Journal of Information Assurance and Security.
ISSN 1554-1010 Volume 8 (2013) pp. 220-229

SRPRS: Situation-Aware Reputation Based Proactive Recommender System

Punam Bedi[1] and Sumit Kumar Agarwal[2]

[1, 2] Department of Computer Science, University of Delhi,

Delhi, India

punambedi@ieee.org, sumitsagarwal@rediffmail.com

***Abstract*: The proactive recommender systems are widely used intelligent applications which automatically deliver (i.e. push) recommendations to the users, without explicit request from them. Such systems are very effective in the application domains where the availability of items changes often and rapidly, as they help users timely receive their interested information in the form of push content. However, if these systems push uninterested information to the users, or even push interested information to the users but at inappropriate context, then the users' acceptance of pushed recommendations by these systems will decrease enormously. Hence for improving users' acceptance in proactive recommender systems, determining right push context (situation assessment) and finding relevant items for the target users are very crucial. This paper presents a Situation-Aware Reputation Based Proactive Recommender System (SRPRS) that pushes relevant items to the target user at the right context only. The recommendation process in the proposed system is divided into two phases: (i) situation assessment phase and the (ii) item assessment phase. In situation assessment phase, the SRPRS system analyzes the current situation i.e. whether or not the current context needs a recommendation to be pushed. In item assessment phase, the suitable items are selected as recommendations using proposed location-aware reputation based collaborative filtering (LRCF) algorithm. SRPRS uses fuzzy logic as an inference technique to handle uncertainty in situation assessment phase. The prototype of SRPRS has been designed and developed for restaurant recommendations. Performance of the implemented prototype system is evaluated using precision, recall metrics and users' subjective feedback.**

***Keywords*:** Reputation, Situation-Awareness, Fuzzy Logic, Multi-Agent System, Pro-activity, Recommender Systems.

1. Introduction

We are dependent upon the opinion of our acquaintances, reviews in the newspapers, magazines, and general surveys etc, for simple decisions of our daily life, like which place to visit, which movie to watch, which book to read, which restaurant to eat in. These opinions and reviews help us to find interesting products or services. This support from the society provides a shortcut to select a good alternative as otherwise the cost and effort required is usually not deemed to be worth the trouble. Recommender systems are intelligent applications that provide assistance to users by giving personalized product recommendations based on user preferences to handle this information overload problem [5, 6, 30]. Conventional recommender systems usually pursue a request-response pattern i.e. such systems give item suggestions on an explicitly request by the user. In mobile recommender systems, users can not browse through many search results and suffer from other restrictions in the user experience, due to the limitations of mobile devices such as small display size or missing keyboard [8, 26].

The proactive recommender systems aim to reduce interaction in order to achieve better user experience in mobile environment by pushing recommendations to a user at the right context, without explicit user request [2]. As the user does not request for items in proactive recommender systems, improving user's acceptance on proactively delivered recommendations is a big challenging task in these systems. The determination of right push context (situation assessment) and finding relevant items for the target user are two main crucial issues for improving user's acceptance in proactive recommender systems. The proposed SRPRS system is a multi-agent system that handles these issues efficiently in two phases; first the situation assessment i.e. determining whether or not the current context needs a recommendation to be pushed and the second phase generates recommendation for the target user when a first phase indicates a promising situation using proposed location-aware reputation based collaborative filtering algorithm (LRCF). This algorithm augments the concept of reputation with the recommendation process to deal a set of geographically located recommendable items. Every user in this system is represented by an agent. SRPRS uses fuzzy logic as an inference technique to handle uncertainty in situation assessment phase. The fuzzy logic is derived from fuzzy set theory to deal with reasoning that is approximate rather than precise [16, 29]. The SRPRS system also requires a minimal level of interaction with the end user, as it provides suggestions to the users based on their profile. The main contribution of this research work is to achieve two aspects (relevance of information and unobtrusiveness [22]) of proactive recommender systems.

The rest of the paper is organized as follows: Section 2 reviews the related work in this area. Our proposed approach SRPRS with a detailed description of each step is presented in section 3. Experimental details and evaluations are shown in section 4 and finally section 5 concludes the paper.

2. Related Work

A lot of research has been done in literature on context-awareness, mobile computing, personalization, recommender systems and location based services. In the survey paper on mobile recommender systems, Ricci [24] discussed that various systems make use of current user's behavior along with contextual information to improve personalization in mobile devices. Kenteris et al. [18] recently surveyed on electronic mobile guides and acknowledged that now a day's pro-activity has gained much attention in personalization and recommender system research. The approaches discussed in [1, 14] push all items that are located near the user's position, without concerning his preferences. The systems developed in [9, 20] push items based on user's preferences but these systems do not estimate the right context to push items.

Ciaramella et al. [10] presented a rule-based mobile service recommender system that uses a fuzzy logic to determine a user's situation, but this system pushes all services associated with the determined situation to the user without concerning his preferences. Nguyen and Hoang [23] proposed mobile push-delivery recommendation methodology that provides proactively relevant recommendations to the user at appropriate context. Yeung and Yang [27] presented AHP model that uses Bayesian network to predict user's interest that accounts for delivering proactive personalized recommendations. Tair et al. [25] presented architecture for context-aware proactive recommender system based on reduction-based theory. The approaches discussed in [13, 15, 17, 19] generate personalized point of interest recommendations in mobile environment using context-aware collaborative filtering approach.

