

A LOCATION-BASED RECOMMENDER SYSTEM FRAMEWORK TO IMPROVE ACCURACY IN USER- BASED COLLABORATIVE FILTERING

Kasra Madadipouya

Department of Computing and Science, Asia Pacific University of Technology &
Innovation

ABSTRACT

Recommender systems are utilized to predict and recommend relevant items to system users. Item could be in any forms such as documents, location, movie and articles. The mechanism of recommender system is based on examination which includes users' behaviors, item ratings, various logs (e.g. user's history log) and, social connections. The main objective of the examination is to predict items which have great potential to be liked by users. Although, traditional recommender systems have been very successful to predict what user might like, they did not take into consideration contextual information such as users' location. In this paper, we propose a new framework with the aim of enhancing accuracy of recommendations in user-based collaborative filtering by considering about users' locations.

KEYWORDS

Recommender Systems, Collaborative Filtering, Pearson Correlation Coefficient, Location-based recommender, Content-based filtering

1. INTRODUCTION

In our daily life when people face with unfamiliar options that they cannot make decision about selecting appropriate alternative, often they ask their friends to guide them and help them to choose the right thing. Besides that, sometimes, people read the magazines and newspaper reviews to make up their mind and make right decision. Nowadays, online recommender systems have begun to provide a technological proxy for this social recommendation process [1], in which they could be used to determine that if a certain user would like a specific item via making prediction, or to recommend top K items (top-K recommendation) to a user based on user's preference and analyzing user's behavior. Recommendations and recommender systems are used widely in various areas. All systems utilized their own data plus implement standard recommendation algorithms according to their needs to optimize their recommender systems. Research in recommender systems reflect two algorithmic approaches are utilized in recommender systems which are collaborative filtering and content-based filtering.

Collaborative filtering divided to two different types which are known as memory-based collaborative filtering and model-based algorithm [2].

The memory-based algorithm is called as classic approach which provides recommendations to users based on utilizing statistical equations. Memory-based approach is mostly used for user-based collaborative filtering. In [3] researchers, employed user-based method to develop a system which was based on Usenet. Their system, allowed all registered users to rate articles based on their interest. Gathered rankings from the users were used to make predictions for other users and

gave ranked articles to them. Ranks of users for various article compared with each other and made recommendations for the users who have similar ranking. The system was based on this concept that if users have the same ranking in the past they more likely to have similar tastes in the future again. In [4] author establishes a new way to compute the similarity between items. They employed set of items as nearest neighbors to make recommendations. The new approach is called as item-based collaborative filtering.

Additionally, collaborative filtering can be utilized with various types of model-based approaches. In short comparison, in model-based approach user database is employed to learn a model which lately helps in making predictions for users, while in memory-based approach operations are done in whole user database for predictions [2]. A prediction could be a value which is acquired based on calculation about a user. For model-based approaches which are based on collaborative filtering two well-known algorithms are Bayesian networks and cluster models [2].

Cluster models utilize like-minded users as clustered classes. In cluster models users' ratings determine a specific user should be placed in which class-membership. In Bayesian way titles and their values which are allowed for ratings are considered as variables Bayesian network. In this way, the recommender system is able to learn the structure of the network which is for encoding the dependencies between titles, and the conditional probabilities [5].

Although collaborative filtering is widely used in industry but it has some issues when it comes to recommending items to users. For instance, if a new item is added to a system, the system cannot recommend the item to users at first since the item is not received any rank from users. In other word, the item will never be recommended unless the item is received rank or finding similar items to the current item [6].

This issue is known as cold start. The cold start problem is just not include new items but it includes new users or users who have not rate any items as well. This means that the system cannot provide recommendation to new users at first. Another issue happens when a user interests is very unique and a system could not classify the user or find a similar peers to the user which finally leads to poor recommendations [7].

The second type of recommender system is known as content-based recommender systems. In content-based filtering by contrast of collaborative filtering, items are selected to recommend based with comparison of the user's preference which is obtain from user's profile. Basically, content-based approach tries to give recommendations to user based on the items that the user liked in the past, however, collaborative approach recommends items to a user based on other users preferences who have the similar taste with the active user [6]. Therefore, it can be said that in content-based approach items are recommended to users based on the item description/information instead of other users' reviews or preferences [8]. Algorithms that are used for implementing content-based filtering are standard machine learning methods such as naïve Bayes classification which is used widely. In naïve Bayes approach it assume that probability of every word event is related to the document class but it separate or independent of the word's position and context.

