www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 5 Issue 09 September 2016 Page No.17973-17978

Flight Recommendation System based on user feedback, weighting technique and context aware recommendation system

Mohit Tuteja
Computer Science and Engineering
Maharaja Agrasen Institute of Technology
New Delhi, India

Abstract:

In this paper, a recommender system for recommending flights to customers on the basis of user preferences, weightage technique and context aware system is proposed to help the consumers of service oriented environment to discover and select the most appropriate flight services from a large number of available ones. This recommender system provides the user with desired selection options, real-time information and recommends the user of itineraries that best fit his preferences, based on his previous purchases. These preferences are learnt from either explicit or implicit feedback provided by the user. But, past experiences show that that only a few numbers of users provide information about their preferences explicitly. The FRC uses implicit feedback to capture the preferences which are stored in the user's profile for future personalized recommendations. The context aware method provides recommendations to the users regarding their environment and the details of the situation in which they are thus personalizing user's experience. The proposed approach is yielded to overcome the problems caused by ignoring the contextual information. Most of the existing systems use the data from the individual user combined with the data from other users to make a recommendation. The current system only uses the data from the user to provide the feedback. This results in more personalized recommendation.
The content based approach, rather than looking for weight of one feature, calculates the over-all weight of the item in context, which is more important when the recommendation is based on several attributes. Hence this relates to comparing the current item against a case base and determining the overall weight and the status of the item in terms of recommendation.

Keywords: Recommender system, weighting technique, user preference, context aware, content based recommender system

Introduction:

The Flight recommendation client (FRC) is a recommender system that searches for flights on behalf of the user. It has a good user friendly web interface. The FRC takes in the user input and generates a set of itineraries that meet the input criterion. The result set usually consists of a significant amount of flights. The user might get overwhelmed by the large number of choices. Hence a recommender system is incorporated that sorts out the results set based on the user's purchase history. To make accurate recommendations, the FRC should be able to learn the user preferences. So, the system stores information about the user preferences, every time a purchases is made. It uses the stored preferences as the criterion for sorting in the recommendation process. In addition to using the implicit feedback for collecting preferences, the FRC uses the feature weighting technique to further improve the recommendations.

The brief introduction of four types of recommender systems:

Content-based recommender system

This recommendation method is based on the item description and a profile of the preferences of the user. The items are described by keywords [5] and a user profile is built, indicating the type of items that this user likes. [1]Basically, the algorithm of this type of method try to recommend items that are similar to those that a user liked in the past, i.e., the algorithm compares items previously rated by the user with candidate items, and the best matches are recommended. The system uses a model of the user preference and a history of the user interaction to create a user profile. The system also creates a content-based profile of users based on a weighted vector of item features. These weights are important because they denote how important each feature is to the user and can be computed from individually rated content vectors using several techniques. These weights can be decreased or increased, based in the user opinion, as a like or dislike button.

Collaborative Filtering

The collaborative filtering has its methods based in users' behaviors preferences or activities and predicting what users will like most, based on their similarity to other users. This method is used to make automatic predictions - filtering - about the interests of the users, collecting information and preferences of several users [3]. This recommendation method requires, usually, the user participation - i.e. the profile analysis - and algorithms [6] capable of match people with similar interests. Basically, in this method, the user expresses his opinion by rating the items in the system, then the systems matches the user's ratings against other users and finds users with similar interests. In other words, the system looks for users that share the same rating patterns with the actual user and then it uses the ratings from those like-minded users that were found to calculate a prediction for the actual user. [4]

Hybrid Recommender Systems

The hybrid recommender system is a method that combines the collaborative-filtering and the content-based filtering, and can be more effective in some cases. This method can be implemented in different ways, using the other two methods. The first way to do that is adding content-based capabilities to a collaborative-based approach. Also, it can be done by making content-based and collaborative-based predictions individually and then combining them, or by unifying the approaches into one model. [8] There are several studies that proving that this method can provide more accurate recommendations than the other two pure methods. A famous example of system that uses this method is Netflix, that make recommendations by offering movies that share similarities with films that a user has rated highly - content-based filtering - as well as by comparing the watching and searching habits of similar users - collaborative filtering.[2]