Hong et al. [12] proposed an agent-based model for proactive personalized services based on context history. This model enables proactive recommendations based on the user profile which is deduced from the context history. But the training time is extremely insignificant in their proposed model. Melguizo et al. [22] in their work have discussed the three main requirements of proactive recommender system: (a) Relevance of information: the right information to the right user at the right time, (b) Long Term Memory: what the user has done and using it, (c) Unobtrusiveness: avoid disturbing and irritating the user. Bader et al. [3] have focused only on first aspect (relevance of information) of proactive recommender system in their work. Bader et al. [4] presented an argumentation based explanation approach to enhance transparency of proactively delivered recommendations in order to achieve better user acceptance.

Although the considerable amount of work has been done on proactive personalized recommender systems, the solution of improving user's acceptance in proactive recommender systems is still a challenging task. We propose a novel approach SRPRS which improves user's acceptance by pushing relevant items to him using proposed location-aware reputation based collaborative filtering algorithm at the appropriate context only. SRPRS also handles uncertainty that is implicit within situation assessment phase using fuzzy logic.

3. Proposed Situation-Aware Reputation Based Proactive Recommender System

In this section, we present situation-aware reputation based proactive recommender system (SRPRS) that automates the process of pushing the relevant restaurants to the mobile user according to his preferences and contextual information using multi-agent approach. Section 3.1 presents the architecture of SRPRS system. The recommendation generation process of SRPRS is discussed in section 3.2. An illustration is shown in section 3.3

3.1. Architecture of the SRPRS

SRPRS is a multi agent system where every user in this system is represented by an agent. The agents within SRPRS communicate with each other to generate the recommendation. The basic building blocks of this system are shown in figure 1. On the client side, a normal off-the-shelf Internet browser is the only component that the user sees on the mobile screen during usual operation. Using this component only, the mobile user can interact with the system. Whenever a mobile user registers with the system using this component, a corresponding user agent (UA) is created at the server. The basic functionality of UA is to infer and keep information about mobile user. The UA coordinates with other components of the server such as context information collector, situation assessment module and recommendation engine for pushing item suggestions to a target user. The context information collector component of the server periodically detects the information about the mobile user such as his current location and time in order to obtain a good estimate of contextual attributes (distance, time etc). The situation assessment module determines the right push context (situation) for suitable item suggestion. The recommendation engine maintains the access database and generates the recommendations for the UA using location-aware reputation based collaborative filtering algorithm (LRCF). Figure 1 shows the architecture diagram of SRPRS system.

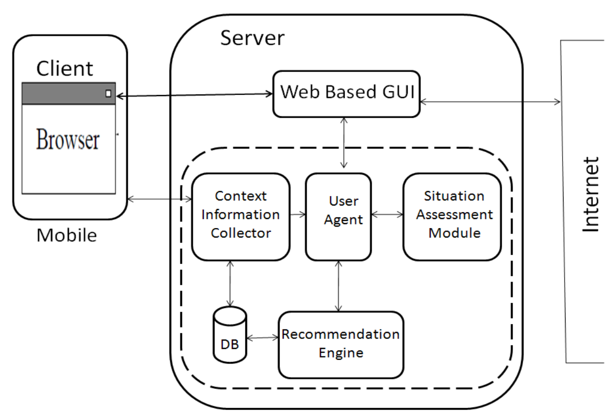

Figure 1. Architecture of the SRPRS System

Additionally SRPRS contains two more system agents: similarity agent (SA) and items reputation agent (IRA). These system agents execute in the background at the recommendation engine component of SRPRS, to periodically compute the similarity between the users and reputation of all items.

3.2. Recommendation Generation

The recommendation generation process of SRPRS is divided into two phases: (i) situation assessment phase and the (ii) item assessment phase. The situation assessment phase is executed periodically in the background and the item assessment phase is only executed when the first phase indicates a promising situation.

3.2.1. Phase 1: Situation Assessment

In this phase, the system needs to determine whether or not the user must receive a recommendation. To do so, the system calculates a context level using fuzzy logic. The fuzzy logic is basically a multi-valued logic that allows intermediate values to be defined between conventional evaluations like yes/no, true/false, black/white etc [29]. The context level is a number between 0 and 1. The context level is estimated based on three contextual attributes such as distance, time and budget. These three contextual attributes are used as linguistic input variables within the system. All the values for these input variables are mapped by the system as fuzzy number by using suitable fuzzy sets. Levels of these input variables as fuzzy sets are defined as below.

Distance = {Near, Moderate, Far}

Time = {Pre-time, In-time, Post-time}

Budget = {Inexpensive, Affordable, Expensive}

The levels of output variable i.e. context level as fuzzy set within system is defined as below.

Context Level = {Low, Medium, High}

The standard triangular membership function is used by the system to represent the regions for each input and output variables. The representation of input parameter distance is shown in figure 2. Similarly the other inputs and output parameters (time, budget and context level) are also defined within the system.

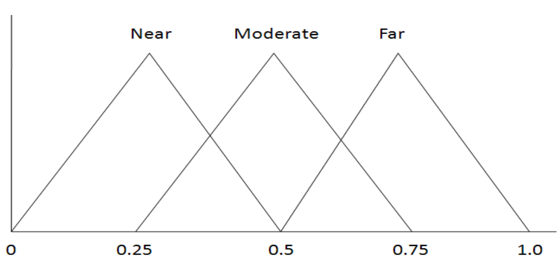

Figure 2. Triangular Membership function for input parameter Distance

The relationship between the output variable and input variables are defined by rules within system in the following format:

(1) IF (Distance is ‘Near’) AND (Time is ‘In-time’) AND (Budget is ‘Affordable’) THEN Context Level is ‘High’
(2) IF (Distance is ‘Far’) AND (Time is ‘Post-time’) AND (Budget is ‘Expensive’) THEN Context Level is ‘Low’
(3) IF (Distance is ‘Moderate’) AND (Time is ‘In-time’) AND (Budget is ‘Affordable’) THEN Context Level is ‘High’

The other rules are formulated in similar way. The most widely used following defuzzification method is used to calculate the crisp value of output parameter context level within the system.

$$\text{Centroid of area} = \int_Z \mu_A(z) z\,dz \Big/ \int_Z \mu_A(z)\,dz \quad (1)$$

Where $\mu_A(z)$ is the aggregate output MF [29].

