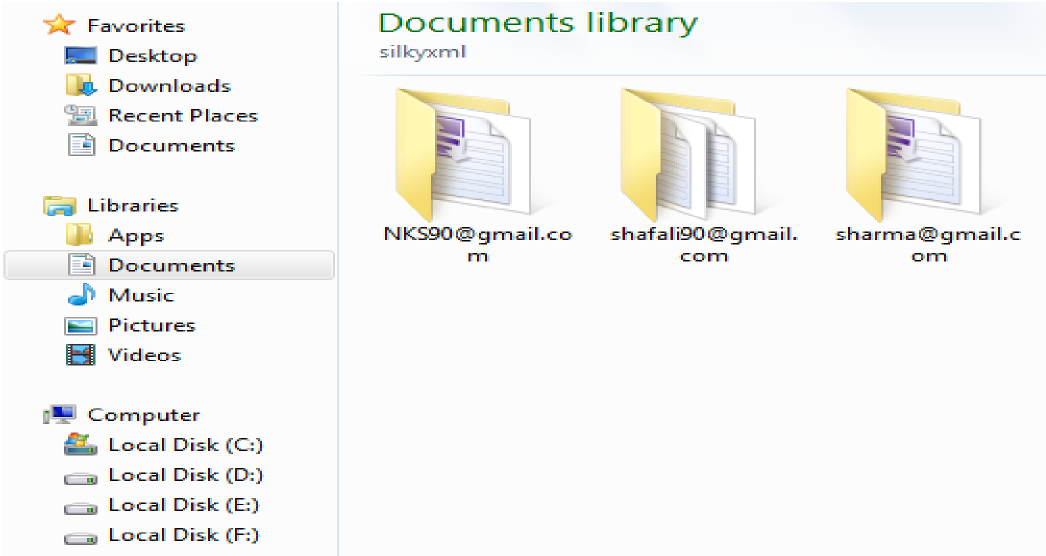


Figure 5.4: Directories of users in which RuleML rules are stored

5.4.2 Domain database creation

In the use scenario a restaurant database is chosen as the domain database. Restaurant database contains information about various restaurants in the city Patiala. Restaurant database contains total 9 tables is shown in figure 5.4:

1. Category: total 25 different food categories are considered.
2. Restaurant_info: contains information about various restaurants. Restaurant dataset considered in this scenario has 100 different places.
3. Rule_info: contains information about various rules stored in different directories
4. User_rating: contains information about rating given by different users to various restaurants.
5. Offers: contains information about various offers going on in restaurants.
6. User_info: contains information about users who have registered themselves in the system.
7. Tags_info: user positive and negative tags information is stored in the Tags_info table.
8. Distance_history: preferred distance chosen by the user to travel to a restaurant during different visits to the restaurant.
9. Restaurant_tags_info: contains information about various tags given to a restaurant or the tags for which the restaurant is famous.