The rest of the paper is organized as follows, in the next section a review regarding related works on how to consider users' location to improve the quality of recommendations is provided. In section three dedicated to the new proposed framework to improve accuracy in user-based collaborative filtering by considering users' locations in details. Finally, in the last section, conclusion and future work are discussed.

2. RELATED WORK

The general drawback of the baseline collaborative filtering is that users' locations are not considered in providing recommendations as well as in clustering or grouping stage, since in the reality mostly the users who are located in the same location or in near location with short distance almost have the same or at least close tastes to each other [9]. Therefore, in order to provide better recommendations more in depth analysis regarding the users; preferences required by recommender systems [10]. One approach to enhance the quality of recommendation is utilizing the users' locations. This type of recommender systems are known as location-based recommender systems [11].

In location-based recommender system, recommendations are provided by considering about users locations such as where they are from as one the important parameter in providing recommendations to users. The information regarding users' locations mostly acquired in an explicit form from users [12][13]. Users' locations can be highly crucial and influential in categorizing users according to a specific type of product or even classification based on taste similarity. In a work from [14], demographic information (including users' location) has been utilized in recommending books as well as suggesting range of products and services. Based on certain situations content-based and collaborative approach might not be able to provide recommendation to users particular in a situation when a user is new to the system which this phenomenon called as cold start problem [15]. In location based recommendation system is constructed based on correlations among system users which up to some extent very close to collaborative technique with the difference that type and usage of data inside of the matrix is different. The benefit behind demographic recommendation is to reduce dependency of the system to track customer history activity which required providing recommendations by content-based and collaborative filtering [16].

Among early works in recommendation systems, [17] suggest a method which utilized demographic groups from marketing research in order to provide a range of services and products. In the method, brief survey conducted for data gathering for user's categorization. In another work by [18], an approach introduced which machine learning was used for demographic data classification. Another example of demographic recommender system is LifeStyle Finder [19]. In that system which was implemented in the United States, population of U.S divided into 62 clusters according to the people lifestyle characteristics as well as their responses to survey and purchasing history which recorded in by the system.

In another work by [20], a new recommender system for tourism industry was introduced which was called more Tourism. The system designed based on a hybrid recommendation algorithm which provided very useful information regarding tourist attraction to users according to analysis of their profiles (location, age, etc.) which included tourist attraction locations, schedule and amount of time needed to visit the place. Since, moreTourism designed for mobile platform and used mobile connectivity facility, the system helped customers to make decisions based on their location, timetable, context and mobility needs.

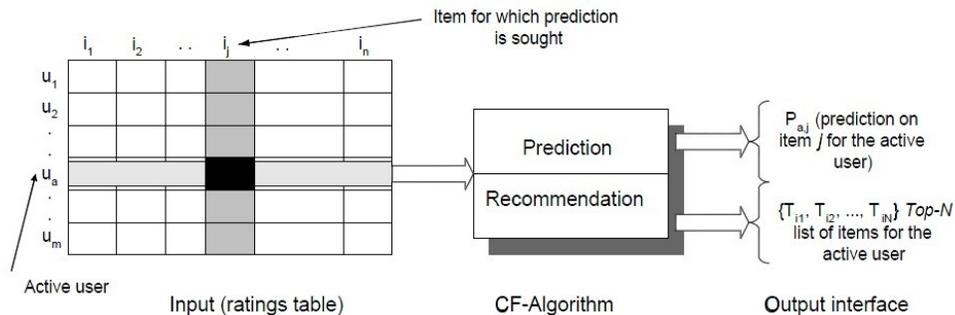
In the proposed approach content-based filtering and collaborative filtering utilized with taking social recommendations into consideration based on provided tags by customers. Tags used for establishing customer's cloud tag which held weighted customer's tags based on their ratings and compared with all tags provided by various customers. Furthermore, the system established a folksonomy as an undirected graph to represent relation between tags.

3. THE SYSTEM ARCHITECTURE AND PROCESS

As mentioned earlier, the basic idea of this framework is to consider about the users' locations in providing recommendations by using baseline algorithm. By contrast of the baseline collaborative filtering approach that does not consider about the user location in providing recommendations, in this frame work current user location is considered in which user gets different results based on the location. This reflects the fact that if the user changes his/her location the received recommended items will be affected because of changing in the location. Secondly, user preferences that lead him/her to select an item, and correspondingly potentially generates better recommendations based on the user location which increases the locality of the recommended items. In addition to that the framework improves the locality of the recommendations in both peers selection stage and item recommendations with gathering some information about user location and considering about user location in all recommendation processes.

