POI Recommendations for Groups

A Geolocation Based Approach

Abhishek Agarwal*
abhishek.agarwal@tum.de
Technical University of Munich
Germany

ABSTRACT

With the increasing amount of geo-referenced data available online today, several location-based recommender systems exist for groups in the domain of Point-of-interest (POI) recommendation. POI based Group Recommender Systems (GRS) aim to recommend the most agreeable place to meet for a group of users. In this paper, we present a geolocation-based POI group recommender system that leverages the areas that the group frequents the most. We first compute the most relevant geolocation cluster of each individual based on their past rating information. We then use these clusters to find the most suitable areas for a group and employ location-based pre-filtering to generate the group profile. Later, we use Collaborative Filtering to recommend a list of POIs in the areas which are suitable for everyone in the group. Experiments on the Yelp dataset show that the proposed method improves the quality of group recommendations and outperforms the baseline approach.

CCS CONCEPTS

• **Information systems → Recommender systems**; **Location based services**; • **Human-centered computing → Collaborative and social computing**.

KEYWORDS

group recommendation, geographical space partitioning, poi recommendation, DBSCAN

1 INTRODUCTION

The enormous collection of choices available on the Web often makes it quite difficult for us to choose products or services. Recommender Systems help us tackle this issue of information overload by analyzing our preferences or characteristics and then recommending relevant products or services to us. It has emerged as an essential tool for online platforms and local-search services such as Amazon, Netflix, and Yelp in assisting users to make the right decisions and save time.

There exist several techniques to recommend products or services to individual users [4]. We can classify most of these techniques recommender systems into four broad categories: content-based, collaborative filtering, context-aware, and hybrid recommender systems. In Content-Based RS [13] products are recommended based on the similar products the user has consumed in the past. In Collaborative Filtering (*CF*) RS, products are recommended based on the ratings of similar users. The CF algorithms [11, 21]can either be memory-based or model-based. In a memory-based approach, recommendations are generated based on the ratings of the k nearest neighbor (most similar users), while for the model-based approach, a model is developed based on the user ratings and then used to predict the preferences of the users. Contextual information such as location and time, are leveraged to recommend products in the Context-Aware RS [1]. Finally, the Hybrid RS techniques [6] combine two or more recommendation approaches into one.

In several real-life scenarios, the recommendation process involves more than a person, and we need to recommend products or services for a group of individuals. For example, recommending an event to a group of friends or recommending a good restaurant to a group of colleagues. Unlike individual recommendations, the group recommendations need to combine the preferences of each individual in the group to generate suitable recommendations. Thus, it can be more challenging than recommend products or services to a group due to the diversity of interests among the group members.

Various aggregation strategies exist to help aggregate recommendations for a group of individuals based on the Social Choice Theory. Broadly, Group Recommender Systems (GRS) follow either of the two basic aggregation strategies [14, 15]:

- Aggregated Predictions: where the recommendation step precedes the aggregation step, or
- Aggregated Models: where the aggregation step precedes the recommendation step.

In the first approach, we first recommended items to the individual members, and then aggregate the individual recommendations to generate group recommendations. In the second approach, we generate a group profile from the preferences of the individual members and use it to generate recommendations for the group. We compare the two approaches in Fig. 1. Some of the works [5] have also compared the predictive quality of these two approaches.

Several aggregation functions have been proposed over the years as there exists no optimal function to aggregate preferences of the individuals in a group. These functions can be broadly classified [15] in the following three categories:

- Majority-Based: focuses on those items which are the most popular (e.g., Plurality Voting, Borda Count, Copeland Rule)
- Consensus-Based: takes into account preferences of all the individuals in a group (e.g., Average, Average without Misery, Additive Utilitarian)
- Borderline: considers a subset of the user preferences (e.g., Majority Voting, Least Misery, Most Pleasure)

Various platforms such as Yelp and Foursquare recommend local search service and Point-of-Interest (POI) by leveraging the spatial and social information of the users. POI recommendation, i.e., recommending for users unvisited POIs (e.g., restaurants, bars, shopping malls, and events) based on users' check-in records and