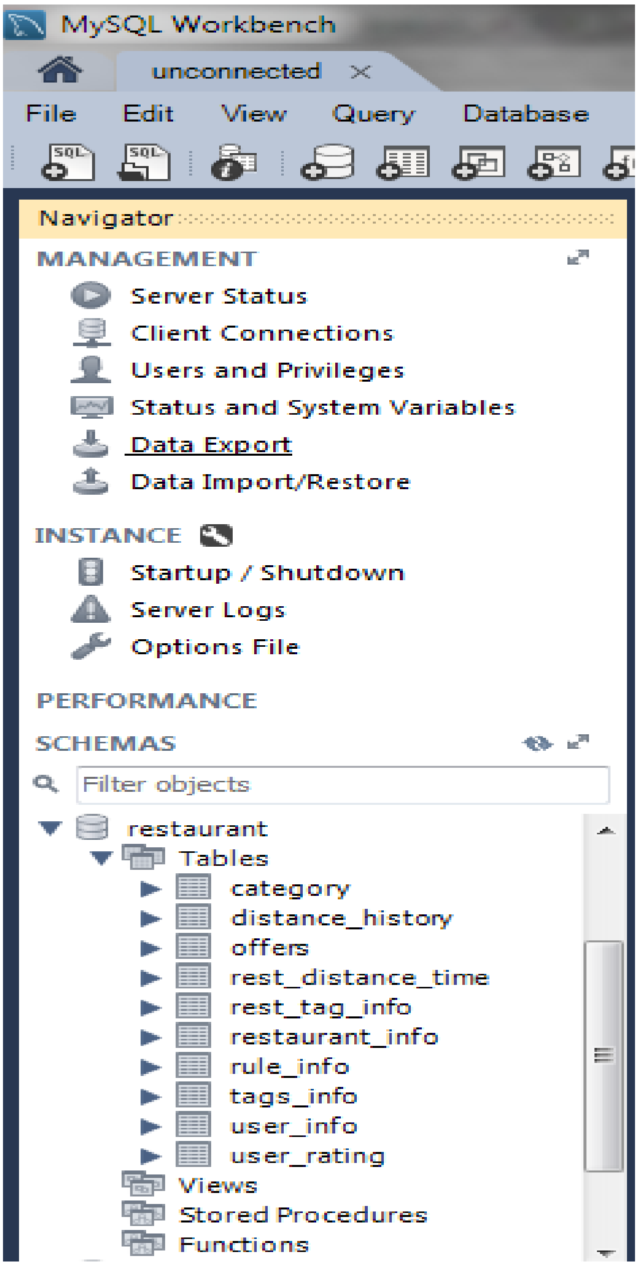


Figure 5.5: Tables in restaurant database

The restaurant domain dataset is used during the process of contextual pre-filtering. Contextual data obtained from the rules is used during the process of contextual pre-filtering. Contextualized dataset is then used in the process of ranking.

5.4.3 Contextual Pre-filtering

The restaurant domain dataset contains many places but only few of them are important from user point of context. Based on the contextual data obtained from the user defined rules contextual pre-filtering is done. To extract the contextual data from the user defined rules XML reader is used. XML reader read all RuleML rules defined by the user and contextual information obtained from all the rules. Information obtained from the rules is used in the process of contextual pre-filtering.

The pre-filtering process should be done in a particular order. The following is the order in which rule "*If weather is sunny, day is Wednesday, time is between 5 p. m.-10 p.m. and distance should be less than 5 km then I would like to have pizza*" contextual data should be applied:

1. First of all current weather condition is obtained from the weather web service. If current weather matches the weather condition defined in the user rule then we proceed further otherwise no further processing is required.
2. The restaurant domain database is filtered based on the type of food they serve. All the restaurants are discarded that do not serve pizza.
3. Based on the time context given in the user defined rule all the pizza restaurants that are closed by that time are filtered out.
4. Distance between the user current position and all the restaurants are calculated. Also the time required to reach each restaurant is calculated. To do this Google distance matrix API is used. Google distance matrix API calculates the distance between active user current location that is Thapar University Patiala and all the restaurants that are in the contextualized dataset after the step 3. From this place distance between various places and the user is calculated. Distance matrix API not only calculates the distance but also calculates the time required to reach each restaurant from the user location. The following equation calculates the time required to reach the restaurant T_P:

   $$T_P[i] = P_{CT}[i] - (current\_time + P_U[i])$$

   Where $P_{CT}[i]$ is the closing time of the restaurant and $P_U[i]$ is the time required by the user to reach the restaurant. Now if $T_P \leq 0$ then the restaurant is discarded form the dataset.
5. All the restaurants that have distance more than the user defined distance context are discarded. After contextual pre-filtering restaurant dataset is reduced to 8 different places.