If this estimated crisp value of context level exceeds a threshold, then second phase will be initiated. Otherwise this phase is executed again after some time period.

3.2.2. Phase 2: Item Assessment

The second phase evaluates the relevant items to be recommended for the target user using LRCF algorithm. This algorithm deals with a set of geographically located recommendable items, A= $\{I_1, I_2…., I_n\}$, based on the parameter R_{outer} that establishes the limits where the items might be recommended. This parameter, R_{outer} is used to compute a subset, A’ $\subseteq$ A, that includes those items that are suitable to be recommended to the user according to his location and ignoring the remaining ones because they are far away from user’s location as shown in figure 3.

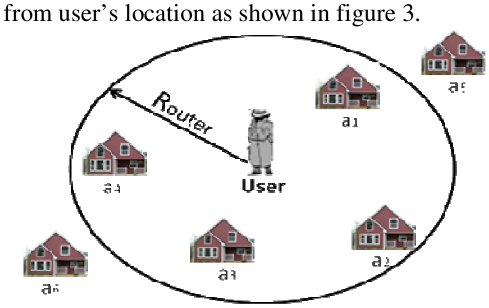

Figure 3. Region for recommendations: items a_5 and a_6 are discarded because they are outside

The LRCF algorithm works in two processes: offline process (preprocessing phase) and online process. In offline process, the reputation of each item and similarity between the users are periodically computed to form items reputation vector (IRV) and users’ similarity matrix (USM) by the system agents IRA and SA respectively. The computed IRV and USM are stored in the database for future recommendation generation. In online process, the recommendations are generated and pushed to the target user.

LRCF algorithm is briefly outlined as follows:

Offline Process (preprocessing phase)
Step 1. Input data normalization
Step 2. Create items reputation vector (IRV), users similarity matrix (USM) and store the information in the database
Online Process (Recommendation process for the target user)
Step 1. Select the similar users and aggregate their recommendation (items) lists.
Step 2. If an aggregated list obtained from step 1 contains at least top n items then go to step 5 else go to step 3.
Step 3. Filter items from IRV based on the parameter R_{outer} and aggregate them.
Step 4. Aggregate both lists obtained from steps 1 and 3.
Step 5. Push top n items of aggregated list to the target user
Step 6. Update USM and IRV using feedback mechanism

The following subsections explain LRCF algorithm in detail.

Offline Process (preprocessing phase):

Step 1) Input data normalization:

The system stores the input data in the form of user-item rating matrix. In this matrix rows represent the users, columns represent the items and the value in i^{th} row and j^{th} column represents the rating of i^{th} user for j^{th} item. This input matrix consists of ratings in the discrete scale 1 to 5. The system normalizes these ratings in the range 0 to 1 as follows:

$$r_{ij} = \frac{r_{ij}}{\sum_{j=1}^{n} r_{ij}} \quad (2)$$

where

r_{ij} represents the rating of i^{th} user for j^{th} item in the user-item rating matrix

n denotes the total number of items in the user-item rating matrix

Step 2) Create items reputation vector (IRV), users' similarity matrix (USM) and store the information in the database:

(i) The system agent IRA computes reputation of each item in the background periodically to form an IRV from the user-item rating matrix. The reputation of an item depends on the following three factors:

- Average rating of users for an item
- Number of users who rated that item
- How close the rating of users to each other for that item.

The reputation of an i^{th} item (ROI_i) in IRV is computed as follows:

$$ROI_i = \frac{3 \times avg_i \times (n_i/N) \times (1/SD_i)}{avg_i \times (n_i/N) + (n_i/N) \times (1/SD_i) + avg_i \times (1/SD_i)} \quad (3)$$

where

avg_i represents the average rating of users for an i^{th} item

n_i represents number of users who rated i^{th} item in the user-item rating matrix

N denotes total number of users in user-item rating matrix

SD_i denotes standard deviation of the ratings given by individual users for the i^{th} item. A small SD_i indicates that the ratings are around the mean i.e. ratings of the users are close to each other.

(ii) The similarity between users is computed by the system agent SA to form USM using user-item rating matrix. There are number of possible similarity computing measures, for example the Euclidean distance metric, cosine/vector similarity metric and Pearson correlation coefficient metric. The Pearson correlation coefficient metric is used by SA for computing similarity sim(x, y) between the two users x and y as follows:

$$Sim(x,y) = \begin{cases} P_{x,y} = \frac{\sum (r_{xi} - \bar{r}_x)(r_{yi} - \bar{r}_y)}{\sigma_x \sigma_y} & if\ P_{x,y} > 0 \\ 0 & otherwise \end{cases} \quad (4)$$

where

r_{xi} and r_{yi} denote the ratings of users x and y for i^{th} item respectively.

$\bar{r}_x$ and $\bar{r}_y$ denote the average ratings of user x and y respectively.

σ_x and σ_y denote the standard deviation of the ratings of users x and y respectively.

Pearson correlation coefficient $P_{x,y}$ lies in between -1 to 1. $P_{x,y} <= 0$ indicates that x and y are not correlated, therefore negative and zero correlation are not considered and all such values are set to zero in USM.