Context-aware recommender system

In order to solve some problems of traditional recommender system, context-aware recommender system has attracted the attention of the academic. [7] Contextual information is an important factor influencing the accuracy of the recommendation. The purpose of the context-aware recommender system is to consider the position, time and some additional information in the process of recommendation.[9][10]

The means to use the contextual information can mainly be classified into two kinds in recommender system, which are as follows:

(a) Recommended by context-driven query and retrieval;

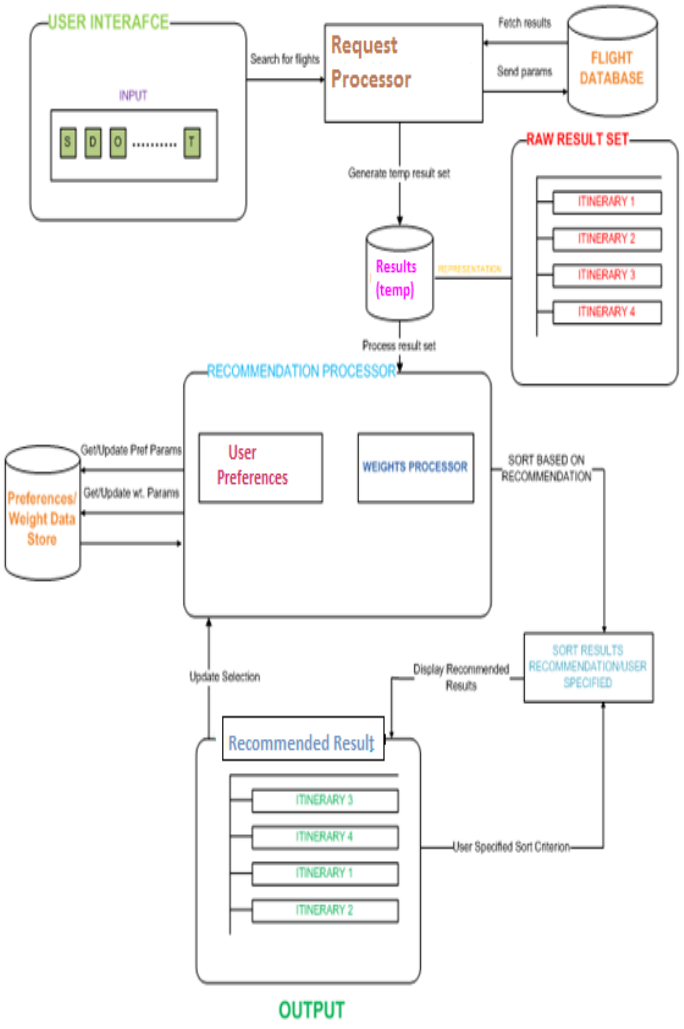
(b)Recommended by context preference extract and evaluation

The recommendation process mainly includes four stages: contextual information acquisition, user preference extraction, context-aware recommendation generated, recommendation evaluation and improvement [10].

The evaluation index mainly includes recommendation accuracy, diversity, novelty and coverage. The most used index is accuracy. In addition, user satisfaction is also used as an evaluation index of context-aware recommendation system.

Methodology:

DOI: 10.18535/ijecs/v5i9.31



The Recommendation process/design has been depicted in the above figure. Let's go to the detail description of the process.

1. We have the user interface; the user enters the input through. The inputs are captured and sent over to the query processor. There is a high level of the abstraction once the inputs are obtained.
2. The query processor forms a query to obtain the raw results that meet the input criterion. It generates a temporary result set, which could be processed in the recommender system. The bounded region labeled "raw result set" is just a representation of the result set.
3. The FRC's core component, the "recommendation processor" does the processing of the result set in two steps.

a. Preference processor: This does the computation of the total counts of the preferences. It takes the first item in the result set and determines the range of the preference attributes. Hence obtains the 'preference identity', which is the compared to the user profile to obtain the number of hits in the purchase history. Thus, determining the preference count for each attribute. This also enters the current selection into the user profile, when the user is done with the purchase process. In the process, keeps track of all the preferences and the respective cases the user has made purchases in.