The framework contains the following steps as demonstrated in the following part,

- 1- Reading user profile from the database and keep the user profile in the main memory (for memory-based approach).
- 2- Applying similarity index function for the user to find the peers.
- 3- Adding α for those peers who are in the same place with the active user.
- 4- Applying some filtrations on selected peers to reduce and remove non-related peers.
- 5- Selecting items for providing recommendations to the active user.
- 6- Showing the recommended items to the active user.



Figure_1 (Overview of collaborative filtering process)

As it seems from the above figure the framework is based on the collaborative filtering in which first the users necessary information about the active user will be stored in different database from the user profile in memory since some temporary analysis and changes might be applied in the user profile which will not be overwritten on the user profile or change it. After copying it, the user profile will pass through from peers selection stage, in that stage, firstly, similarity function is applied for the active user against other users of the system to find which user has the similar taste to the current user. Then peers are selected for the active user. In other word, users will be classified in different groups which each group has the users with the same taste. After that, peers location will be compared with the active user and those who are located in the same geographical location with the active user, their similarity indexes will be increased by adding constant value which is known as α . Then some filtrations will be applied to the peers who are called threshold-based selection, according to which users whose similarity exceeds a certain threshold value are considered as neighbors of the target user [21] and be selected as final peers to

give recommendations to the active user based on them. Hence, in this stage all peers who have less than 0.5 similarity index (threshold) will be aborted and will not be considered in further steps. After applying filtration to the users in different peers groups, top peers of the active user will be nominated for providing recommendations. Finally, top rated items of the peers who gained the highest similarity indexes will be selected for recommendations. In the final stage, those recommended and selected items are shown to the active users as recommendations.

In addition to what mentioned in above, the requirements of the systems are listed as below, Requirements:

- User
- User profile
- Database of items
- User demographic location (include in user profile database)
- Similarity index function (here Pearson Correlation)

Based on the required data for providing recommendations, the output of the framework would be,

- List of selected peers $(p_1, p_2, p_3, \dots, p_n)$
- List of recommended items $(i_1, i_2, i_3, \dots, i_n)$ for active user (u)

Furthermore, all steps that have been taken in the framework with applying collaborative filtering alongside of Pearson equation to provide recommendations are explained in the following list.

- 1- Finding peers for the active user.
- 2- Gaining geographic locations of peers and active user from database.
- 3- Weighting peers based on the location closeness to the active user.
- 4- Applying filtration to exclude non-related peers.
- 5- Select highest ranked peers.
- 6- Recommend items based on top peers ranked items.

Now in the following section all elements and steps of the framework are explained in details.

3.1. Creating user profile phase

The first step in this recommender system is to create user profile for the process of the recommendation. As expressed before there are two types of user profiles are created for this recommender system. The first one is created by the user and the second one is created by the system to keep some necessary data related to the recommender system algorithms and the system as well. In the first user profile, user should explicitly determine his/her geographic location and also his/her preferences and his/her mood if the user is interested. The core component in the user profile that created by the user is his/her geographical location. This means that the user is asked to provide information about location explicitly. Additionally, if the user will not provide any geographical location, the system could not apply any locality algorithm and locality improvement for the user and hence, the system functions such as standard approach. The geographic location allows the algorithm to provide better recommendation for the user with using user location in some steps of recommendation calculations.

For the system profile the framework scans the database and keeps necessary fields about the user such as user location, number of the rated items and list of the rated items from the database. The gathered information will be kept in the main memory of the system since the selected approach is memory-based. Some changes could be applied in system profile but these changes are not affected the user profile and are just useful and utilized for the framework.

3.2. Applying similarity index function

In the baseline approach the recommendations are suggested just by applying the similarity index between the user and other users. The following equation demonstrates Pearson similarity function.

$$sim(a, b) = \frac{\sum_{p \in P} (r_{a,p} - \bar{r}_a)(r_{b,p} - \bar{r}_b)}{\sqrt{\sum_{p \in P} (r_{a,p} - \bar{r}_a)^2} \sqrt{\sum_{p \in P} (r_{b,p} - \bar{r}_b)^2}}$$

The outcome of the similarity index is a number between -1 to 1 which reflects the lowest and highest similarity between the active user of the system here and a user of the system respectively. For the above formula, the average rating of the each user is also necessary. This average is gained by dividing the number of the user rated items to the total user ratings. The average of the user rating could be gained offline means before calculating the similarity index. For storing the average rating, acquired numbers should be saved in separate table of the database, however, offline processing of the average rating has one disadvantage. The drawback of this method is if a user rates new item(s), the average rate is changed but in the database the old one is stored. The most common solution for this issue is to update the table of average rating periodically when the system load is low. Duration for updating is also highly depends to the system user and system activeness, for instance, for the news website that daily hundreds news could be added to the system and the users are highly active the average time has to be much shorter than an e-commerce system. The results of the similarity index of each user is kept in a separate table in the database. The selected type for the similarity index could float or double since the number has floating point.