Once the normalized user-item rating matrix, IRV and USM are created, the database contains the following information: normalized ratings of all users for the items, reputation of each item and similarity of each user on others for the online process.

Online Process (recommendation process for the target user):

The online process starts, when the system agent UA determines the right push context by coordinating with situation assessment module. In this process UA finds relevant items for the target user. The online process is described in detail as follows:

Step 1) Select the similar users and aggregate their recommendation (items) lists:

The UA selects those users as similar users from USM, whose similarity value exceeds some threshold. Then UA retrieves the rated items (item list) of selected similar users from user-item rating matrix. The UA generates an aggregated list from all retrieved item lists of similar users using the following aggregation method:

(i) Identify the distinct items from the retrieved item lists of similar users

(ii) Compute the degree of importance (DoI) for each distinct item I_i as follows:

$$DoI_i = ROI_i \times DD(u, I_i) \quad (5)$$

where

ROI_i is the reputation an item I_i obtained from IRV

$DD(u, I_i)$ is the distance decay function for the item I_i and the target user u. The objective of this function is to minimize the DoI of an item, when the distance of user's current location from that item increases. This decay function is defined on two parameters R_{outer} and R_{inner} as depicted in figure 4. R_{outer} parameter is introduced at the starting of section 3.2.2. The R_{inner} parameter defines the inner radius of circle centered at target user u that considers distance of an item from the user's current location is low and thus DoI of an item that lies in the range of Rinner, is completely dependent on ROI.

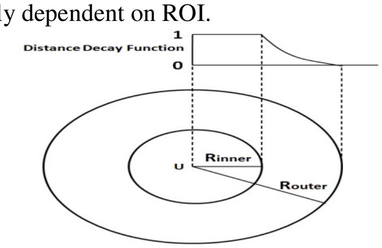

Figure 4. Distance decay function

According to the proposal of Yang et al. [28], the distance decay function is defined as follows:

$$DD(u, I_i) = \begin{cases} 1 & \text{if } dist(u, I_i) \le R_{inner} \\ e^{-(dist(u,\ I_i) - R_{inner})} & \text{if } R_{inner} < dist(u, I_i) \le R_{outer} \\ 0 & \text{if } dist(u, I_i) > R_{outer} \end{cases} \quad (6)$$

where

$dist(u, I_i)$ is the Euclidean distance from the target user u to the item I_i

(iii) Arrange the nonzero DoI items in descending order of their DoI

Step 2) If an aggregated list obtained from step 1 contains at least top n items then go to step 5 else go to step 3:

The UA counts the number of items in the aggregated list obtained from step 1. If this count is greater or equal to the value of top n then UA executes directly step 5 because he has sufficient number of items to push to a target user, otherwise UA executes step 3 for finding additional relevant items for the target user.

Step 3) Filter items from IRV based on parameter R_{outer} and aggregate them:

The UA filters and aggregates the items from IRV using the following steps:

(i) The UA filters those items from IRV, whose reputation value is greater than some threshold and their distances from the target user's current location is less than the value of R_{outer}.

(ii) The UA computes the DoI for each filtered item using equation (5).

(iii) Arrange the items in descending order of their DoI.

Step 4) Aggregate both lists obtained from steps 1 and 3:

The item lists obtained from step 1 (list1) and step 3 (list2) are aggregated as follows:

(i) Place list1 at the top of aggregated list.

(ii) Filters out all items of list1 from list2 then remaining items of list2 are appended within the aggregated list.

Step 5) Push top n items of aggregated list to the target user:

The UA selects top n items from the aggregated list and pushes them on the target user mobile.

Step 6) Update USM and IRV using feedback mechanism:

The USM and IRV are updated based on the target user's feedback. In feedback mechanism, the system displays previous ratings to the target user for the pushed items, if exists and then asks him to rate the pushed items. UA updates the user-item rating matrix based on new ratings of the target user and then accordingly the system agents IRA and SA updates IRV and USM respectively.

3.3. An Illustration

To ease the discussion starts with offline process, toy example shown in table 1 is used, where U_1-U_{10} are users and I_1-I_{10} are items (restaurants) rated/unrated by the user. Table 2 shows the computed IRV for the items I_1 to I_{10} using equation (3). The calculated USM for the users U_1 to U_{10} using equation (4) is depicted in table 3.

To better understand the online process, Let UA represents a target user U_9, threshold be 0.70 for selecting similar users of UA. The users U_3 and U_5 qualify the threshold criteria as shown from USM (table 3). UA retrieved the items (I_5 and I_8) of U_3 and (I_3, I_4, I_5, I_6 and I_{10}) of U_5 from user-item rating matrix (table 1). For simplicity let among all distinct retrieved items (I_3, I_4, I_5, I_6, I_8 and I_{10}) by UA, the items I_3 and I_5 are within R_{inner} range and rest of the items (I_4, I_6, I_8 and I_{10}) are outside the R_{outer} range. The computed DoI of these items using equation (5) is shown in table 4. The items I_5 and I_3 are stored in order as the outcome of step1 of online process in list1. Let the value of top n is 3, therefore in this example UA executes step 3 of online process. Let item's reputation threshold be 0.30 for filtering the items from IRV. The items I_1, I_3, I_4, I_5, I_6, I_7, I_8 and I_{10} are initially filtered from IRV (table 2). Among these items only I_1, I_3, I_5 and I_7 qualify the R_{outer} criteria. The computed DoI of these items using equation (5) is shown in table 5. The items I_5, I_3, I_1 and I_7 are stored in order, according to their DoI in descending order as the outcome of step 3 of online process in list2. The items I_5, I_3, I_1 and I_7 are stored in order as an aggregated list after combining list1 and list2. Finally top n items (I_5, I_3 and I_1) are selected to be pushed to the target user.