b. Weights processor: It does the job of fetching the weights and updating the weights cumulatively every time the user makes a successful purchase. The updating process is associated with incrementing the weight for the preferences that are selected (purchased) in the purchase process by an incrementing fraction. At the same time, decrements the weights of unselected attribute. This adds a significant feature to the system. If over a period of time, consensus are drawn on the overall preference counts for all the users and a particular preferences is bound to have a significantly higher weight over the others, it can be down by just changing the incrementing/decrementing fraction. The flexibility of the weight change sums up to one the many significant features of the FRC.

The recommendation processor does a quick computation on the results obtained from the preference and weight processors and determines the 'scores' for each and every item in the result set.

4. The scores obtained from the recommendation processor are ranked in the 'Sort results process' thus generating the final 'recommended result set'.
5. Selections made in the recommended result set are updated in the user profile. The final result set can be further sorted based on the user specified sort criterion.

Proposed architecture:

We have suggested an algorithm to recommend customers or users flights based on their searches and past experiences:

For each impression:

Flights to be recommended $=\sum_{i=1}^{k}\left((2^{rel\,i}-1)/\log_2(i+1)\right)$

reli =
5: for booked flights
1: for clicked or searched flights
0: for all the rest

The Flight recommendation algorithm had been designed with the concept of implicit feedback and Feature weights in mind. Calculations are made to compute the total weight of each entry in the result set. The results are then sorted based on these weights.
The following is a detailed description of the algorithm.

Pre-conditions:

1. The status of the user login is known. (if the user is logged in or not)
2. The user has entered the input

Post condition:
The results are sorted in the order of recommendation with the most recommended result at the top and the least at the bottom.

Algorithm:
The set of flights that meet the user input criterion are fetched. Let's call the temporary unsorted result set $R_{temp}$and the size of the result set 'n'. The algorithm needs to compute the value R_{temp} (i, w) where (0 <i< = n) and w represents the weight. Hence R_{temp} (i, w) represents the weight w of an itinerary i in the result set R_{temp}.

There are four attributes based on which the total weight can be calculated Price P, Airline A, Time of the day T and stopover attribute S.

R_{temp} (P_i), R_{temp} (A_i), R_{temp} (T_i) and R_{temp} (S_i) represent the value of price, airline, time of day and stopover for the itinerary i respectively where (0 <i<= n).

Let's assume that for itinerary i, $P_{id,}$ $A_{id,}$ $T_{id,}$ S_{id} represent the identities of the values R_{temp} (P_i), R_{temp}(A_i), R_{temp}(T_i) and R_{temp}(S_i) in the database.

Calculating the identities for the itinerary i.
R_{temp}(P_i) is compared to the average of all the prices in the $R_{temp}$to determine the price identity $P_{id.}$
The average of n prices in $R_{temp,}$ $R_{tempPriceAvg}$is calculated using the following expression.

$R_{tempPriceAvg}$ = $\sum R_{temp}(P_i) / n$ where n is the total number of itineraries in $R_{temp}$and 0<i<= n

A_{id} is nothing but the value of R_{temp}(A_i)

T_{id} is calculated based on the R_{temp}(T_i)'s value range which is nothing but the quarter of the day. For example if 0<= R_{temp}(T_i) < 6, the T_{id} would be 1, since it is the first quarter of the day.

S_{id} is the value of R_{temp}(S_i)
Now that we have the identities P_{id} A_{id}, T_{id} and S_{id} for the itinerary i in the result set Rtemp,
The weight R_{temp} (P_i, w), R_{temp} (A_i, w), R_{temp} (T_i, w), R_{temp} (S_i, w) of the respective fields is attained from the database by counting the number of occurrences of each identities $P_{id,}$ $A_{id,}$ $T_{id,}$ S_{id} , for the User multiplied the current value of the respective weight W(P_{id}), W(A_{id}), W(T_{id}), W(S_{id}) filed for the user.

