Place Recommendation System

Using Clustering and Collaborative Filtering

Paras Kumar
2016CSB1047
Contribution: 50%

Nitin Gandhi
2016CSB1045
Contribution: 50%

1. INTRODUCTION AND BACKGROUND

Big Tech companies like YouTube and Netflix recommends us videos to watch according to our activity. Recommending relevant items brings traction to the company. In a world full of data, it is necessary for us to get relevant, customer specific recommendation. In our final project of data mining course we have built a recommendation system using the BrightKite dataset.

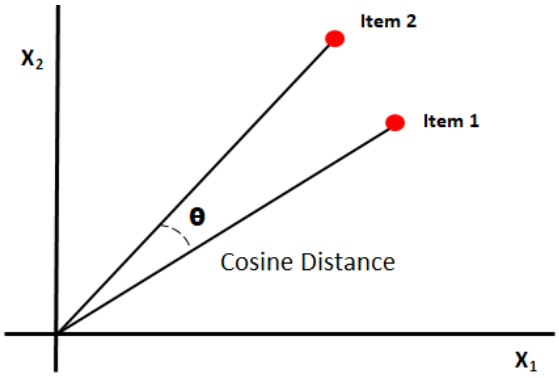
2. AIM

To Recommend interesting places to visit based on other users check-in history who are most similar to the given user and using a fast clustering algorithm to find major regions/places.

3. APPROACH AND METHODOLOGY

Assumption: Since we do not have the rating of the places that the users are visiting so we are assuming that the person who attends a place likes it. This system can be improved if we have an explicit rating system and review mechanism. Currently, we do not have that data, so we are sticking with the implicit ratings.

We need some similarity metric sim(x, y) for similar users to capture the intuition of their likes. For each user we have an array where if the an entry is one then the user has visited this cluster otherwise it is zero. Then we calculate similarity between two such vectors and take the K most similar users based on the places(clusters) they have checked-in and check for the most occurring location. This most occurring location is the recommended location, we are using Google Maps API service to get name of recommended places.



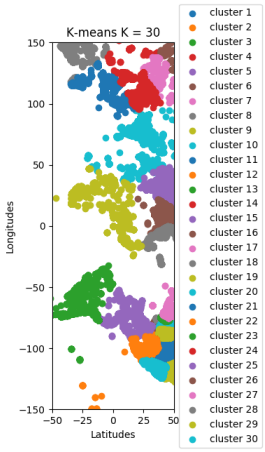
: Fig 1: Cosine Similarity

USER ID	CLUSTER 1	CLUSTER 2	CLUSTER N
1	1	0	0
2	0	1	1
M	1	0	1

: Fig 2: Finding K most similar users

3.1 Task 1: Clustering

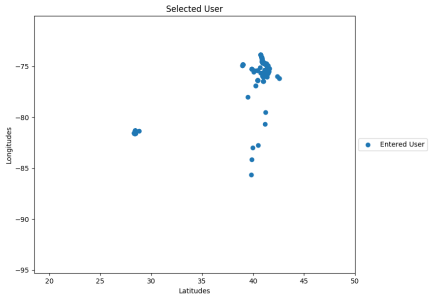
There are total 4,747,287 check-in records, so we have used K-means clustering algorithm which has a running time of O(kn). It takes about 3 minutes to cluster the complete dataset.



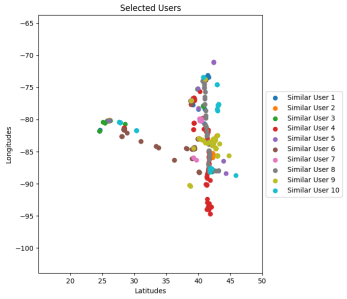
: Fig 3: Clusters

3.2 Task 2: Recommendation Logic

We use the clusters found in previous step to construct a matrix where every row represents a user and every column a cluster. It is not possible to use location ids in the column because there are around 4 Lakhs of such ids and then the matrix will have around 4 Lakhs x 55k users which can not fit in the memory, so we are using clusters. So, now we compare the similarity between a given user and rest of the users and find the K most similar users. We then simply find the most frequent clusters or places for these K users and these are the location ids we will recommend.



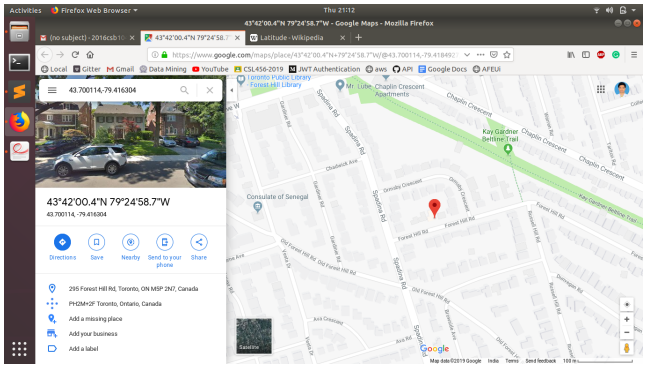
: **Fig 4: Entered user**



: **Fig 5: Similar users found**

3.3 Task 3: Recommending a place

We first convert the location id to geolocation then we use Google Maps API service to get the name of location from geolocation. The corresponding location is opened in new tab of the default web browser and the user can easily explore the nearby places on his web browser.



: **Fig 6: Recommended place**

4. EXPERIMENTAL RESULTS AND EVALUATION

4.1 Experimental Settings

We have used the complete BrightKite dataset and our recommendation system can handle very large amount of data, since the complete system has running time of O(kn). We are running k = 10 iterations of K-means and it takes about 3 minutes to cluster the complete data and writes these clusters to file called cluster.txt which can be then used by the recommendation system. We don't run K-means again unless the check-in data changes.

4.2 Evaluation Measures

Since, we can't directly analyze the quality of our recommendation until there are some users who can explicitly or implicitly rate our recommendation. So, we are relying on some indirect measures to analyze our results. We are plotting graphs after every task to check if that individual step gave correct results and we have assumed that if all the plots generated are correct then our final recommendation will also be correct.

4.3 Experimental Results

We have successfully built a recommendation system which asks for user id and opens the recommended place in the default browser. In particular we get the recommended place for user id = 2000 as shown in the Fig 6. If we turn off plotting, we can get recommendation under 1 min of running time. So, this kind of recommendation system can be practically used with little bit of improvement like caching support.

4.4 Discussion on Results

After completing Task 1 we get the clusters (Fig 3) which represents major locations on the map then after Task 2 we get the similar users (Fig 5) which is actually similar to the user under observation (Fig 4). Then finally after Task 3 we get the recommended place with latitude = 44 degree and longitude = 79 degree which is the most visited place by the similar users (Fig 5). So, we can say that our recommendation system is performing as expected and it is producing correct results.

5. CONCLUSION

This was the first ever recommendation system which we created and we learnt what common challenges one has to face to build an efficient recommendation system. Testing the recommendation system is as hard as deigning a recommendation system and such systems are supposed to be intelligent so we can never have a perfect recommendation system, there is always some room for improvement.

6. GITHUB REPOSITORY LINK

https://github.com/parasKumarSahu/Place-Recommendation-System