Table 1: Toy example of user-item rating matrix after normalization in the range 0 to 1

Users	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	I_{10}
U_1	0.1765	0.0588	0.1176	0.2941	0.0588	0.0000	0.0000	0.2353	0.0588	0.0000
U_2	0.2308	0.0769	0.1538	0.0769	0.3846	0.0000	0.0000	0.0000	0.0769	0.0000
U_3	0.0000	0.0000	0.0000	0.0000	0.8333	0.0000	0.0000	0.1667	0.0000	0.0000
U_4	0.0000	0.0588	0.0000	0.1176	0.2941	0.0588	0.1765	0.2941	0.0000	0.0000
U_5	0.0000	0.0000	0.2308	0.1538	0.3846	0.0769	0.0000	0.0000	0.0000	0.1538
U_6	0.0833	0.1667	0.2500	0.4167	0.0833	0.0000	0.0000	0.0000	0.0000	0.0000
U_7	0.0909	0.0000	0.0000	0.0909	0.0000	0.1818	0.4545	0.0909	0.0000	0.0909
U_8	0.2500	0.1250	0.1875	0.3125	0.0000	0.0000	0.0000	0.1250	0.0000	0.0000
U_9	0.0000	0.0000	0.0000	0.0000	0.5556	0.0000	0.0000	0.0000	0.1111	0.3333
U_{10}	0.1818	0.0909	0.1818	0.0909	0.0000	0.4545	0.0000	0.0000	0.0000	0.0000

0 indicates not rated

Table 2: The Items Reputation Vector (IRV)

	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	I_{10}
Reputation Value	0.3921	0.2479	0.4249	0.4601	0.6856	0.3827	0.3611	0.3970	0.1935	0.3478

Table 3: The Users' Similarity Matrix (USM)

Users	U_1	U_2	U_3	U_4	U_5	U_6	U_7	U_8	U_9	U_{10}
U_1	1.0000	0.1114	0.0000	0.2228	0.0000	0.6188	0.0000	0.8550	0.0000	0.0000
U_2	0.1114	1.0000	0.7403	0.2000	0.6419	0.2172	0.0000	0.1503	0.5617	0.0000
U_3	0.0000	0.7403	1.0000	0.6963	0.7113	0.0000	0.0000	0.0000	0.7985	0.0000
U_4	0.2228	0.2000	0.6963	1.0000	0.2573	0.0000	0.1930	0.0000	0.3113	0.0000
U_5	0.0000	0.6419	0.7113	0.2573	1.0000	0.3249	0.0000	0.0000	0.7284	0.0000
U_6	0.6188	0.2172	0.0000	0.0000	0.3249	1.0000	0.0000	0.7916	0.0000	0.0743
U_7	0.0000	0.0000	0.0000	0.1930	0.0000	0.0000	1.0000	0.0000	0.0000	0.0411
U_8	0.8550	0.1503	0.0000	0.0000	0.0000	0.7916	0.0000	1.0000	0.0000	0.1248
U_9	0.0000	0.5617	0.7985	0.3113	0.7284	0.0000	0.0000	0.0000	1.0000	0.0000
U_{10}	0.0000	0.0000	0.0000	0.0000	0.0000	0.0743	0.0411	0.1248	0.0000	1.0000

Table 4: The Computed DoI of all distinct items received from similar users

	I_3	I_4	I_5	I_6	I_8	I_{10}
DoI	0.4249	0.0000	0.6856	0.0000	0.0000	0.0000

Table 5: The Computed DoI of filtered items from IRV

	I_1	I_3	I_5	I_7
DoI	0.3921	0.4249	0.6856	0.3611

4. Experimental Details

The techniques and tools that are used in implementation of the prototype SRPRS are listed below:

- Java Server Pages (JSP) for creating user interface
- Java Agent Development Environment (JADE) for creating multi-agent environment
- MySQL 5.0.24 for backend database
- MATLAB 2010 for developing a fuzzy inference system that is required for situation assessment module

To prepare a dataset for the experiment, we collected the information of restaurants of Delhi such as restaurants name, their opening and closing times, food type, average food cost per person, and their addresses from http://www.zomato.com/ncr/restaurants website and stored into local database. Then we retrieved and stored the longitude and latitude of these restaurants into local database to build a restaurant dataset by using reverse geo-coding tool available at http://www.distancesfrom.com/latitude-longitude.aspx. The prepared restaurant dataset contains the information of 2166 restaurants of Delhi, India.

4.1 Evaluation Metrics

The performance of SRPRS was evaluated in two parts: (i) the standard metrics, precision and recall for evaluating the goodness of recommendations pushed by SRPRS and (ii) an online user's subjective feedback for determining the overall acceptance and usefulness of the system. Precision is a measure of accuracy or fidelity and recall or sensitivity is a measure of completeness. Precision score of 1.0 signifies that all recommendations retrieved were relevant. Recall score of 1.0 signifies that all relevant recommendations were retrieved [11]. One of the ways to evaluate precision and recall is to predict the top N items for recommendation. To evaluate the goodness of recommendations in SRPRS, the top N recommendations were pushed to all registered users and first 80 users' feedbacks were considered for evaluation. Goodness of these pushed recommendations depends on users' satisfaction measure which was depicted by the users in the form of acceptance/ non-acceptance feedback. The precision and recall of top N recommendations can be defined as follows:

$$Precision = \frac{1}{|U|}\sum_{u \in U} \frac{|acc\_items(u,cloc) \cap top(u,cloc)|}{|top(u,cloc)|} \quad (7)$$

$$Recall = \frac{1}{|U|}\sum_{u \in U} \frac{|acc\_items(u,cloc) \cap top(u,cloc)|}{total\_items(u,cloc)} \quad (8)$$

where

cloc denotes the current locality of user u (locality where the user u is currently situated in)

top(u, cloc) denotes the set of top N pushed items of cloc to user u

acc_items(u, cloc) $\subseteq$ top(u, cloc), denotes the set of accepted items of cloc by user u

total_items(u, cloc) denotes the set of total items of cloc, where user u is currently in.