*Submitted in partial fulfillment of the requirement for module IN2107

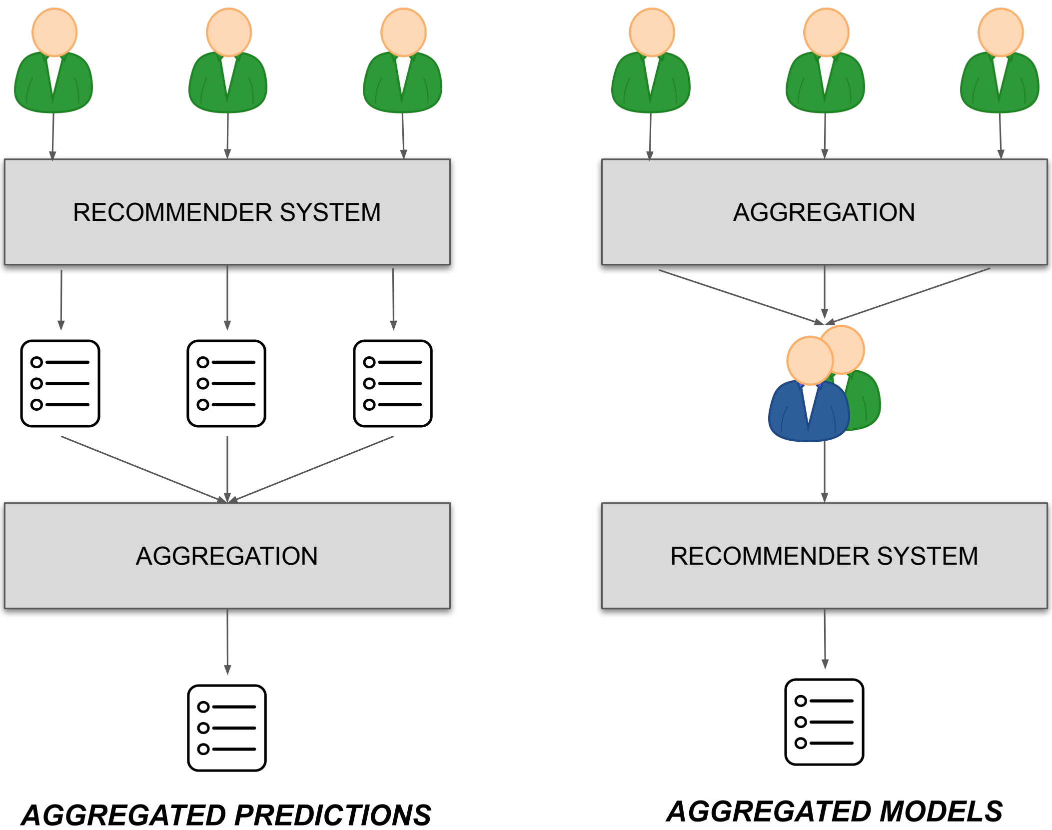

Figure 1: Aggregation Strategies for Group Recommender Systems

ratings have gained much interest in recent years. It is different from the traditional recommendation of items like books and movies as the geographical location of both the user and item plays an essential role.

Group Recommendations in the domain of POI aims to suggest places like restaurants and malls for a group of individuals. Most conventional Conventional GRS for POI often do not take into account the geographical location of all the individuals in a group when generating recommendations. The geographical location of the individuals in a group plays a vital role as people prefer eating, shopping, or watching movies in the regions which are near to their home, workplace, and other relevant areas. Often a considerable amount of time is spent deciding an optimal location to meet for all the individuals in a group.

In this paper, we took the restaurant recommendation as an example and proposed a personalized POI recommender system for groups. We put a particular emphasis on the geographical location of a group of individuals to generate recommendations only in the areas that the individuals of the group frequent the most. We leverage the geographic information of the individuals in the group both before and after generating recommendations. It helps us filter out the POIs which are not located in the vicinity of the individuals in a group.

In this work, first, we discuss some of the existing frameworks for Group Recommender Systems. Later, we present our framework where the geographical location of individuals in a group plays an essential role when generating recommendations for the group.

2 RELATED WORK

2.1 Group Recommender System

There exist a variety of review articles that discuss the various aspects of the group recommendation systems [8, 10]. In this section, we discuss some of the state-of-the-art approaches for the group recommendation systems in brief. POLYLENS [17] aims to recommend movies to small groups. It merges the recommendations generated for individual users by nearest neighbor methods and uses Least Misery Strategy as the aggregation function.

The Pocket Restaurant Finder [16] recommends restaurants to a group. Each user in the group has to rate their preferences for cuisine type, restaurant amenities, cost, and distance on a 5-point rating scale. The individual preferences are then aggregated, and the group is presented with a list of the potential restaurants they can visit together.

INTRIGUE [2] is a GRS for tourism. The demographic information of the original group is used to generate several subgroups. Later the preferences of each sub-group are aggregated using the weighted form of the Average strategy to generate recommendations for the group.

