CONTEXT AWARE VENUE SUGGESTIONS SYSTEM FOR CLOUD BASED ARCHITECTURE

Miss Pratiksha P.Nagare [1], Prof. Nitin Shivale[2]

[1] *Student, Department of Computer Engineering, JSPM's BSIOTR, Maharashtra, India*
[2] *Assistant Professor, Department of Computer Engineering, JSPM's BSIOTR, Maharashtra, India*

ABSTRACT

Nowadays, recommendation systems have been significant evolution in the field of knowledge engineering. There are many existing recommendation systems that are based on collaborative filtering approaches that make them easy to implement. Performance of the existing collaborative filtering-based recommendation system suffers due to cold start, data sparseness, and scalability. Recommendation problem is characterized by the presence of many conflicting objectives or decision variables, such as user preferences and venue closeness. This paper includes Context Aware Venue Suggestions System for cloud based architecture in mobile depending on their moods. The MobiContext utilizes multi-objective optimization techniques to generate personalized recommendations. This system mainly depends upon the status updated by user. Status helps to find location and with the help of location we can trace places. In this system user can update his status, according to that status system can find the location of that user and system can also trace the places nearby that location.

Keyword: - *Collaborative Filtering, Recommendation System, Mobi-Context, Recommendation Framework.*

---

1. INTRODUCTION

Context aware venue suggestion system mainly focuses on the status updated by user, this status helps to find locations and with the help of these locations we can trace places. In this system user can updates his status; according to that status system can find the location of that user. System can also trace the places nearby to that location. This system is used to save Cost &Time. Information systems (IS) are area under discussion to enthusiastically changing state of affairs in the IS delivery phase.

The existing system can be described in detail by different data which can be defined as any information required that characterize the circumstances of an entity where an entity can be a person, place, or object which is considered to be significant to the communication among the user and the application together with the user and the applications themselves. The context data can be taken into account in IS delivery thereby increasing the usability and user satisfaction.

Composite and widespread IS may reduce the user pleasure and, if possible, users may choose substitution ways of carrying out their tasks. Like for instance, nowadays public are avoiding use of public e-services and are preferring physical services. Recommender systems are extensively used in order to recover the practice of software and tools which provide suggestions by recommending the items which the users might likely be interested for. The recommender systems are gradually becoming more popular. A variety of such systems pay attention on civilizing and evaluating the collaborative-filtering technique. They use secret information about historical user activity, user profile information and other information to match users with recommended items. Progressively, the recommender systems initiate to use more diverse data and data sources, for instance, social network data. In a number of cases the perspective information may be applicable to estimate the most suitable recommendations for users by making use of context-aware recommendation systems. Latest investigations are made on location-based and weather-dependent recommendation algorithms and methods. Discovering proper context data is yet a confront in recommender systems evaluation. The projected approach involves the use of different context data that can be retrieved from both internal and external data sources. The context dealing out includes not only reading the context data, but also context data analysis that helps to forecast the context data and user behavior.

The recommender systems are habitually item oriented and put forward the items in which users may be mostly interested in. Like for an example, e-commerce sites such as Amazon.com suggest items that the users would buy while content based systems recommend stuff based on textual analysis, e.g. in the area of research, citations can be suggested for the research by analyzing words in research papers. With the intention of improve the recommender algorithms, hybrid recommender systems are urbanized by combining various recommendation algorithms and methods in one IS. The approach considers that recommendations could be any kind of software entity and instances of recommendations involve suggestions to carry out a function, procedure, and workflow or to perform data processing operations.

Different type of context information be capable of be using an input to the recommendation module. The modeling phase is well thought-out as significant as the models can help to deal with complexity and are easy perceptible for stakeholders without any specific IT skills. Recommendation modeling includes specifying a set of business rules that help to describe the software entities context dependencies and decides which recommendations should be run in each background situation.

