Where Could We Go? Recommendations for Groups in Location-Based Social Networks

Frederick Ayala-Gómez
Eötvös Loránd University,
Faculty of Informatics
Budapest, Hungary
fayala@caesar.elte.hu

Bálint Daróczy
Inst. Computer Science and Control,
Hungarian Academy of Sciences
(MTA SZTAKI)
Budapest, Hungary
daroczyb@ilab.sztaki.hu

Michael Mathioudakis
Computer Science Department
Aalto University
Espoo, Finland
michael.mathioudakis@aalto.fi

András Benczúr
Inst. Computer Science and Control,
Hungarian Academy of Sciences
(MTA SZTAKI)
Budapest, Hungary
benczur@sztaki.mta.hu

Aristides Gionis
Computer Science Department
Aalto University
Espoo, Finland
aristides.gionis@aalto.fi

ABSTRACT

Location-Based Social Networks (LBSNs) enable their users to share with their friends the places they go to and whom they go with. Additionally, they provide users with recommendations for Points of Interest (POI) they have not visited before. This functionality is of great importance for users of LBSNs, as it allows them to discover interesting places in populous cities that are not easy to explore. For this reason, previous research has focused on providing recommendations to LBSN users. Nevertheless, while most existing work focuses on recommendations for individual users, techniques to provide recommendations to groups of users are scarce.

In this paper, we consider the problem of recommending a list of POIs to a group of users in the areas that the group frequents. Our data consist of activity on Swarm, a social networking app by Foursquare, and our results demonstrate that our proposed *Geo-Group-Recommender* (GGR), a class of hybrid recommender systems that combine the group geographical preferences using Kernel Density Estimation, category and location features and group check-ins outperform a large number of other recommender systems. Moreover, we find evidence that user preferences differ both in venue category and in location between individual and group activities. We also show that combining individual recommendations using group aggregation strategies is not as good as building a profile for a group. Our experiments show that (GGR) outperforms the baselines in terms of precision and recall at different cutoffs.

CCS CONCEPTS

•Information systems →Location based services; Personalization; Recommender systems;

KEYWORDS

Group Recommendation; Location-Based Social Networks; Recommender Systems

ACM Reference format:
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 WebSci '17, Troy, NY, USA, June 25-28, 2017,* 10 pages.
DOI: http://dx.doi.org/10.1145/3091478.3091485

1 INTRODUCTION

Location-Based Social Networks (LBSNs) are platforms that enable people to share online their whereabouts (the places they visit and whom they visit them with) – and, in turn, learn the whereabouts of their online friends. This is achieved via *check-ins*, i.e., posts that contain the location (latitude, longitude) of a user and the exact venue, e.g., a restaurant. Using this information users get to know where their friends are. Additionally, check-ins create a timeline of the places that users have visited. Utilizing check-in information, LBSNs recommend venues as Point of Interests (POIs) that users might like to visit.

Recommendations for new places to visit are of major importance for users of LBSNs. For example, in metropolitan areas or while on holidays, users often wish to discover new places that they would be interested in – yet such information is often not readily available. Note that the task of recommending new POIs is different than that of recommending other types of items (e.g., movies, news) in that geography also comes into play. Recalling Tobler's first law of geography: "everything is related to everything else, but near things are more related than distant things" [26].

In this work, we focus on a particular variant of the recommendation task: one that seeks to *recommend a new POI to a group of users*. To see why this task deserves particular attention, consider that, when users choose a venue to visit with somebody else (e.g., friends or family), their venue of choice can generally be different than if they do so alone. For example, consider a person that is a big fan of hamburgers – but hangs out with friends who prefer sushi.

Publication rights licensed to ACM. ACM acknowledges that this contribution was authored or co-authored by an employee, contractor or affiliate of a national government. As such, the Government retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for Government purposes only.
WebSci '17, Troy, NY, USA
© 2017 Copyright held by the owner/author(s). Publication rights licensed to ACM.
978-1-4503-4896-6/17/06…$15.00
DOI: http://dx.doi.org/10.1145/3091478.3091485

Such situations of conflicting tastes and interests pose a challenge for the recommendation task: what POIs to recommend for a group of users if the individual preferences differ? Moreover, note that we restrict ourselves in recommending *new POIs* (i.e., ones that the group has not visited in the past), as such recommendations are of most practical interest (compared to recommending POIs that the group has already visited) and most commonly deployed on real-world LBSNs (like Foursquare).

The problem of group recommendations has been studied before, but in different settings. For instance, there is work on recommending relevant music [18], movies [21], holidays [19], news [9]. In the setting of LBSNs, on the other hand, most earlier work has focused on recommendations for individual users [4].

To address the gap in the literature, our work addresses the following research questions.

RQ1: How do groups behave in LBSNs?
RQ2: How do preferences change when users are alone vs. when they are in a group?
RQ3: How to recommend items in the areas that a group frequents?

For all questions above, our analysis is based on a new dataset from Swarm, a LBSN developed by Foursquare. The data cover activity in three major cities: Istanbul, Izmir and Mexico City. The code used for data collection, analysis, experimentation, and the dataset are available for academic purposes[1].

With respect to **RQ3**, the use case scenario is that of a group of users who plan to meet and look for recommendations for a new place to try: in a first step, they are prompted by the system to select an area among the ones they have been to in the past; in a second step, the system provides them with recommendations for the selected area. The techniques we study implement the latter (second) step. For all techniques, individual and group preferences are assumed known and a single venue is recommended. Moreover, the group is passive towards the provided recommendation – i.e., there is no interaction between the users and the recommender system to shape the recommendation (e.g., via voting and a concensus mechanism).