A group recommender system that generates a recommendation based on the individuals' preferences for features of the TV program was proposed in [22]. It uses a variant of the Average Strategy aggregation function and assigns a score of -1 (dislikes), +1 (likes), or 0 (neutral) to each selected feature based on the individuals' rating for a program. The feature vector for the group is computed, such that it minimizes its distance compared to individuals' feature vectors. This is similar to calculating the average rating per feature.

GROUPFUN [18] recommends music to a group of individuals attending an event together. The users are first asked to create their playlist, and then the group is formed. The recommendations are then generated based on the probabilistic voting scheme, where the probability of each music is computed from its global popularity.

2.2 POI Group Recommender Systems

Due to the increasing availability of user preference and behavioral data, several recommender systems have been proposed recently to suggest POIs to a group of users. The behavior of the group has been studied in [19] to recommend POIs to groups in Location-Based Social Networks. It proposes a hierarchical Bayesian model that learns the group and location activities. It then generates group recommendations using matrix factorization in a collaborative filtering framework.

A Geo Group Recommender (GGR) presented in [3] recommends locations to a group of users in the areas with the most frequent group presence. It combines the group geographical preferences, category, and location features, and group check-ins to show that recommending POIs near the areas were groups move is feasible, and training a model using group profiles performs better than combining individual recommendations.

A Context-Aware GRS has been proposed in [23] take into account the importance of location and the intragroup influence in POI group recommendation. They employ distance prefiltering and distance ranking adjustment by considering the impact of contextual factors on the user's ranking deviation and modeling them using neural networks.

Most of the work in the domain on POI Group Recommender Systems is based on check-ins made by the users. The data about these check-ins might be sparse as most people do not often check-in when they visit a particular POI. Thus, it can prove to difficult to design a reliable recommendation framework that generates suitable POI recommendations based on these sparse check-in data. An alternate solution is to use the ratings given by the users instead of the check-ins. Therefore, in this paper, we propose a GRS framework based on the geo-location and ratings given by the user.

3 OUR PROPOSED METHOD

In this section, we present our proposed framework for the POI group recommendation. In the first step, we identify the areas where each user frequents the most by computing the Top-K individual clusters for each user. In the second step, we identify the areas where each group frequents the most based on the results of the previous step. To achieve this, we compute the Top-M clusters for each group using the Top-K individual clusters of the group members. It helps us filter out the areas that are most accessible for each group. Finally, we generate a group profile for each group and use it to generate a ranked list of POIs. The outline of our framework is presented in Fig. 2.

3.1 Top-K Individual Clusters

In this step, we aim to identify the areas that the given user frequents the most by computing the Top-K clusters for each user. For this, we leverage the rating history of the user. Based on the user's rating history, we retrieve the list of POIs he has visited in the past. We cluster the POIs based on its geolocation using the well-known clustering technique, DBSCAN [20], with Haversine distance as the metric.

The DBSCAN is scalable and fast for low dimensional data like geolocation, which has only two dimensions, i.e., latitude and longtime. Additionally, it can accurately detect outliers as well in the data. We modify the DBSCAN algorithm to allow clusters to have a maximum length of L_{max}. It allows us to prevent the DBSCAN from forming large clusters in high-density areas and, at the same time, get a fully deterministic result. We compute the convex hull of all the points in the cluster and then calculate the length of the resulting polygon to get the length of each individual cluster, L_{IC}.

For each user, U in the data set, we do the following to compute its Top-K clusters:

- Compute clusters using DBCSAN, as explained above.
- Calculate the score for each cluster by counting the number of POIs (points) in a given cluster divided by the total number of POIs he has visited in the past. If two clusters have the same score, the one with a higher frequency (number of POI visits) and a higher average rating is chosen.
- Select the clusters that have the K highest score as the Top-K clusters for the user U. We represent the Top-K clusters, $IC_0, IC_1 ... IC_K$ of each user U by its geographical center for convenience.

Figure 3 shows the Top-3 Individual Clusters for an individual sample user who has visited various restaurants in Phoenix. The highlighted areas depict the areas the users frequents the most.

3.2 Top-M Group Clusters

For a given group, G, having N users $U_1, U_2, ..., U_N$, we aim to find the Top-M clusters for the group, which represents the most common areas that are frequently visited by the individuals in the group. We use the Top-K clusters of the individual users computed in Section 3.1 to find the Top-M for the groups. To accomplish this, we use the modified DBSCAN algorithm presented in Section 3.1.