Both precision and recall measures are clearly conflicting in nature. If the number of top N recommendations pushed increases, then the absolute number of relevant items (i.e. recall) increases while at the same time precision is decreased. But since both precision and recall are important in estimating the performance of a system that generates top N recommendations, they can be combined together with equal weights to get a single metric, the F_1 metric. The F_1 metric consists of weighted combination of precision and recall. The general formula for F_1 measure is as follows:

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (9)$$

Higher values of F_1 indicate a more balanced combination between recall and precision. An online users' subjective feedback was taken to determine the overall acceptance and usefulness of SRPRS. To evaluate these properties, the nine semantic differential adjective pairs [21] were used. A total 32 participants took part in this feedback evaluation. The participants' were asked to rate the system's performance against these nine parameters on a five-point scale, ranging from -2 to +2, where -2 means "strongly disagree" and +2 is "strongly agree"

4.2 Experimental Results

The system was tested and analyzed on restaurant dataset by varying the values of top N viz. 3, 5, 10 and the number of users' involved in recommendations viz. 10, 50 and 80. The following observations were made from the results obtained:

(a) Influence of number of users involved in recommendations: As shown in table 6, with the increase in number of users from 10 to 80, precision and recall both increases. The identical effect is depicted in figure 5.

Table 6: Sample output obtained from restaurant dataset when Top N = 5 with varying number of users (Here Precision, Recall and F_1 metrics are in percentage)

	Number of Users		
	10	**50**	**80**
Precision	66.34	69.08	70.12
Recall	14.79	16.13	16.9
F_1	24.19	26.15	27.24

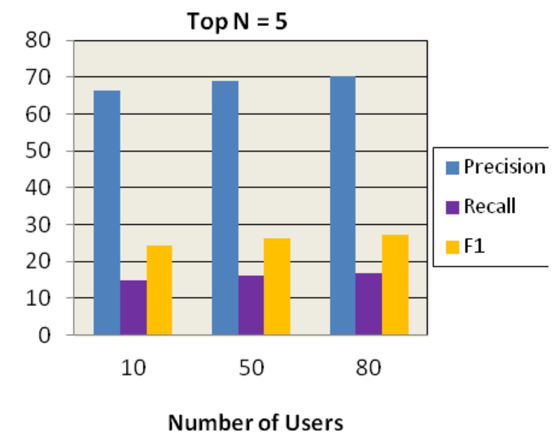

Figure 5. Partial sample output illustrating influence of number of users involved in recommendations

(b) As top N increases, recall increases and precision decreases. It can be seen in table 7 and figure 6 that recall increased as the value of top N increased from 3 to 10. On the other hand, precision dropped smoothly as the top N number of recommendations increased. This is likely as the average quality of the recommendations made decreases as the number of recommendation to be made increases. F_1 measure indicates that the best combination of precision and recall is achieved for high values of top N.

Table 7: Sample output obtained from restaurant dataset when number of users = 80 and Top N value is set at 3, 5, 10. (Here Precision, Recall and F_1 are in percentage)

	Top N = 3	Top N = 5	Top N = 10
Precision	84.02	70.12	48.28
Recall	12.08	16.9	23.16
F_1	21.13	27.24	31.31

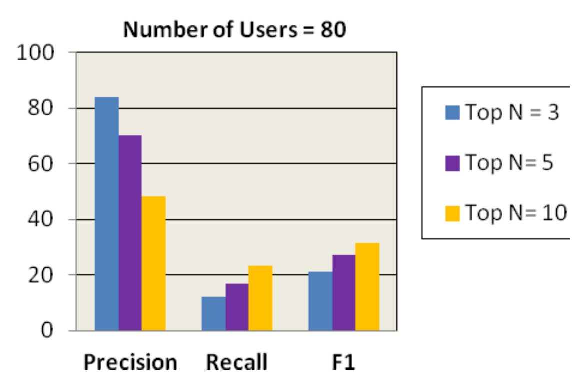

Figure 6. Effect of Top N on Precision and Recall

(c) Comparison of LRCF with conventional collaborative filtering (CF) approach: Conventional CF approach was implemented using Pearson Correlation Coefficient metric and compared with our proposed LRCF algorithm. The comparative results obtained between conventional CF and LRCF are shown in the table 8 and figure 7.

Table 8: Comparative results obtained between conventional CF and LRCF on restaurant dataset when number of users = 80 and Top N = 5. (Here Precision, Recall and F_1 are in percentage)

	CF	LRCF
Precision	58.93	70.12
Recall	27.87	16.9
F_1	37.84	27.24

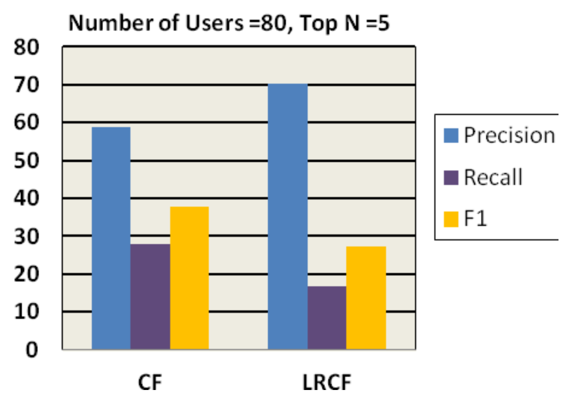

Figure 7. Comparative graph between CF and LRCF Approach

Results from table 8 illustrate that while generating recommendations; precision is increased by approximately by 11% using LRCF approach as compared to conventional CF approach.