The weights R_{temp} (P_i, w) = count (P_{id}) * W (P_{id})

Where count (P_{id}) is the number of total occurrences of the value P_{id} in the database and W (P_{id}) is the current weight of the attribute P_{id}.
Similarly,
R_{temp} (A_i, w) = count (A_{id}) * W (A_{id})
R_{temp} (T_i, w) = count (T_{id}) * W (T_{id})
R_{temp} (S_i, w) = count (S_{id}) * W (S_{id})
Now that we have all the values required, the total weight for the itinerary i is computed by adding the obtained values
R_{temp} (i, w) = R_{temp} (Pi, w) + R_{temp} (A_i, w) + R_{temp} (T_i, w) + R_{temp} (S_i, w) for 0 <i<= n

Thus the final result set R_{final} is obtained by sorting the result set R_{temp} based on the values of R_{temp} (i, w) in the descending order.

DOI: 10.18535/ijecs/v5i9.31

Experimental work:

To evaluate the effectiveness of the method, performance is measured using two factors like precision and coverage. Recommendation precision means number of correct recommendations i.e. proportion of relevant recommendations to the total number of recommendations. Precision is given by the formula,

$$\text{precision} = \frac{(T(p) \cap R(p))}{R(p)}$$

Coverage of the system is the proportion of relevant recommendations to the all pages that should be recommended. Where R(p) is recommendation set and T(p) is session. Precision of the recommendations are measured for varying number of recommended pages. So based on above proposed system we have worked on practical evaluation using PHP.

Comparative study of precision rate between proposed and existing method based on number of pages ranked.

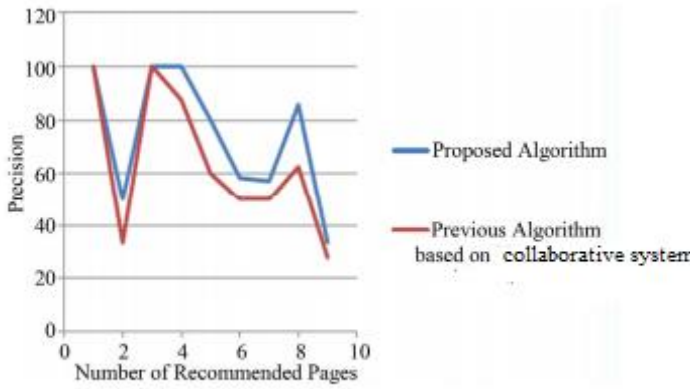


Figure Precision comparative analysis.

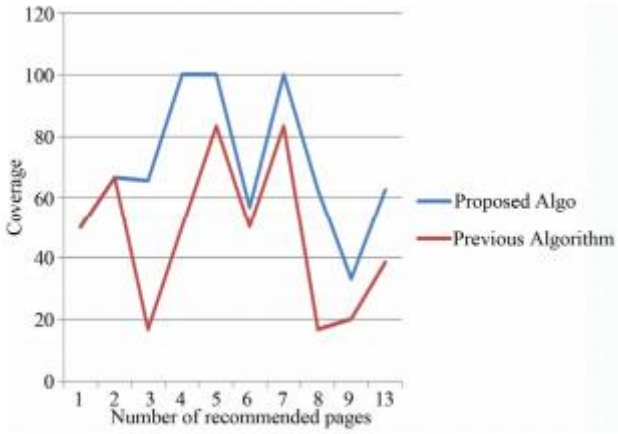
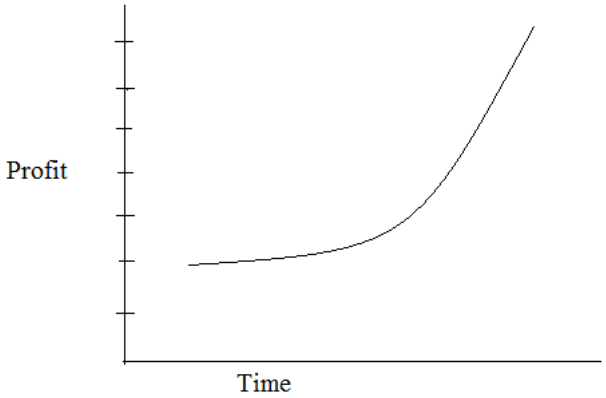


Figure Coverage comparative analysis.