For each individual user in the group, we have a maximum of K Individual Clusters. Thus a group G consisting of N users, $U_1, U_2, ..., U_N$, will have a maximum of $N \times K$ individual clusters from which the Top-M Group Clusters are computed. We use the modified DBSCAN algorithm to first combine the clusters that are close to each other, given that the combined cluster's total length is less than L_{max}. We then calculate the score for each of the remaining clusters by counting the number of unique users in a given cluster divided by the group's size, N. If two clusters have the same score, the one with a higher frequency (number of POI visits) and a higher average rating is ranked higher than the other. We select the clusters that have the M highest-ranked clusters as the Top-M Group Clusters for the given group G.

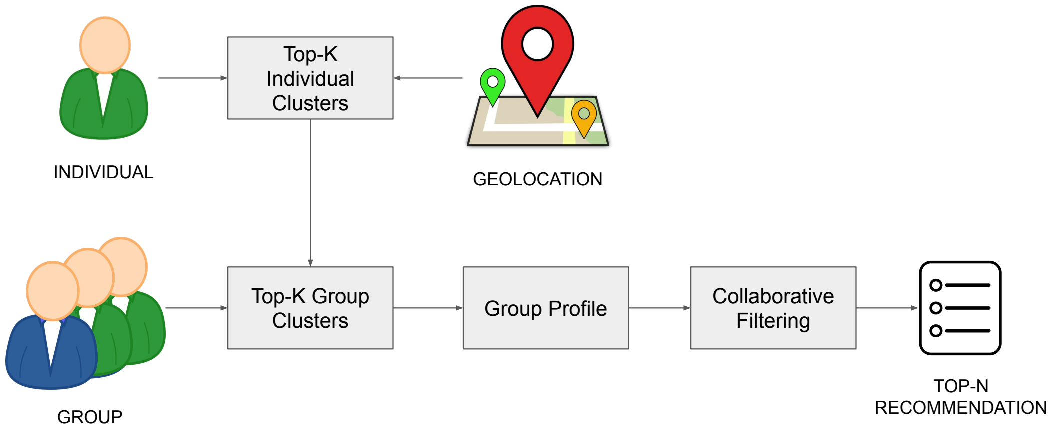

Figure 2: Outline of our Framework

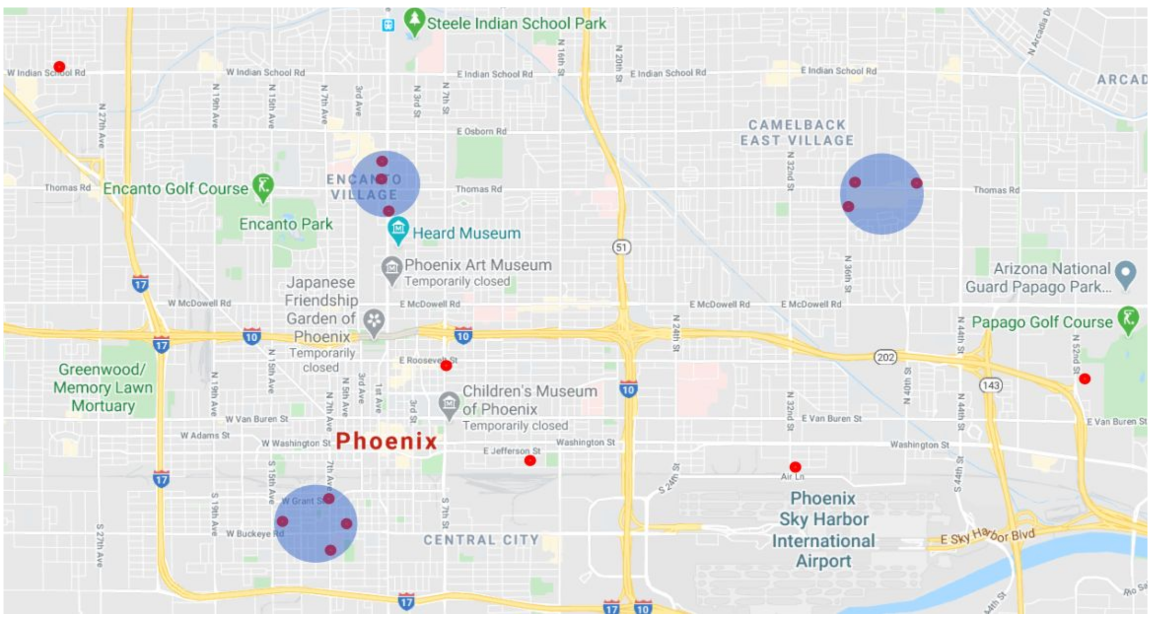

Figure 3: Example for Top-K Individual Clusters

Figure 4 shows the Top-3 Individual Clusters for three individual users, each represented by a different color, who have visited various restaurants in Phoenix. The Top-3 Group Clusters encircled in blue represent the most common areas the groups frequents the most.