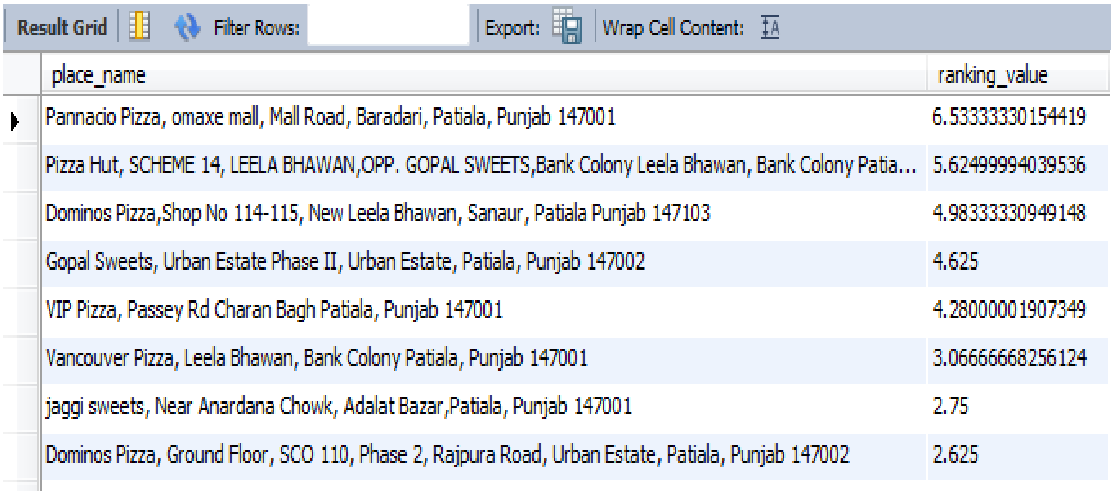
5.4.4 Ranking

Contextualized restaurant dataset is used in the process of ranking. In the location based application recommendations are provided based on the user location. As an example Thapar university Patiala is considered as the active user current location.

In the ranking function there are total 4 parameters. The values of all 4 parameters are calculated to get the overall ranking of that place of the user. The ranking function R (u, p) used in the system to rank nearby places is defined as follows:

$$R\ (u, p) = RDist\ (u, p) + Rule\ priority + R_{User\text{-}pref}\ (u, p) + Pub\_Eval\ (p)$$

The values of all the 4 parameters are calculated based on the information in the restaurant domain dataset. Individual values of 4 parameters are added to obtain the overall rating of the place for the user. Once the rankings of all places are obtained then top-k places based on ranking function are presented on the Google map. Value of k in the Top-k is 10 chosen by the user. After the filtering process only 8 places left in the dataset.



place_name	ranking_value
Pannacio Pizza, omaxe mall, Mall Road, Baradari, Patiala, Punjab 147001	6.53333330154419
Pizza Hut, SCHEME 14, LEELA BHAWAN,OPP. GOPAL SWEETS,Bank Colony Leela Bhawan, Bank Colony Patia...	5.62499994039536
Dominos Pizza,Shop No 114-115, New Leela Bhawan, Sanaur, Patiala Punjab 147103	4.98333330949148
Gopal Sweets, Urban Estate Phase II, Urban Estate, Patiala, Punjab 147002	4.625
VIP Pizza, Passey Rd Charan Bagh Patiala, Punjab 147001	4.28000001907349
Vancouver Pizza, Leela Bhawan, Bank Colony Patiala, Punjab 147001	3.06666668256124
jaggi sweets, Near Anardana Chowk, Adalat Bazar,Patiala, Punjab 147001	2.75
Dominos Pizza, Ground Floor, SCO 110, Phase 2, Rajpura Road, Urban Estate, Patiala, Punjab 147002	2.625

Figure 5.6: Ranking list top-8 restaurants serving pizza

5.4.5 Presenting Top-K places on the Google map

Top-K places obtained after the ranking process are presented on the Google map. To present information on the Google map firstly the addresses of various places need to be converted into geographical coordinates. To convert restaurant addresses into geographical coordinates Google Geocoding API is required. JavaScript in Geocoding API is required to dynamically convert addresses into geographical coordinates such as Latitude and longitude. Top-K places based on user location corresponding to user defined context in the form of rules are shown in figure 5.6. In the use scenario after the ranking process Top-8 places are left to recommend.