The result of an online users' subjective feedback to determine the overall acceptance and usefulness of SRPRS is shown in figure 8. The average of users' feedback ratings are taken for each differential adjective pairs in this evaluation. As can be seen from the result, the semantic differential questions revealed a positive mind-set towards SRPRS, while some users articulated mild concerns about SRPRS being annoying or irritating.

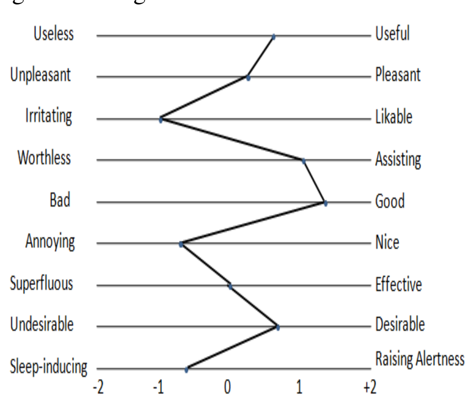

Figure 8. Semantic differential results of SRPRS

4.3 *Discussion*

The main contribution of our proposed approach SRPRS is that it pushes relevant items to the target user at the right context. A list of pushed restaurants to the target user is shown in figure 9. The user may click on any pushed recommendation to get the details of that restaurant as shown in figure 10.

Figure 9. A list of pushed restaurants to the target user

Figure 10. Restaurant details showed after selecting a restaurant

The proposed algorithm LRCF gives better results than conventional CF approach and it also handles the main weakness of conventional CF i.e. new user (cold start) problem. This problem occurs in conventional CF when recommendations are to be generated for the user, who has not rated any item or rated very few items. In this case, it is extremely difficult for the system to find similar users of a target user for recommending the items. This problem is solved in LRCF through the filtering of items from IRV (step 3 of online phase). Since if none of the similar users are found then LRCF can recommend the items based on their reputation value.

Our approach SRPRS also improves the unobtrusiveness concern of proactive recommender systems by pushing the items to the users at the appropriate context only. Our approach SRPRS also handles uncertainty that is implicit in situation assessment.

5. Conclusion and future work

A Situation-Aware Reputation Based Proactive Recommender System (SRPRS) has been designed and developed for suggesting restaurants to the mobile users. The main emphasis in the presented work is the determination of right push context and generation of relevant recommendations for a target user in order to achieve better user's acceptance. The situation assessment has been done to determine right context for pushing the items in the system. The uncertainty while situation assessment is also handled using fuzzy logic. The relevant recommendations are pushed to the target user using proposed LRCF algorithm. The proposed system SRPRS is implemented using multi-agent approach and its performance is evaluated and compared with conventional CF based approach. It is found that our approach generates better results as compared to conventional CF based approach.

As a future work, we will be working towards the user interface of proactive recommender systems for achieving minimum user intervention in order to improve user's acceptance.

References

[1] L. Aalto, N. Gothlin, J. Korhonen, T. Ojala, "Bluetooth and WAP push-based location-aware mobile advertising system", In Proc. 2nd International. Conf. Mobile Systems, Applications and Services, 2004, pp 49-58.

[2] G. Adomavicius, A. Tuzhilin, "Context-aware recommender systems", Recommender Systems Handbook, Springer, US, 2011, pp 217-253.

[3] R. Bader, W. Woerndl, V. Prinz, "Situation Awareness for Proactive In-Car Recommendations of Points-Of-Interest (POI)", In Workshop on Context Aware Intelligent Assistance. Karlsruhe, 2010

[4] R. Bader, A. Karitnig, W. Woerndl, G. Leitner, "Explanations in Proactive Recommender Systems in Automotive Scenarios", In Workshop on Decision Making and Recommendation Acceptance Issues in Recommender Systems, 2011.

[5] P. Bedi, S. Agarwal, "Aspect-oriented Trust Based Mobile Recommender System", International Journal of Computer Information Systems and Industrial Management Applications, Vol (5) 2013, pp 354-364.

[6] P. Bedi, A. Sinha, S. Agarwal, A. Awasthi, G. Prasad and D. Saini, "Influence of Terrain on Modern Tactical Combat: Trust-based Recommender System", Defense Science Journal, India, Vol. 60, No. 4, pp. 405-411, (2010).

[7] P. Bedi, S. Agarwal, "Situation Aware Proactive Recommender System", 12th International Conference on Hybrid Intelligent Systems, pp 85-89, Pune, India, December 4-7, 2012, IEEE Xplore.

[8] P. Bedi, S. Agarwal, "Aspect-Oriented Mobility-Aware Recommender System", International Conference on World Congress on Information and Communication Technologies, pp 191-196, WCIT 2011, December 11-14, 2011 at Mumbai, India, USA: IEEE Xplore.

[9] J. E. Castro, H. Shimakawa, "Mobile advertisement system utilizing user's contextual information", In Proc. 7th Int.ernational Conf. Mobile Data Management, 2006, pp 91-98.