3.3 Group Recommendation

Once we have identified the areas a particular group frequents the most, we aggregate the ratings of individual users into a group profile. For each group, we only consider the ratings for POIs located in the Top-M Group Clusters of the given group. It helps us filter out POIs that are too far away for the individuals in the group. We use Consensus-Based aggregation functions such as Average (AVG) to aggregate the individuals' ratings in the group. Finally, we apply Collaborative Filtering (CF) to group profiles to generate a list of suitable recommendations for each group. We again filter out the POIs that are not located in the Top-M Group Clusters of the given group from the list to get the Top-N recommendations for each group.

4 EXPERIMENTAL SETTINGS

In this section, we discuss the setup for evaluating our proposed framework in detail. In Section 4.1, we discuss in detail the data set we have used. In Section 4.2, we present the details about the synthetic groups generated for our experiments while in Section 4.3 4.3, we explain the different metrics we use to evaluate the performance of our recommender system.

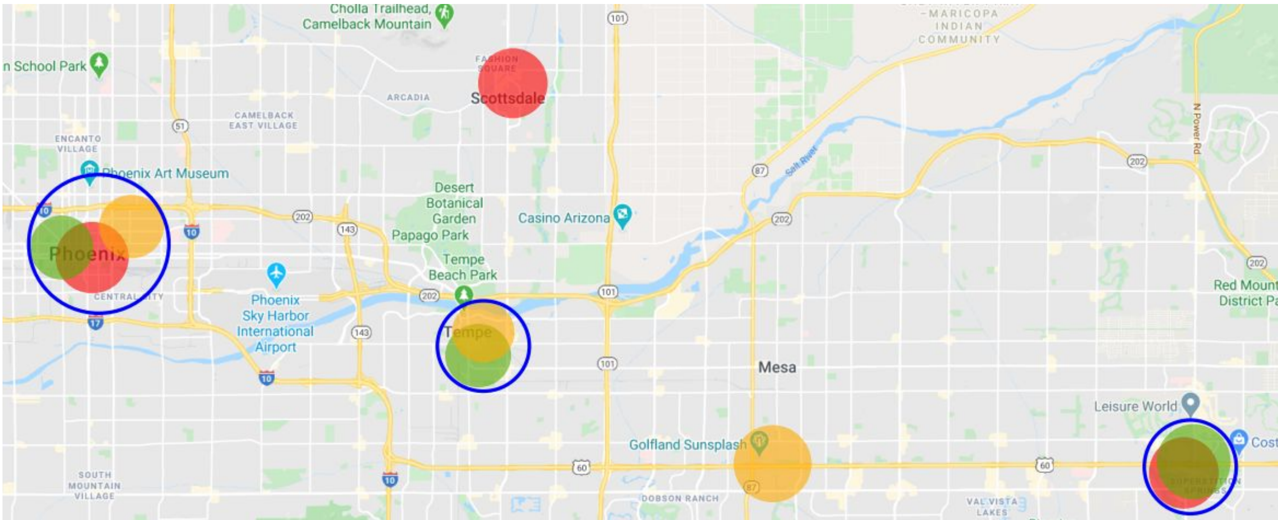

Figure 4: Example for Top-M Group Clusters

4.1 Datasets

Since there are no datasets publicly available that contains both individual and group ratings, evaluation of a group recommender system can be more challenging than conventional recommendation systems. For our experiments, we use the dataset from Yelp's Dataset Challenge[1] to test the performance of our proposed framework. The original dataset contains about 6.7 million ratings given by 1.6 million users for about 200 thousand businesses spread across 10 metropolitan areas. It also consists of information about aggregated check-ins and tips. It contains 1.4 million business attributes like hours, parking, availability, and ambiance.

For our experiments, we test the proposed framework in the domain of restaurants. In the Yelp dataset, businesses are not explicitly classified as restaurants. Therefore we leverage the tags associated with each business to identify the restaurants. In our pruned dataset, about 3.9 million ratings were given by about 1 million users for approximately 55 thousand restaurants. We further prune the dataset by only considering the restaurants in Phoenix, which has the highest number of ratings in the dataset among the 10 metropolitan areas. Additionally, to filter out the inactive users, we only consider users who have 10 or more ratings. Our final dataset consists of 5.6K users who have given 120K ratings for 3.5K businesses.