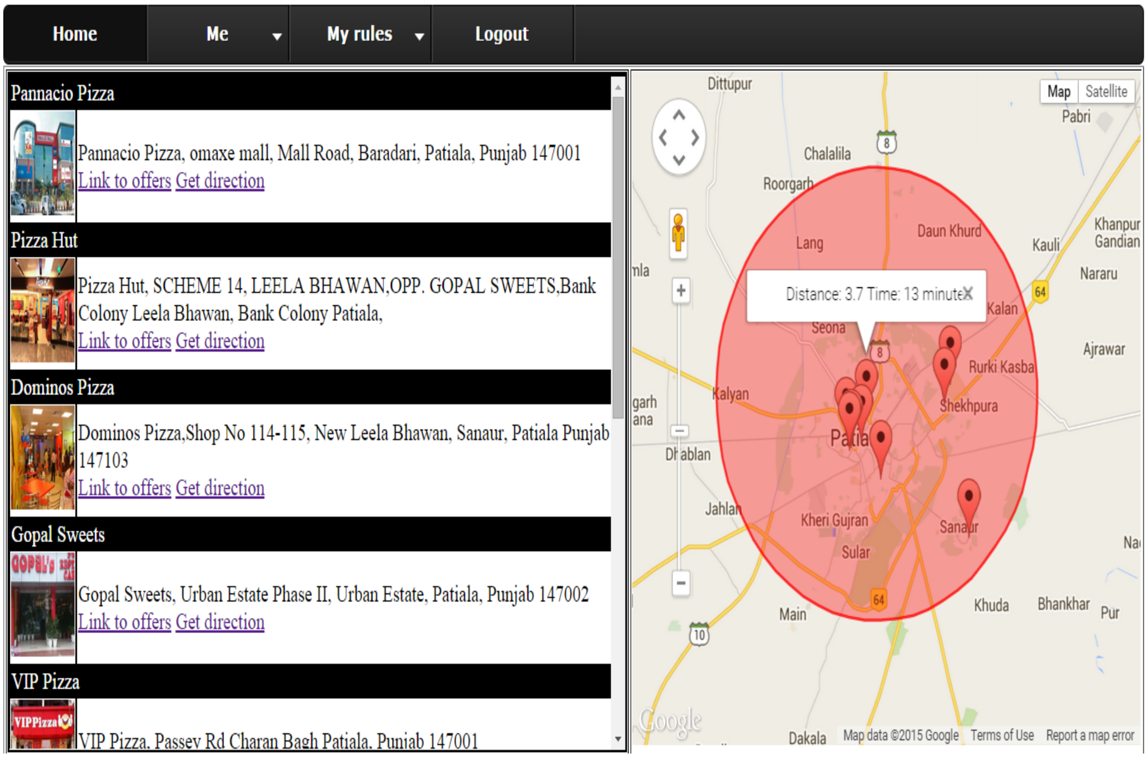


Figure 5.7: Top-K places on the Google map

Radius of distance that the user wants to travel is shown on the Google map. When the user mosehover on the restaurant information at the time a pop up is shown on the marker to showing distance required to reach the restaurant along with the time required to reach the restaurant.

CHAPTER 6
CONCLUSION AND FUTURE SCOPE

In thesis we have presented an approach named as location based context aware recommender system through user defined rules by integrating user location and user preference in the form of user defined rules.

In this chapter, a brief overview of thesis work and how this work can be extended further is described.