At the moment variability in IS delivery becomes progressively more important. When business processes changes, software sustaining the processes must be accustomed in view of fulfilling the goals of the organization. Spirited software ought to be capable to deal with the changeability including minimal efforts. Variation in business processes can be exaggerated by internal and/or external context. Software adjustment to change in form of user recommendations by combining various recommendation modules in existing software which permit to deal with inconsistency without an important effort as the recommendations can be easily altered in recommendation module exclusive of changing the underlying software.

The potential of using context-aware recommendations, and proposal for an approach for modeling context aware recommender systems had been described by various authors. The modeling approach is based on the Capability Driven Development (CDD) method used in development of adaptive systems. The potential of using context-aware recommender systems is analyzed by exploring a use case from the e-government domain.

2. RELATED WORK

Rizwana Irfan, Osman Khalid, Muhammad Usman Shahid Khan [1] has discussed a cloud-based framework MobiContext in“MobiContext: A Context-aware Cloud-Based Venue Recommendation Framework” that produces optimal recommendations by considering the trade-offs among real-world physical factors, such as person’s location and closeness of that location. The significance and novelty of the proposed framework is adaptation of collaborative filtering and bi-objective optimization approaches, such as scalar and vector. In the proposed approach, data sparseness is addressed by collecting together the user-to-user similarity computation with confidence measure that appraise amount of similar interest indicated by the two users in the venues commonly visited by both of them. Moreover, a solution to cold start issue is given by introducing the HA inference model that assigns ranking to the users and has a precompiled set of popular unvisited venues that can be recommended to the new user.

Qi He, Jian Pei, Daniel Kifer, Prasenjit Mitra, C. Lee Giles [2] describes the novel problem of context-aware citation recommendation, and built a context-aware citation recommendation prototype in CiteSeerX in “Context-aware Citation Recommendation”. The system is capable of recommending the bibliography to a manuscript and providing a ranked set of citations to a specific citation placeholder. It is developed a mathematically sound context-aware relevance model. A non-parametric probabilistic model as well as its scalable closed form solutions is then introduced.

Michael Roberts, Nicolas Ducheneaut, Bo Begole, Kurt Partridge [3] demonstrated both that a client-server based mobile recommendation system in “Scalable Architecture for Context-Aware Activity-Detecting Mobile Recommendation Systems” practically for large scale deployment and one can obtain a good density of users per server node. With the current system, it does the majority of the computation on the back-end server infrastructure. With increasing capability of mobile devices, it should be possible to perform more of this work on the client in particular, systems which allow clear migration of data and computation between small devices and more extensive facilities which provides larger (server) devices which are interested in directions to pursue. However, particularly in the case of algorithms of collaborative filtering, it even makes sense to get extensive server build-out.

3. PROPOSED SYSTEM AND FRAMEWORK

3.1 Architecture

The information gathering process is mainly depends upon the status updated by user in the context. Status is help to find location which will help to trace the places. In this system user can updates his status. According to the status systemcan find the location of that user and systemcan also trace the places nearby that location according to the user's mood. As shown in figure the context can be the mood of user that can be entered by user or it can be acquired from the social sites. The structure of our system is as shown in figure.