4.2 Group Formation

The Yelp dataset does not have any explicit information about groups. We compute synthetic groups by leveraging the information about social networks present in the dataset. We first build an undirected graph of G to represent all the users and their social networks for the given city. We add an edge e_{uv} to this graph G for every two users u and v who are friends on the Yelp platform. Then we compute cliques of varying sizes in the graph and use it to form groups of required sizes.

[1]https://www.yelp.com/dataset

We form a total of 1500 synthetic groups of varying sizes for our experiments. The users in each of these groups have rated at least 4 or more restaurants together. We calculate the similarity of a group using the Cosine Similarity measure. We report the size and similarity distribution of the groups in 5. The mean similarity of the groups in our data set is 0.223, with a standard deviation of 0.041. It can be challenging to generate groups with very high similarities as often, individuals in a group have different preferences.

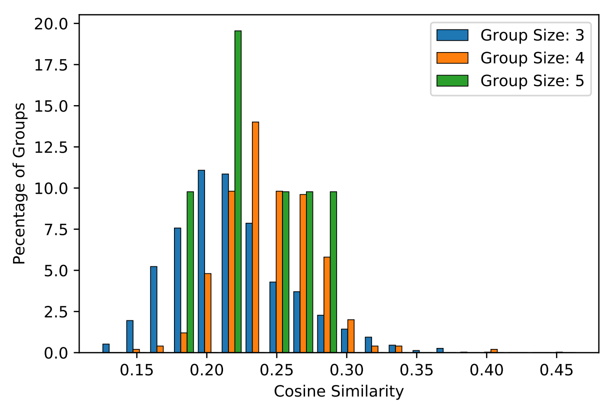

Figure 5: Similarity Distribution for Groups

4.3 Evaluation

To evaluate the performance of our proposed GRS, we follow the same procedure as proposed in [9]. For each group, we randomly withheld the rating of an item (restaurant) that all the individuals have rated together. These ratings are then used as the test set to evaluate the performance. The remaining ratings are used as the training set for the group recommendation algorithm to generate top-N recommendations for each group. We employ the metric Hit

Rate proposed in [7] to evaluate the quality of the top-N recommendations for each group. The number of hits is the number of items in the test set that were also present in the top-N recommended items returned for each group.

$$Hit\ Rate(HR) = \frac{Number\ of\ Hits}{Total\ Number\ of\ Groups}$$

An HR value of 1 indicates that the algorithm was always able to recommend the items from the test set, whereas an HR value of 0.0 indicates that the algorithm could not recommend any of the items from the test set.

To analyze where each hit occurred in the top-N list, we use Average Reciprocal Hit-Rank (ARHR). If h is the number of hits that occurred at positions $p_1, p_2, ..., p_h$ within the top-N lists (i.e., 1 $\leq p_i \leq$ N), then the average reciprocal hit-rank is equal to:

$$ARHR = \frac{1}{Total\ Number\ of\ Groups} \sum_{i=1}^{h} \frac{1}{p_i}$$

4.4 Comparison

In this work, we use the model-based Collaborative Filtering technique Singular Value Decomposition (SVD)[12]. We combine it with our framework to generate suitable recommendations for each group. Thus, our approach is the combination of SVD and the average aggregation strategy, referred to as Clustering GRS.

A baseline version of the algorithm, Baseline GRS, is used to compare the performance of the proposed approach. In Baseline GRS, we first average the ratings to aggregate the preferences of all the users in the group and then use the SVD algorithm to generate recommendations for each group. Unlike our proposed approach, we do not perform any pre-filtering or post-filtering based on geolocation for the Baseline GRS.

5 RESULTS AND DISCUSSION

In this section, we examine the performance of our group recommendation approach. First, we analyze in detail the Group Clusters generated in Section 3.1 & 3.2. Later, we analyze the performance of our group recommendation approach and then discuss the overall results. In the proposed Clustering approach, we first generate Top-K Individual Clusters for each user in the dataset and then use these clusters to generate Top-M Group Clusters, as discussed in Section 3.

In Table 1, we report the mean radius and standard deviation of the Group Clusters by varying the values of K and M, respectively. We keep the value of L_{max} constant as 2 kilometers for all our experiments. We can notice that the radius of Group Clusters is close to the value of L_{max}, which indicates that our proposed approach does not allow the formation of large clusters. We can also observe that higher values of K and M result in Group Clusters with a smaller radius.