Now, if applying similarity index for each active user expanded to the all users of the system then for each user of the system one similarity index array is existed. In other word, each user of the system is compared with the other users of the system based on the similarity function. For instance, if the system contains 5 users then each user is compared 5 times with the other users and at the end the overall number of the comparison would be 25 which is the size of the array/table that is kept in the database. This array contents similarity number of all peers and in the next step, this array will pass through some filtration to remove those users and just select top candidates as final peers for providing item recommendations. The below figure depicts the array of the similarity index.

Users \ Similarity	1	2	...	a	...	b	...	n
1								
2								
...								
...				r				
...								
u				r		?		
...								
...				r				
...								
...				r				
...								
...				r				
...								
m - 1								
m								

Figure_2 (Similarity index matrix)

3.3. Identifies peers that are located in the same place with the active user and adding α

In this step of the framework after applying similarity index row of the similarity index from the table that related to the active user is loaded to the main memory and copy of it is created, in this step based on the system user profiles and the part of the similarity index, it is checked whether the peers are located in the same location with the active user or not. If the location of a peer is the same as the active user, then the similarity index of the active user is added with α . The constant value could be changed based on the many parameter such as the size of the dataset or sparseness of the database. In fact, the optimal value of α is more depended on the data and the users of the system. However, selection high α value could cause another issue in the recommendations which is called as overspecialization which all recommended items would be just related to the one geographical place. On the other hand, if the selected α value is too low, then the framework will lose its efficiency. Thus, the best value of the α could be the average value which in most of the cases $\{\alpha \in \mathbb{R} | 0.25 \leq \alpha \leq 0.3\}$ is the optimal value of the α in the average system. This means that the similarity index of the user who is in the same location with the active user is increased by α number which is between 0.25 to 0.30.

3.4. Peers selection phase

In this phase, after adding α value to those peers who are in the same location with the active user, those peers who have the similarity index equal or more than 0.5 are selected as candidates of recommending items to the active. In similarity index of all peers are compared with that constant value and if they are eligible they are selected for the item recommendations. This phase of the system is called filtration, in this step those peers who gained low similarity index are excluded from the list and be aborted since if they are kept on the system, could cause accuracy reduction in the final recommender lists (items). It is clear that a peer who has -1 similarity index those not have any common with the current active user and if any items would be suggested from this user will not match with the taste of the active user. In addition, the chance that those peers with the lower similarity index than 0.5 are rare except in the case that dataset of the system is really sparse those users who are involving in the system are distributed around the world and the number of the users that are from the same country are low or not existed. In such critical circumstances, the framework system automatically detects low number of peers by counting them and if the number of peers for some random selected users for sampling will be low, then the constant value of the condition will be decreased to give more chance to the users who have

uses similarity index to pass the filter. The aim of this step is to avoid starvation in providing recommendations to the active user even though that there is not exact match for the active user.

3.5.Items selection phase

This phase is the last calculation phase before providing list of recommendation items to the user. In this phase based on the shortlisted peers or top users, ten items are suggested to the active user. Selection algorithms for the items selection could be different, for instance in some data mining system the value of each recommended items are calculated and the highest rated are selected as mentioned by [22]. In another algorithm, the selection function is based on the highest rated item of each top peers. For the framework, the second algorithm is applied with little modification to consider locality of the items more in the selection. In this step of the framework, highest rated item of each user is selected to display to the user but if the number of the items are exceeded from ten, then some items should be dropped from recommendation list. Based on that those items that are nominated from peers that are in the different location from active user are nominated for removing from the list unless their rates are considerably higher than the other items. If the ratings among those local and non-local are the same then, non-local once will be selected for dropping.

In addition to that the framework also check duplication of the recommended items before going to dropping phase since some items could be duplicated. For instance, two peers rated one item high and now the item is selected by the system twice as the highest rated item of each user. In such condition, duplicate items will be removed and instead other items will be replaced for each peer who has not any item in the list.