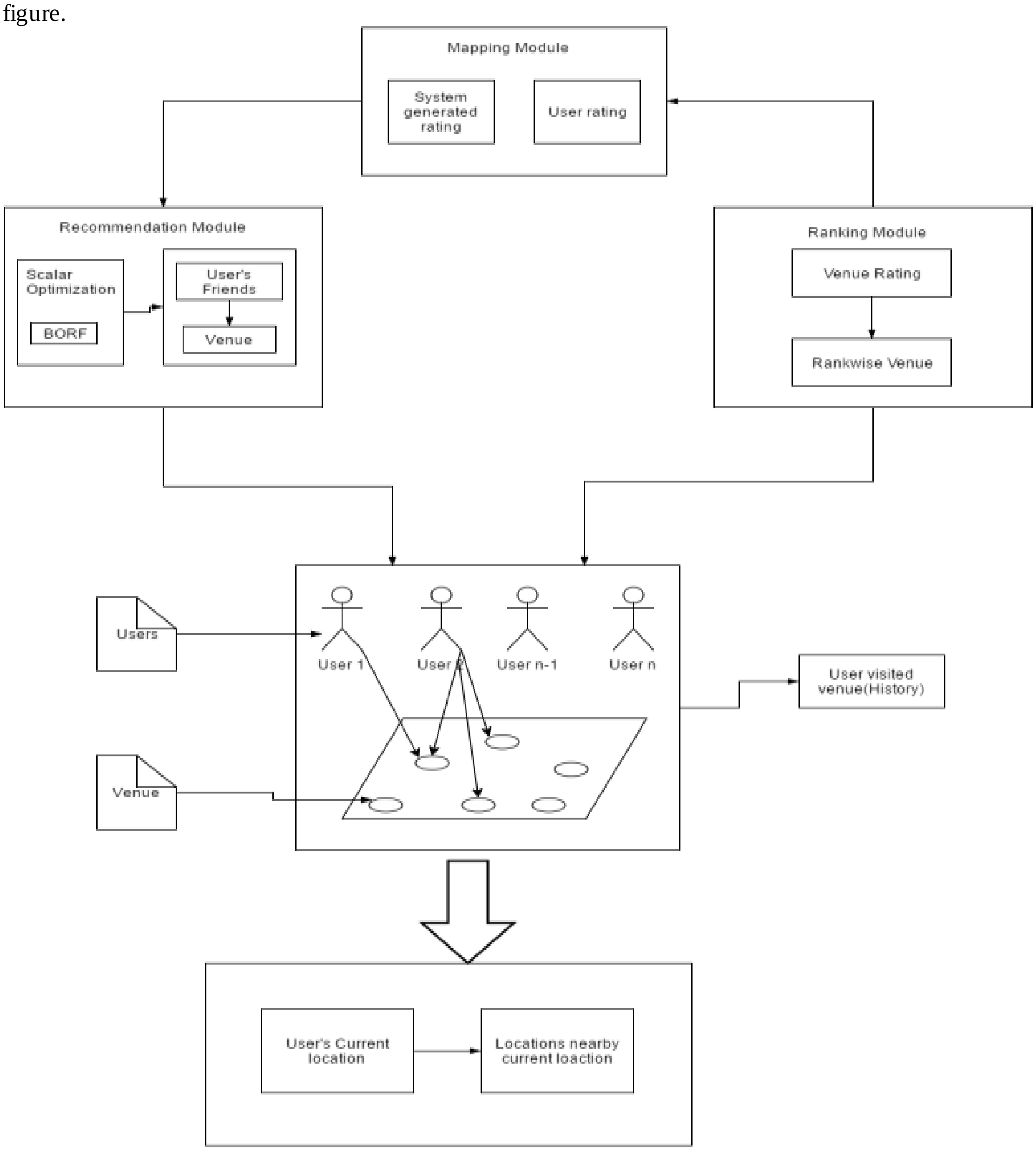

Fig -2:Architecture of the system

3.2 Mathematical Model

Similarity is calculated using users and users check-in at a venue. It is calculated using Pearson Correlation Co-efficient where value of PCC ranges between -1 and +1. -1 indicates preferences of two users which do not match and 1 indicates higher degree of similarity between 2 users.

$$\frac{\sum \text{Venues of user's } c \,\&\, c' \text{check-ins} \left(\begin{matrix} \text{No. of check}-\text{ins} \\ \text{by user } c \text{ at venue } v(r_{cv}) \end{matrix} - \begin{matrix} \text{Average no. of check}-\text{ins} \\ \text{of user } c(\bar{r}_c) \end{matrix} \right) * \left(\begin{matrix} \text{No. of check}-\text{ins} \\ \text{by user } c' \text{ at venue } v(r_{c'v}) \end{matrix} - \begin{matrix} \text{Average no. of check}-\text{ins} \\ \text{of user } c'(\bar{r}_{c'}) \end{matrix} \right)}{\sqrt{\sum \text{Venues of user's } c \,\&\, c' \text{check}-\text{ins} \, (r_{cv} - \bar{r}_c)^2 \sum \text{Venues of user's } c \,\&\, c' \text{check}-\text{ins} \, (r_{c'v} - \bar{r}_{c'})^2}}$$