We report and compare the performance values of the different approaches in Fig. 6 & 7. We measure the different HR and ARHR values for Top-3, Top-5, Top-10, and Top-20 recommendations by varying the values of K and M from 3 to 7. From Fig. 6 & 7, we can conclude that our proposed approach, Cluster GRS, outperforms Baseline GRS both in terms of HR and ARHR for all values of N. Additionally, for $K = 3$ & $M = 5$, we get the best results for our Cluster GRS approach irrespective of the evaluation metric. Thus, we can conclude that our proposed geolocation-based GRS approach helped us achieve more suitable recommendations than the traditional approach.

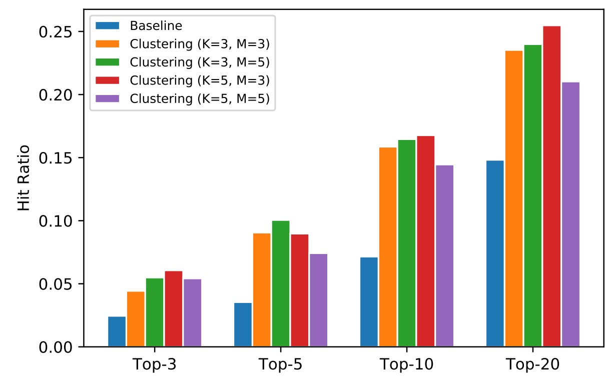

Figure 6: Comparison of HR values

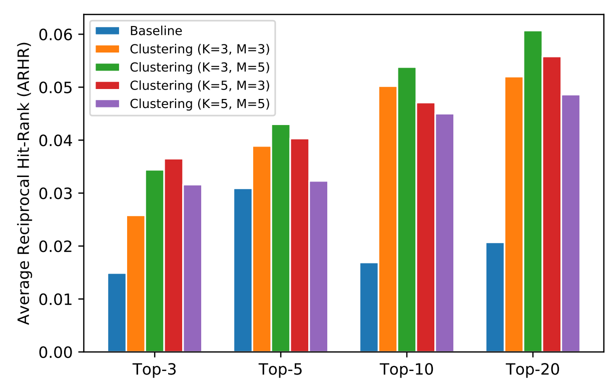

Figure 7: Comparison of ARHR values

6 CONCLUSION

Group Recommender systems have been an active research topic for the past decade. In this paper, we discussed some of the limitations of the existing group recommendations for POIs. We presented a geolocation-based approach to improve the quality of recommendations. Experimental results on the Yelp dataset show that our approach helps generate more suitable recommendations for the group. In the future, we would like to test our approach in other local-search services domains and study the influence of size and similarity of the groups. We also aim to improve the recommender system's performance further by integrating the social network information of the users.

REFERENCES

[1] Gediminas Adomavicius and Alexander Tuzhilin. 2015. *Context-Aware Recommender Systems.* Springer US, 191–226. https://doi.org/10.1007/978-1-4899-7637-6_6

K	M	Mean Radius (in kms)	Standard Deviation Radius (in kms)
3	*3*	1.8815	0.5093
3	*5*	1.8755	0.514
5	*3*	1.5987	0.551
5	*5*	1.5468	0.5677

Table 1: Analysis of Group Clusters

[2] Liliana Ardissono, Anna Goy, Giovanna Petrone, Marino Segnan, and Pietro Torasso. 2003. Intrigue: Personalized recommendation of tourist attractions for desktop and hand held devices. *Applied Artificial Intelligence* 17, 8-9 (2003), 687–714. https://doi.org/10.1080/713827254

[3] Frederick Ayala-Gómez, Bálint Daróczy, Michael Mathioudakis, András Benczúr, and Aristides Gionis. 2017. Where Could We Go? Recommendations for Groups in Location-Based Social Networks. In *Proceedings of the 2017 ACM on Web Science Conference*. Association for Computing Machinery, New York, NY, USA, 93–102. https://doi.org/10.1145/3091478.3091485

[4] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez. 2013. Recommender systems survey. *Knowledge-Based Systems* 46 (2013), 109 – 132. https://doi.org/10.1016/j.knosys.2013.03.012

[5] Ludovico Boratto and Salvatore Carta. 2015. The Rating Prediction Task in a Group Recommender System That Automatically Detects Groups: Architectures, Algorithms, and Performance Evaluation. *Journal of Intelligent Information System* 45, 2 (2015), 221–245. https://doi.org/10.1007/s10844-014-0346-z

