Time Effective Vehicle Booking: A Novel Data Online Cluster Search Recommender System

Priyanka¹, Kanojia Sindhuben Babulal²

¹ M. Tech Research Scholar, Department of Computer Science & Engineering, UIT Allahabad, U.P., India.
² Assistant Professor, Department of Computer Science & Engineering, UIT Allahabad, U.P., India.

ABSTRACT

In mundane life of everyone, people are tending towards extracting some external knowledge provided by recommendations so as to get decisive about an artifact of interest. Recommender systems or sometimes also called as recommender engines automates the process of recommendations. They are of three types: content based approach, collaborative approach and hybrid approach. In this thesis improvisation of hybrid approach is executed by proposing and developing a Data Rendering Cluster Search approach. Mechanisms of both the approaches are almost similar, basic difference though lies in the technique of portioning of the dataset fed to both the algorithms. In Data Rendering Cluster Search approach location is partitioned forming clusters and searching is done into the cluster nearest to the user to provide cab recommendations, thereby giving better time minimization than hybrid approach. Data Rendering Cluster Search is user friendly and easy to anticipate. It provides better time efficiency and high accuracy than former approach and thus can be counted as one of the best location based recommender system.

Indexing terms/Keywords

Recommender System, Data Rendering, DRCS, Data Division, online vehicle recommender system.

1. INTRODUCTION

The goal of a recommendation system is providing meaningful recommendations to a collection of users for items or products that are interesting to them [1]. Some real world examples are books on Amazon, movies on Netflix, social media and e-commerce sites etc. The design of recommendation system depends on characteristics and domain of the data available. For process of decision making, obtaining recommendations from trusted resources is a critical component, it brings user desirable properties. This thesis focuses on online cab booking recommender systems which is becoming important tool for both customer and service provider to fetch information of availability of cabs on request made by the user to their nearest possible location. With this location based tool searching of the cabs based on required location, suggestions to the user while booking a cab, is made easy. They tend to provide suggestions and services to user in easy and fastest way possible [2]. Some of the famous service providers are OLA cabs, Uber Cabs, Taxiguide etc. which uses online location based recommender system to provide information about the cab to customer and information about the customer to the cab driver after filtering information required by processing user’s query request [11].

The approaches of recommender systems can be broadly categorized as

1. Content Based Approach
2. Collaborative Filtering Approach
3. Hybrid Approach

1.1 Content Based Recommender System

Recommended items are the items that are similar in content that a user has liked in the past or matches with the attributes of user. i.e. CB suggests items based on item features or user interest profiles. Personalized profiles are created automatically through user’s description and feedback.

1.2 Collaborative Filtering Recommender System

A user is recommended items based on the past ratings of all users collectively upon those items. User feedback is collected in form of ratings for an item in a given domain. CF is a process of evaluating information using information of other people [3]. CF has become one of the most prominent personalization technique which is being improved continuously.

1.3 Hybrid Recommender System

In order to leverage the strengths of content-based and collaborative recommenders, there have been several hybrid approaches proposed that combine the two. One simple approach is to allow both content-based and collaborative filtering methods to produce separate ranked lists of recommendations, and then merge their results to produce a final list [12]. Hybrid approach can also be seen as combination of two or more recommender systems for achieving synergy between them.

2. LITERATURE REVIEW

In this area some work had done by many researchers. Good et al. [10] use collaborative filtering along with a number of personalized information filtering agents. Predictions for a user are made by applying CF on the set of other users and the
I S S N 2 2 7 7 - 3 0 6 1  
V o l u m e 1 5  N u m b e r 1 3  
long_right = ((1-lon_right) - 1);

String insertQuery = "insert into vehicleLocation (vehicleName, lat, long, lat_left, lat_right, long_left, long_right) "+ "+values ('car "+j+Math.random()+","+lat+","+long+","+lat_left+","+lat_right+","+long_left+","+long_right+"');
//Fetch User Location (in lat/ long)(eg: lat=48.6142, long = 52.4649)

- Loading appropriate database using select query given below:

"select id, lat, long from vehicleLocation where lat_left in ("+lat_left+","+(lat_left+1)+","+(lat_left-1)+") and long_left="+long_left

//Query Result will be number of rows of vehicles that are around 200Kms from user location.