The following graph determines the expected output of our research, it reflects percentage of profit increased by implementing recommendation system into online flight booking over a period of time.

While implementing recommendation systems in an e commerce business plan the profit showcased by its use is accounted to be around 20%, but in the long term as represented by the graph below, profit increases exponentially.



Conclusion:

This report describes the Flight Recommendation Client (FRC), an application that enables the user to search for and receive recommendations to find flights. The search result comprises a large number of itineraries. Hence the recommendation system cuts down the burden on the user by recommending the flights that best fit his preferences by placing them at the top.

Explicit feedback leads to an increase in the user's frustration level. The recommendation process uses implicit feedback. The feedback is obtained by making some observations on the

user's purchase history. The FRC makes successful recommendations by using the technique of assigning feature weights and user specific preference based recommendations. The recommendation algorithm was designed with the Flight Recommendation client (FRC) application in mind. Yet, it can be applied to other domains that use implicit feedback. The key to the success of the FRC is the ability to sort the results based on user preferences and personalizing it in the recommendation process.

Challenges and Future Research Directions:

The influence of various parameters on the recommendation process is therefore currently of major interest. This challenge has been identified by several researchers.

1. Discovering valid context types and instances and then implementing them are therefore serious challenges that CARS should face and resolve.
2. Identify the development of high performing context-aware recommender systems and testing them on practical applications as an important challenge.
3. Another important challenge is the evaluation and lack of publicly available datasets. In order to assess the impact of various contextual parameters, datasets are needed that contain contextual data.

Future work:

Context acquisition: majority of the recommender systems rely on a combination of explicit, implicit and/or inferred contextual data. The context sensors that can be used are Computing context, Location context, Time context, Activity context, Social relation context etc.

Different methods can be proposed to incorporate contextual information in the recommendation process, including recommendation via context-driven querying and search, contextual pre-filtering, contextual post-filtering and contextual modeling. Many systems can be developed which rely on a recommendation via context driven querying and search method. These systems match contextual data to resource metadata in order to retrieve suitable resources. So this method will help in suggesting correct recommendations to people depending upon what their taste and need is.

References:

[1] Farman Ullah, Ghulam Sarwar, Sungchang Lee, Conference on electronics, telecommunications and computers, CETC 2013, procedia technology, 17(2014) 528-533

[2] Poonam B. Thorat, R. M. Goudar, SunitaBarve, Survey on Collaborative Filtering, Content-based Filtering and Hybrid Recommendation System, international Journal of Computer Applications (0975 – 8887) Volume 110 – No. 4, January 2015

[3]Yoon Ho Cho, Jae Kyeong Kim, SoungHie Kim. A personalized recommender system based on web usage mining and decision tree induction, 2002

[4]Nguyen Thai Nghe, Lucas Drumond, ArtusKrohnGrimberghe, Lars SchimdtThieme, Procedia computer science`1(2010) 2811-2819

[5] Kwan ghee Hong, Hochoeol Jeon, Changho Jeon, User profile based personalized research paper recommendation system, 2011

[6] Kwan ghee Hong, Hochoeol Jeon, Changho Jeon, personalized research paper recommendation system using keyword extraction based on user profile, journal of convergence information technology, 2013

[7] A. Tejeda lorente, J. bernabe Moreno, Cporcel, e.herreraviedma, integrating quality criteria in fuzzy linguistic recommender system for digital libraries, 2014

[8] A Hybrid Web Recommendation System Based on the Improved Association Rule Mining Algorithm Ujwala H. Wanaskar ,Sheetal R. Vij , DebajyotiMukhopadhyay, Journal of Software Engineering and Applications, 2013,

[9] Yan Liu, Yangyang Xu, Mei Chan, context aware recommendation system with anonymous user profile learning, SEKE-2015

[10] SofianeAbbar, MokraneBouzeghoub, context aware recommender system: a service oriented approach, 03 march 2016