[6] Erion Çano and Maurizio Morisio. 2017. Hybrid recommender systems: A systematic literature review. *Intelligent Data Analysis* 21, 6 (2017), 1487–1524. https://doi.org/10.3233/IDA-163209

[7] Mukund Deshpande and George Karypis. 2004. Item-Based Top-N Recommendation Algorithms. *ACM Transactions on Information Systems* 22, 1 (Jan. 2004), 143–177. https://doi.org/10.1145/963770.963776

[8] Anthony Jameson and Barry Smyth. 2007. *Recommendation to Groups.* Springer Berlin Heidelberg, 596–627. https://doi.org/10.1007/978-3-540-72079-9_20

[9] Heung-Nam Kim, Majdi Rawashdeh, and Abdulmotaleb El Saddik. 2013. Tailoring Recommendations to Groups of Users: A Graph Walk-Based Approach. In *Proceedings of the 2013 International Conference on Intelligent User Interfaces*. Association for Computing Machinery, New York, NY, USA, 15–24. https://doi.org/10.1145/2449396.2449401

[10] Michal Kompan and Maria Bielikova. 2014. Group recommendations: Survey and perspectives. *Computing and Informatics* 33, 2 (2014), 446–476.

[11] Yehuda Koren and Robert Bell. 2015. *Advances in Collaborative Filtering*. Springer US, 77–118. https://doi.org/10.1007/978-1-4899-7637-6_3

[12] Y. Koren, R. Bell, and C. Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42, 8 (2009), 30–37.

[13] Pasquale Lops, Marco de Gemmis, and Giovanni Semeraro. 2011. *Content-based Recommender Systems: State of the Art and Trends.* Springer US, 73–105. https://doi.org/10.1007/978-0-387-85820-3_3

[14] Judith Masthoff. 2011. *Group Recommender Systems: Combining Individual Models.* Springer US, 677–702. https://doi.org/10.1007/978-0-387-85820-3_21

[15] Judith Masthoff. 2015. *Group Recommender Systems: Aggregation, Satisfaction and Group Attributes.* Springer US, 743–776. https://doi.org/10.1007/978-1-4899-7637-6_22

[16] Joseph F McCarthy. 2002. Pocket restaurantfinder: A situated recommender system for groups. In *Workshop on Mobile Ad-Hoc Communication at the 2002 ACM Conference on Human Factors in Computer Systems*, Vol. 8.

[17] Mark O'Connor, Dan Cosley, Joseph A. Konstan, and John Riedl. 2001. *PolyLens: A Recommender System for Groups of Users.* Springer Netherlands, 199–218. https://doi.org/10.1007/0-306-48019-0_11

[18] George Popescu and Pearl Pu. 2012. What's the Best Music You Have? Designing Music Recommendation for Group Enjoyment in Groupfun. In *CHI '12 Extended Abstracts on Human Factors in Computing Systems.* Association for Computing Machinery, 1673–1678. https://doi.org/10.1145/2212776.2223691

[19] Sanjay Purushotham, C.-C. Jay Kuo, Junaith Shahabdeen, and Lama Nachman. 2014. Collaborative Group-Activity Recommendation in Location-Based Social Networks. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Crowdsourced and Volunteered Geographic Information.* Association for Computing Machinery, New York, NY, USA, 8–15. https://doi.org/10.1145/2676440.2676442

[20] Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, and Xiaowei Xu. 2017. DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN. *ACM Transactions Database Systems* 42, 3, Article 19 (July 2017), 21 pages. https://doi.org/10.1145/3068335

[21] Yue Shi, Martha Larson, and Alan Hanjalic. 2014. Collaborative Filtering beyond the User-Item Matrix: A Survey of the State of the Art and Future Challenges. *ACM Computer Survey* 47, 1, Article 3 (May 2014), 45 pages. https://doi.org/10.1145/2556270

[22] Zhiwen Yu, Xingshe Zhou, Yanbin Hao, and Jianhua Gu. 2006. TV Program Recommendation for Multiple Viewers Based on User Profile Merging. *User Modeling and User-Adapted Interaction* 16, 1 (2006), 63–82. https://doi.org/10.1007/s11257-006-9005-6

[23] Q. Zhu, S. Wang, B. Cheng, Q. Sun, F. Yang, and R. N. Chang. 2018. Context-Aware Group Recommendation for Point-of-Interests. *IEEE Access* 6 (2018), 12129–12144.