[10] A. Ciaramella, M. G. C. A. Cimino, B. Lazzerini, F. Marcelloni, "Situation-aware mobile service recommendation with fuzzy logic and semantic web", In Proc. 9th Int. Conf. Intelligent Systems design and Applications, 2009, pp 1037-1042.

[11] J. Herloker, J. A. Konstan, L. G. Terveen, J. T. Riedl, "Evaluating collaborative filtering recommender systems", ACM Transactions on Information Systems, 2004,22(1), pp 5-53.

[12] J. Hong, E. H. Suh, J. Kim, S. Kim, "Context-aware system for proactive personalized service based on context history", Expert System and Applications, Elsevier, 2009, 36: 7448-7457.

[13] T. Horozov, N. Narasimhan, V. Vasudevan, "Using location for personalized POI recommendations in mobile environments", In Proc. of the Int. Symposium on Applications on Internet, IEEE Computer Society, Washington, USA, 2006, pp 124-129.

[14] N. Hristova, G. O'Hare, "Ad-me wireless advertising adapted to the user location, device and emotions", In Proc. 37th Annual Hawai International Conf. System Sciences, 2004, pp 285-294.

[15] H. Huang, G. Gartner, "Using context-aware collaborative filtering for POI recommendations in mobile guides" In Lecture notes in Geoinformation and Catrography, Springer, Berlin Heidelberg, 2012, pp 131-147.

[16] S.R. Jang, T. Sun and E. Mizutani, "Neuro-Fuzzy and Soft Computing Pearson Education", 2007.

[17] D. Jannach, Zanker, J. M. Noguera, O. Seider, "Developing a conversational travel advisor with advisor suite", Information and Communication Technologies in Tourism, 2007, pp 43-52.

[18] M. Kenteris, D. Gavalas, D. Economou, "Electronic mobile guides: a survey. Personal and Ubiquitous Computin", 15(1): 97-111.

[19] M. H. Kuo, L. C. Chen, C. W. Liang, "Building and evaluating a location-based service recommendation system with a preference adjustment mechanism, Expert Systems with Applications, 2009, 36(1), pp 3543-3554.

[20] S. Kurkovsky, K. Harihar, "Using ubiquitous computing in interactive mobile marketing", Journal of Personal and Ubiquitous Computing, 2006, 10(4): 227-240.

[21] V.D. Laan. A. Heino and D. Waard, "A simple procedure for the assessment of acceptance of advanced transport telematics", Transpotation Research Part C: Emerging Technologies. Vol 5, No.1, 1997, 1-10, Elsevier Science Ltd.

[22] P. Melguizo, M. C. Bogers, L. Boves, A. Deshpande, A. V. D. Bosch, J. Cardoso, J. Cordeiro, J. Filipe, "What a Proactive Recommendation System Needs: Relevance, Non-Intrusiveness and a New Long term Memory", In Proc. 9th International Conf. on Enterprise Information Systems, 2007, pp 86-91.

[23] Q. N. Nguyen, P. M. Hoang, "Push delivery personalized recommendations for mobile users", In Proc. 4th International Conf. on Mobile Ubiquitous Computing Systems, Services and Technologies, 2010, pp 412-415.

[24] F. Ricci, "Mobile Recommender Systems", International journal of information technology and tourism, 2010, 12(3): 1-24.

[25] H. A. Tair, M. J. Zemerly, M. A. Qutayri, M. Leida, " Architecture for Context-Aware Pro-Active Recommender System", International Journal Multimedia and Image Processing, 2012, 2(1): 125-133.

[26] W. Woerndl, J. Huebner, R. Bader, D. G. Vico, "A Model for Proactivity in Mobile, Context-aware Recommender Systems", In Proc. 5th ACM International Conference on Recommender Systems, 2011, pp 273-276.

[27] K. F. Yeung, Y. Yang, "A proactive personalized mobile news recommendation system", In Proc. International Conf. on Developments in E-systems Engineering, 2010, pp 207-211.

[28] W. S. Yang, H. C. Cheng, J. B. Dia, "A location-aware recommender system for mobile shopping environments", Expert Systems with Applications, 2008, 34(1), pp 437-445

[29] L.A Zadeh, "Fuzzy Sets. Information and Control", 1965, vol(8), pp 338-353.

[30] J. Zhang, N. Jones, P. Pu, "A visual interface for critiquing-based recommender systems", *EACM C*, 2008, pp 230–239.

Author Biographies

Dr. Punam Bedi received her Ph.D. in Computer Science from the Department of Computer Science, University of Delhi, India in 1999 and her M.Tech. in Computer Science from IIT Delhi, India in 1986. She is an Associate Professor in the Department of Computer Science, University of Delhi. She has about 25 years of teaching and research experience and has published more than 140 research papers in National/International Journals/Conferences. Dr. Bedi is a member of AAAI, ACM, senior member of IEEE, and life member of Computer Society of India.

Her research interests include Web Intelligence, Soft Computing, Semantic Web, Multi-agent Systems, Intelligent Information Systems, Intelligent Software Engineering, Software Security, Intelligent User Interfaces, Requirement Engineering, Human Computer Interaction (HCI), Trust, Information Retrieval, Personalization, Steganography and Steganalysis.

Mr. Sumit Kr. Agarwal received his M.Tech degree in Information Technology from USIT, Guru Gobind Singh Indraprasth University Delhi, India in 2007 and his M.Sc in Information Science from Dr. Bhim Rao Ambedkar University, Agra in 2002. He is a doctorate student in Department of Computer Science, University of Delhi. Mr. Agarwal is an associate member of IETE, Loadhi Road, New Delhi. His research interests include Web Intelligence, Multi-Agent Systems, Trust, Recommender Systems and Aspect-oriented Programming.