4.CONCLUSION AND FUTURE WORK

In this paper a novel method location-based collaborative filtering framework presented for improving the overall accuracy of the recommendations in both peers selection and items selection are presented. The system is designed based on collaborative filtering and Pearson Correlation Coefficient is utilized as similarity index. Many changes are applied in the different stages of the framework in comparison with the standard approach. Besides enhancement in the peers selection stage, in selecting recommendations items some improvements have done.

In the proposed method, the geographic user location considered as constant which means the user location is not changed constantly over short period of time and it is assumed that the user access from his desk or laptop in fixed location over period of time to the news website. For the future work, it is possible to add mobile user capability to the method to provide recommendations for the users who are constantly change their locations. However, the current method may not be sufficient enough for heavy computing due to constant user location change and therefore, modification the method to enhance and overcome the heavy computing problem seems necessary. Furthermore, in this method user location is acquired from the user profile and it will rarely change but for mobile user the user's location could be gathered dynamically without any user involvement.

REFERENCES

- [1] Chen, A. 2011. Collaborative Filtering for Information Recommendation Systems.
- [2] Jannach, D. 2011. Recommender systems. Cambridge: Cambridge University Press.
- [3] Resnick, P., et al. (1994) "GroupLens: an open architecture for collaborative filtering of netnews", paper presented at CSCW '94 Proceedings of the 1994 ACM conference on Computer supported cooperative work , New York, NY, USA, ACM, p.175-186.
- [4] Sarwar, B., et al. (2001) "Item-based collaborative filtering recommendation algorithms", paper presented at WWW '01 Proceedings of the 10th international conference on World Wide Web, New York, NY, USA, ACM, p.285-295.
- [5] Vekariya, V. and Kulkarni, G. (2012). Notice of Violation of IEEE Publication Principles< BR> Hybrid recommender systems: Survey and experiments. pp. 469--473.
- [6] Mcleod, D. and Chen, A. 2009. Collaborative Filtering for Information Recommendation Systems.
- [7] Mahapatra, S., Tareen, A. and Yang, Y. (2008). A Cold Start Recommendation System Using Item Correlation and User Similarity.
- [8] Mooney, R. and Roy, L. (2003) "Content-based book recommending using learning for text categorization", paper presented at DL '00 Proceedings of the fifth ACM conference on Digital libraries , New York, NY, USA, p.195-204.
- [9] Bloch, M., Cox, A., Mcginty, J. and Quealy, K. (2010). A Peek Into Netflix Queues. The New York Times, 8th Jan.
- [10] Junmanee, S. and Hinze, A. (2006). Advanced recommendations in a mobile tourist information system.
- [11] Yin, H., Sun, Y., Cui, B., Hu, Z., & Chen, L. (2013). Lcars: a location-content-aware recommender system. In Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 221-229). ACM.
- [12] Vozalis, E. and Margaritis, K. G. 2003. Analysis of recommender systems algorithms. 2003.
- [13] Ghazanfar, M. A. and Prugel-Bennett, A. 2010. A scalable, accurate hybrid recommender system. pp. 94--98.
- [14] Burke, R. 2002. Hybrid recommender systems: Survey and experiments. User modeling and user-adapted interaction, 12 (4), pp. 331--370.
- [15] Chikhaoui, B. 2011. "An Improved Hybrid Recommender System by Combining Predictions", paper presented at Advanced Information Networking and Applications, Biopolis, 22-25 March. IEEE, pp. 644 - 649.
- [16] Burke, R. (2007). Hybrid web recommender systems. In The adaptive web (pp. 377-408). Springer Berlin Heidelberg.
- [17] Hill, W., Stead, L., Rosenstein, M. and Furnas, G. 1995. Recommending and evaluating choices in a virtual community of use. pp. 194--201.
- [18] Pazzani, M. J. 1999. A framework for collaborative, content-based and demographic filtering. Artificial Intelligence Review, 13 (5-6), pp. 393--408.
- [19] Krulwich, B. 1997. Lifestyle finder: Intelligent user profiling using large-scale demographic data. AI magazine, 18 (2), p. 37.
- [20] Rey-L'Opez, M., Barrag'Ans-Mart'Inez, A. B., Peleteiro, A., Mikic-Fonte, F. and Burguillo, J. C. 2011. moreTourism: Mobile recommendations for tourism.
- [21] Gong, S. (2010). A collaborative filtering recommendation algorithm based on user clustering and item clustering. Journal of Software, 5 (7), pp. 745--752.
- [22] Rafter, R. and Smyth, B. (2003). Item selection strategies for collaborative filtering. pp. 1437--1439.