In above formula, similarity is calculated between two experts c and c' for only those venues that are visited by both users. The similarity calculation for few results into very sparse similarity graph because many of the venues are not visited by either of the two users. This tends to data sparseness problem, which can compute similarity with confidence measure. The confidence measure can be interpreted by conditional probability that one venue visited by a one user is also visited by other user in dataset. The following formula is used to calculate the weight of an edge between two users.

$$\omega_{c\,c'} = \begin{cases} s_r(c, c') & \text{if } s_r(c, c') > 0 \\ & \text{otherwise} \\ P(r_c | r_{c'}) \times \dfrac{1}{1 + \sum_{v \in r_{c'}} |r_{cv} - r_{c'v}|} & \end{cases}$$

Here the parameters r_c and $r_{c'}$ are set of venues checked-in by the user c and c'. $P(r_c | r_{c'}) = P[r_c \cap r_{c'}] / P[r_{c'}]$ is the ratio that both the users can visit the similar set of venues in future. The sum factor in denominator is used to keep value of probability lower than similarity so that preference should be given to positive values of similarity. Thus it implies that an edge is always assigned with a non-zero weight that helps reducing data sparseness due to zero similarity.

Now fitness function is used to compute the ranking score of each recommended venue with an individual where the venues ranked were first computed using the HA inference. The fitness function f1 of an individual in a population can be computed as follows:

$$f1 = \frac{\sum_{i=1}^{n} (\text{ranked venues})}{\text{Total number of venues in a single individual}}$$

The second fitness function f2 computes geospatial distance between current user's location and venue of each corresponding venue of an individual is as follows:

$$f2 = \frac{1}{\sum_{i=1}^{n} \text{cost}(\text{current location of user}, \text{consecutive venues } v)_i \times \text{total number of venues})}$$

3.3 Algorithm

Algorithm 1: General Algorithm

1. Sort locations
2. Sort context's types
3. Retrieve status of person's context.

4. If(context_type==hungry)
Show types of hotel nearby person's location

If(context_type== sad)

Show entertainment locations like places to hangout, multiplex nearby.

If()….

5. Show results.

Algorithm 2: Rating Algorithm

The algorithm is based on the Bayesian Average. This is a mathematical term, a calculation of an object's ranking based on the "credibility" of the ratings it receives. The calculation applies the number of votes and the ratings of the individual submissions, as well as the ratings of all other submissions and the overall number of votes cast. This ensures that the aggregate is always taken into account, which also makes it possible to represent quality.
Bayes' Formula: br = ((avg_num_votes * avg_rating) + (this_num_votes * this_rating)) / (avg_num_votes + this_num_votes)
Legend:
avg_num_votes: The average number of votes of all objects that received at least 1 vote.
avg_rating: The average rating of each object that received more than 1 vote.
this_num_votes: The number of votes for this object.
this_rating: The rating for this object.

$$\text{Rating Value} = \frac{(\text{Ø no. of votes for all submissions} * \text{Ø rating for each submission}) + (\text{no. of votes for this submission} * \text{rating of this submission})}{(\text{Ø no. of votes for all submissions} + \text{no. of votes for this submission})}$$

Algorithm 3: Ranking Algorithm
1. Sort the rating in descending order
2. Retrieve those rating
3. Display rating>4

4. RESULTS

Figure 2:Result 1

Figure 2 shows history of the user visited venues along with the rate of the venues.Rate is generated by the algorithm.