6.1Conclusion

A recommendation model is proposed to recommend nearby places to the user based on user defined RuleML rules and active user current location. User defined rules is the main source of context awareness in the recommender system. Contextual pre-filtering paradigm is used to filter the domain dataset followed by ranking of nearby places based on our ranking function. The ranking function includes four things: user distance preference, rule priority, public evaluation and user tag based preference to a place. The ranking function ranks nearby places and top-k recommendations are presented to the user on the Google map.

The user can mouse hover on various recommended places to view distance and approximate time required to reach that place. The can also click on the link to view various offers going on that place and can he can click on link to get direction to that place.

6.2 Future Scope

In the future work, this proposed system can evolve in various ways. The following are the ways in which the system can be evolve:

1. Integrate this model with traditional recommendation approaches such as content and collaborative based filtering.
2. We can also calculate similarity among different user's rules base to recommend rules to the users.

3. Addition of semantics to the contextual information will make the system more accurate and efficient.
4. Social networking framework could be integrated with the system, where preferences of the user, user profiles and rules can be interchanged with social networking site profiles (e.g. from Google+, Facebook etc.). Further, people that are closer in the social networking graph can share and re-use rules among other people.

REFERENCES

[1]. B. McFee, L. Barrington and G. Lanckriet, "Learning content similarity for music recommendation," *Audio, Speech, and Language Processing, IEEE Transactions,* vol. 20, no. 8, pp. 2207-2218, 2012.

[2]. N. N. Liu, L. He and M. Zhao, "Social temporal collaborative ranking for context aware movie recommendation", *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 4, no. 1, pp. 15, 2013.

[3]. W. Carrer-Neto, M. L. Hernández-Alcaraz, R. Valencia-García and F. García-Sánchez, "Social knowledge-based recommender system. Application to the movies domain", *Expert Systems with Applications,* vol. 39, no. 12, pp. 10990-11000, 2012.

[4]. N. Korfiatis and M. Poulos, "Using online consumer reviews as a source for demographic recommendations: A case study using online travel reviews," *Expert Systems with Applications,* vol. 40, no. 14, pp. 5507-5515, 2013.

[5]. F. Ullah, G. Sarwar, S. C. Lee, Y. K. Park, K. D. Moon and J. T. Kim, "Hybrid recommender system with temporal information", In *Information Networking (ICOIN), International Conference on*, pp. 421-425, 2012.

[6]. K. Verbert, N. Manouselis, X. Ochoa, M. Wolpers, H. Drachsler, I. Bosnic and E. Duval, "Context-aware recommender systems for learning: a survey and future challenges", *Learning Technologies, IEEE Transactions on*, vol. 5, no. 4, pp. 318-335, 2012.

[7]. G.M. Machado and J. P. M. de Oliveira, "Context-aware adaptive recommendation of resources for mobile users in a university campus," *Wireless and Mobile Computing, Networking and Communications (WiMob), IEEE 10th International Conference on*, pp. 427-433, 2014.

[8]. Z. D. Champiri, S. R. Shahamiri and S. S. B. Salim, "A systematic review of scholar context-aware recommender systems." *Expert Systems with Applications,* vol. 42, no. 3, pp. 1743-1758, 2015.

[9]. K. Verbert, N. Manouselis, X. Ochoa, M. Wolpers, H. Drachsler, I. Bosnic and E. Duval, "Context-aware recommender systems for learning: a survey and future

challenges", *Learning Technologies, IEEE Transactions,* vol. 5, no. 4, pp. 318-335, 2012.

[10]. D. Gavalas, C. Konstantopoulos, K. Mastakas and G. Pantziou, "Mobile recommender systems in tourism", *Journal of Network and Computer Applications"*, vol. 39, pp. 319-333, 2014.

[11]. M. N. Gasson, E. Kosta, D. Royer, M. Meints and K. Warwick, "Normality mining: Privacy implications of behavioral profiles drawn from GPS enabled mobile phones", *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on,* vol.41, no. 2, pp. 251-261, 2011.