- operation performed on each row of query result

Foreach(data i-n)
{
LatLong loc1 = new LatLong(userLoc_Lat, userLoc_Long);
LatLong loc2 = new LatLong(data(i).Lat, data(i).Long);
Double distance = LatLongTool.distance(loc1, loc2, LengthUnit.KILOMETER);
//distance will show the exact distance between user and vehicle.
}
//Time comparison is taken from existing approach to new approach, which is observed 10 times lesser.

4. EXPERIMENTAL SETUP AND RESULT ANALYSIS

For developing this application, we are using JDK 1.8 that is JAVA developing kit, i3 processor, 4 GB Ram, Eclipse environment for the result experiment evaluation also used MySQL connector and simple latitude, longitude jar.

Table -1: Comparison analysis of Hybrid and DDRS approaches

<table>
<thead>
<tr>
<th>Algorithm Name</th>
<th>Total Data (rows)</th>
<th>Data Loaded for Business Logic Execution(no. of rows for searching)</th>
<th>Time (milli sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRCS</td>
<td>5000</td>
<td>5000</td>
<td>1640</td>
</tr>
<tr>
<td>Hybrid</td>
<td>5000</td>
<td>2000</td>
<td>146</td>
</tr>
</tbody>
</table>

The above table depicts a statistical comparison of two approaches. Time minimization s observed and compared on executing both algorithm for recommender system. An example of cab booking is taking into an account in any city location. Suppose a user has to book a cab online in Allahabad region of Uttar Pradesh at Civil Lines. The information of specific location is provided at the time of online booking request. When the booking process is completed, the cab company recommender system obtains their nearest cab locations based on booked location.

In Hybrid approach the process of finding taxi is executed by searching all their cabs one by one in that region. Thus the searching time of the cabs in that location is of longer span as compared to DRCS approach.

In DRCS approach mechanism is almost same as in Hybrid approach, having main difference in their technique is that in DRCS approach location is divided into small clusters and searching is performed on the cluster that matches to the customer’s requirement. With the help of DRCS approach we divide the user and cab location into latitude left-right part and longitude left-right part. Cab locations saved into small clusters according to user requirement and their location. Thus Cab searching is easily done from these clusters. Time taken in performing the search for cab minimizes as the cluster searched is the one that matches to location nearest to user’s location and not on all the data in the database like in Hybrid approach.
The above graph depicts query processing result for specified dataset and is printed using chart library. The x-axis represents data and y-axis represents the time taken in performing the search on given dataset. Analysis shows that the DRCS outperforms the Hybrid approach. On increasing the required data as per requirement, above graph clearly shows that the time taken is much less than that in Hybrid approach. The graphical representation shows that the efficiency of the DRCS approach is more and it outperforms the accurate time minimization for cab booking. Thus, DRCS is more time efficient and provides better accuracy than the previous approach.

5. CONCLUSION

In the thesis DRCS approach was introduced for recommendation system for online cab booking with higher accuracy, efficiency and time minimization. Comparisons of the Hybrid and DRCS approach are given where later technique outperformed the former one. Hybrid makes use of Content Based and Collaborative Filtering approaches in the recommender system. DRCS makes use of data division cluster search to produce more accurate result for the available dataset. In DRCS approach mechanism is almost same as in Hybrid approach, having main difference in their technique is that in DRCS approach location is divided into small clusters and searching is performed on the cluster that matches to the customer’s requirement. With the help of DRCS approach we divide the user and cab location and searching is easily done from these clusters. DRCS approach is time efficient as well as provides high accuracy instead of hybrid approach. Working on best location based recommender system model such as the proposed technique, DRCS is user friendly and easy to anticipate. It provides better time efficiency and high accuracy than former approach and thus can be counted as one of the best location based recommender system

6. REFERENCES


[6] An Effective Taxi Recommender System Based on a Spatiotemporal Factor Analysis Model Yu-Ling Hsueh, RenHung Hwang, and Yu-Ting Chen Dept. of Computer Science & Information Engineering, National Chung Cheng University, Taiwan fh sueh@cs.ccu.edu.tw, rhhwang@cs.ccu.edu.tw, wendy10445@gmail.comg 2014


[9] Y. Yue, Y. Zhuang, Q. Li, and Q. Mao, Mining timedependent attractive areas and movement patterns from taxi trajectory data, in the 17th International Conference on Geoinformatics, 2009, pp. 16.