Figure 3:Result 2

Figure 3 shows the locations according to the context are generated. The locations displayed in red are the nearby locations of the user's current location.

Figure 4: Result 3

Figure 3 describes the details of the venue. If a person clicks on"GO", the place will be shown as places visited and the rating provided by the user will be stored.

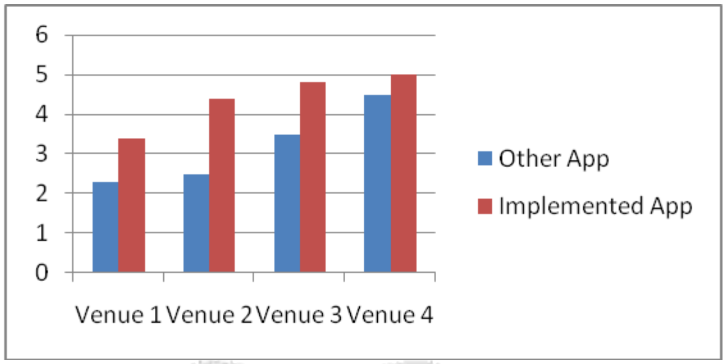

Chart 1: Comparison Graph

Figure 5 shows the graph of the system as compared to other applications. The X-axis shows the accurate venues nearby the user's current location and Y-axis shows the rating of the particular venue.

5. CONCLUSION

In this system it is easy to find location immediately where status is updated and also find the places list nearby that location according to ranking or most visited. Here, proposed a cloud-based framework MobiContext has produces optimized recommendations by simultaneously considering the trade-offs among real-world physical factors, such as person's location and closeness of the location. The significance and novelty of the proposed framework is the adaptation of collaborative filtering and bi-objective optimization approaches, such as scalar and vector. In our proposed approach, data sparseness issue is addressed by integrating the user-to-user similarity computation with confidence measure that quantifies the amount of similar interest indicated by the two users in the venues commonly visited by both of them. Moreover, a solution to cold start issue is discussed by introducing the HA inference model that assigns ranking to the users which also have a precompiled set of popular unvisited venues that can be recommended to the new user.

6. FUTURE WORK

In the nearby age, people face with mass of statistics and information. The information can be consequent from diverse sources and it may be added every day. The phenomenon is identified as information overload. Recommender systems emerged with the scope to help users to discover items that fit their interests and preferences. As a result, the lake of an appropriate survey for showing the applications of recommender systems, recommendation system express out some essential applications and advantages and disadvantages of the last presented systems.

7. ACKNOWLEDGEMENT

I like to acknowledge my vigorous thanks to Prof. Nitin Shivale for giving suggestions which helped me a lot in my research work and I also want to thanks our friends and classmates for helping me in this research work by giving me there timely suggestions and feedbacks on my research work.

8. REFERENCES

[1] **"MobiContext: A Context-aware Cloud-Based Venue Recommendation Framework"**, Rizwana Irfan, Osman Khalid, Muhammad Usman Shahid Khan, 2015.

[2] **"Context-aware Citation Recommendation"**, Qi He, Jian Pei, Daniel Kifer, Prasenjit Mitra, C. Lee Giles. , 2010.

[3] **"Scalable Architecture for Context-Aware Activity-Detecting Mobile Recommendation Systems"**, Michael Roberts, Nicolas Ducheneaut, Bo Begole, Kurt Partridge., 2008.

[4] A. Majid, L. Chen, G. Chen, H. Turab, I. Hussain, and J. Woodward, **"A Context aware Personalized Travel Recommendation System based on Geo-tagged Social Media Data Mining,"** International Journal of Geographical Information Science, pp. 662-684, 2013

[5] M. Ye, P. Yin, and W. Lee, **" Location recommendation for location-based social networks,"** In Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, ACM, pp. 458-461, 2010

[6] C. Chow, J. Bao, and M. Mokbel, **" Towards Location-Based Social Networking Services,"** In Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks, ACM, pp. 31-38, 2010.