[12]. I. Viktoratos, A. Tsadiras and N. Bassiliades, "A context-aware web-mapping system for group-targeted offers using semantic technologies", *Expert Systems with Applications,* vol. 42, no. 9, pp. 4443-4459, 2015.

[13]. A. Veltri, M. Pagell, D. Johnston, E. Tompa, L. Robson, B. C. Amick III, S. Hogg-Johnson and S. Macdonald, "Understanding safety in the context of business operations: An exploratory study using case studies", *Safety science*, vol. 55, pp. 119-134, 2013.

[14]. A. Said, S. Berkovsky, E. W. De Luca and J. Hermanns, "Challenge on context-aware movie recommendation: CAMRa2011", *In Proceedings of the fifth ACM conference on Recommender systems*, pp. 385-386, 2011.

[15]. D. Bouneffouf, A. Bouzeghoub and A. L. Gançarski, "Following the User's Interests in Mobile Context-Aware Recommender Systems: The Hybrid-e-greedy Algorithm", *In Advanced Information Networking and Applications Workshops (WAINA), 26th International Conference on*, pp. 657-662, 2012.

[16]. D. Nadoveza and D. Kiritsis, "Ontology-based approach for context modeling in enterprise applications", *Computers in Industry,* vol. 65, no. 9, pp. 1218-1231, 2014.

[17]. N. Zheng and Q. Li, "A recommender system based on tag and time information for social tagging systems", *Expert Systems with Applications,* vol. 38, no. 4, pp. 4575-4587, 2011.

[18]. Z. Xu, L. Chen and G. Chen, "Topic based context-aware travel recommendation method exploiting geotagged photos", *Neurocomputing*, vol. 155, pp. 99-107, 2015.

[19]. M. Kahng, S. Lee and S.G. Lee, "Ranking in context-aware recommender systems", *In Proceedings of the 20th international conference companion on World wide web*, pp. 65-66, 2011.

[20]. Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, A. Hanjalic and N. Oliver, "TFMAP: Optimizing MAP for top-n context-aware recommendation", *In Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pp. 155-164, 2012.

[21]. D. Gallego, W. Woerndl and G. Huecas, "Evaluating the impact of proactivity in the user experience of a context-aware restaurant recommender for Android smartphones", *Journal of Systems Architecture,* vol. 59, no. 9, pp. 748-758, 2013.

[22]. B. Schilit, N. Adams, and R. Want, "Context-aware computing applications", *In proceeding of the IEEE conference on Mobile Computing Systems and Applications, WMCSA*, pp. 85-90, 1994.

[23]. G. Adomavicius and A. Tuzhilin, "Context-aware recommender systems", *In Proceedings of the 2008 ACM conference on recommender systems, ACM, New York, NY, USA,* pp. 335–336, 2008.

[24]. W. P. Lee, C. Kaoli and J. Y. Huang, "A smart TV system with body-gesture control, tag-based rating and context-aware recommendation", *Knowledge-Based Systems*, vol. 56, pp. 167-178, 2014.

[25]. T. H. Dao, S. R. Jeong and H. Ahn, "A novel recommendation model of location-based advertising: Context-Aware Collaborative Filtering using GA approach", *Expert Systems with Applications*, vol. 39, no. 3, pp. 3731-3739, 2012.

[26]. J. S. F. Tan, E. H. C. Lu and V. S. Tseng, "Preference-oriented mining techniques for location-based store search," *Knowledge and information systems,* vol. 34, no. 1, pp. 147-169, 2013.

[27]. L. O. Colombo-Mendoza, R. Valencia-García, A. Rodríguez-González, G. Alor-Hernández and J. J. Samper-Zapater, "RecomMetz: A context-aware knowledge-based mobile recommender system for movie showtimes", *Expert Systems with Applications*, vol. 42, no. 3, pp. 1202-1222, 2015.

[28]. I. Viktoratos, A. Tsadiras and N. Bassiliades, "Providing a context-aware location based web service through semantics and user-defined rules," *Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics (WIMS14),* pp. 9, June 2014.

[29]. A. Ciaramella, M. G. C. A. Cimino, B. Lazzerini and F. Marcelloni, "Situation-aware mobile service recommendation with fuzzy logic and semantic web", *In Intelligent Systems Design and Applications, ISDA'09, IEEE, Ninth International Conference on*, pp. 1037-1042, 2009.

[30]. C. Kebler, M. Raubal and C. Wosniok, "Semantic rules for context-aware geographical information retrieval", *In Smart Sensing and Context*, pp. 77-92, 2009.

[31]. A. Garcia-Crespo, J. L. Lopez-Cuadrado, R. Colomo-Palacios, I. Gonzalez-Carrasco and Belen Ruiz-Mezcua, "Sem-Fit: A semantic based expert system to provide recommendations in the tourism domain", *Expert systems with applications*, vol. 38, no. 10, pp.13310-13319, 2011.

[32]. D. Martín-Serrano, R. Hervás and José Bravo, "Telemaco: context-aware system for tourism guiding based on web 3.0 technology", *In 1st Workshop on Contextual Computing and Ambient Intelligence in Tourism, Riviera Maya, Mexico*, 2011.

[33]. Furno, Angelo, and Eugenio Zimeo, "Context-aware composition of semantic web services", *Mobile Networks and Applications* , vol. 19, no. 2, pp. 235-248, 2014.

[34]. W. V. Woensel, S. Casteleyn and O. D. Troyer, "Applying semantic web technology in a mobile setting: the person matcher", pp. 506-509, 2010.

[35]. W. Viana, A. D. Miron, B. Moisuc, J. Gensel, M. Villanova-Oliver and .Martin, "Towards the semantic and context-aware management of mobile multimedia", *Multimedia Tools and Applications*, vol. 53, no. 2, pp. 391-429, 2011.

[36]. N. S. Savage, M. Baranski, N. E.Chavez, and T. Hollerer, "*I'm feeling loco: A location based context aware recommendation system*", 2012.

[37]. L. Baltrunas, B. Ludwig, S. Peer and F. Ricci, "Context relevance assessment and exploitation in mobile recommender systems", *Personal and Ubiquitous Computing*, vol. 16, no. 5, pp. 507-526, 2012.

[38]. B. Fang, S. Liao, K. Xu, H. Cheng, C. Zhu and H. Chen, "A novel mobile recommender system for indoor shopping", *Expert Systems with Applications* vol. 39, no. 15, pp. 11992-12000, 2012.

[39]. W. S. Yang, H. C. Cheng and J. B. Dia, "A location-aware recommender system for mobile shopping environments", *Expert Systems with Applications*, vol. 34, no. 1, pp. 437-445, 2008.

[40]. M. Batet, A. Moreno, D. Sánchez, D. Isern and A. Valls, “Turist@: Agent-based personalised recommendation of tourist activities”, *Expert Systems with Applications*, vol. 39, no. 8, pp. 7319-7329, 2012.

[41]. F. Gedikli and D. Jannach, "Improving recommendation accuracy based on item-specific tag preferences," *Transactions on Intelligent Systems and Technology (TIST),* vol. 4, no. 1, pp. 11, 2013.

[42]. S. Sen, J. Vig and J. Riedl, “Tagommenders: connecting users to items through tags”, *In Proceedings of the 18th ACM international conference on World wide web*, pp. 671-680, 2009.

PUBLICATION

Published:

[1]. Silky Sharma and Dr. Damandeep Kaur, “Location Based Context aware recommender system through user defined rules”, International Conference on Computing, Communication and Automation (IEEE ICCCA), at Galgotias, Greater Noida, September, 2015.

VIDEO PRESENTATION

This is link to my YouTube video where I have presented brief summary of thesis: https://www.youtube.com/watch?v=btpeE_ATMR8