We experiment with a large number of techniques drawn from the literature; and present *Geo-Group-Recommender* (GGR), a hybrid recommender system that combines collaborative and content filtering together with a geographical Kernel Density Estimation. Our results show that the proposed recommender system outperforms existing systems and other baselines.

2 BACKGROUND AND PRIOR WORK

The problem of recommending venues for individual users in LBSNs has been widely studied. A recent survey can be found in [4]. State of the art models like Fused Matrix Factorization Framework with the Multi-center Gaussian Model (FMFMGM) [7] and GeoSoCa [27] exploit geographical and social information of users. The idea of including the location preferences in the collaborative filtering learning is presented in GeoMF [15]. Research on recommending venues to groups is still scarce but emerging and promising.

2.1 Group recommendations

Recommender systems for groups are surveyed in [17]. The authors highlight that the use case of the recommender system greatly affects the design. They characterize group recommender systems by considering the following dimensions and we highlight in bold those that apply to our case: (*i*) **individual preferences are known** vs. developed over time; (*ii*) recommended items are experienced by the group vs. **presented as options in a list**; (*iii*) **the group is passive** (e.g., users are not voting) vs. active (e.g., the system helps create consensus) and (*iv*) **recommending a single item** vs. a set.

A summary of different strategies to combine individual preferences to generate group recommendations can be found in [17]. A brief summary of the methods is the following: Average Individual Ratings (AIR) considers the average rating of each item; Average Without Misery (AWM) assigns to items the average of their individual ratings under a certain threshold; Least Misery (LM) considers the minimum of their individual ratings. The authors also present more elaborate methods like graph-based ranking [13], Spearman footrule rank [2], Nash equilibrium [6] and purity and completeness [25].

2.2 Groups in LSBNs

The behavior of groups in LBSNs has been researched for different tasks. For instance, [16] studies companion recommendations where the task is to find friends interested in joining certain POI. [1] focus on recommending an itinerary for touristic groups visiting a city. However, there is scarce research work specialized in POIs recommendation for groups.

Comparing users and groups behavior in LSBN is investigated in [5] using data from Foursquare (i.e. Swarm) and Telecommunications networks. The authors show that the category of the venue and location affects the propensity for groups to meet and discuss that this behavior could affect the POIs recommendation task. Our work is complementary in the following aspects. We analyze check-ins that explicitly mention friends that are together instead of co-located within an hour. We study the behavior of Swarm users from other cities than New York. To measure the category preferences of users and groups, we use Kendall-tau as a ranking correlation metric. Our behavior analysis includes time and distance between check-ins. We use all the POIs check-ins by using clusting with DBSCAN instead of just the top POIs for users and groups. Finally, the authors in [5] did not research the performance of recommender systems for groups in LBSN.

The authors in [23] study group behavior and recommending POIs to groups in LBSNs. To detect the groups, they identify the connected components based on time, location and friends network of the Gowalla dataset. A major drawback of the dataset is that it lacks information about the location itself (e.g., category, popularity). The Gowalla check-ins contains just the latitude, longitude and ID for the POI. To overcome this, the authors retrieve POI around the latitude and longitude using the Foursquare API and then aggregate the categories. Also, check-ins are spread around the world and the authors do not mention any geographical scope limitation for their experiments.

[1] https://github.com/frederickayala/lbsn_group_recsys

Their model, called Collaborative Group Activity Recommender (CGAR), represents group and location activities as topic models that are combined using collaborative filtering techniques. Their model includes latent variables for activity preference and community influence, that express whether an activity at a location is more interesting for one group than to another, as well as how user communities influence the preference of locations. They highlight that preferences between users and groups differ and show that their model personalizes category preferences better than regular strategies to combine individual recommendations (i.e., aggregating by average). Their model outperforms baselines (i.e., CTR, MF) in Mean Recall@K(50-1000), Mean Rating Prediction Accuracy and Mean Root Mean Squared Error.

The main differences between [23] and our work are the following. First, we use a dataset collected from Swarm that does not require any additional technique for detecting groups. Also, our collected dataset contains information about the POI so there is no need to crawl for the venues information. To improve the quality of our results, we include a cleaning step to remove bots and very active users. We present a more comprehensive analysis that highlights not only the category preferences but also the location and time preferences. Recommendations are usually presented in the shape of a ranked lists with few POIs. This is why we evaluate our recommender on the TOP K recommendation setting where K is in the range of [5,50] instead of [50,1000]. Another difference is that we focus on recommending items near the areas where the group check-ins are more concentrated and for three major cities (i.e., Istanbul, Izmir and Mexico City). We tried different recommender systems to generate a ranked list of possible POIs. Finally, we fit a Kernel Density Estimation with Gaussian Kernel per group to prioritize the POIs near the area of recommendation.

Table 1 highlights the differences of *Geo-Group-Recommender* (GGR – our model) in comparison to CGAR and recent LBSN recommender systems for individual users.

Model	Categories	Geography	Social	Group	TOP K @	Prioritize POIs
GGR (Ours)	Yes	Yes	No	Yes	5-50	Group Geo Density
CGAR [23]	Yes	No	Yes	Yes	50-1000	No
GeoMF [15]	No	Yes	No	No	5-100	User Geo Density, POI influence
GeoSoCa [27]	Yes	Yes	Yes	No	2-50	User Social Network
FMFMGM [7]	No	Yes	Yes	No	5-10	No

Table 1: Model Comparison

2.3 Recommender systems

A recent survey on recommender systems can be found in [24]. We are interested in methods that can be used when the user-item consumption lacks explicit ratings (e.g. 1-5 stars). For this purpose, we use implicit matrix factorization with two optimization methods (i.e. Implicit Alternating Least Squares (iALS) [22] and Stochastic Gradient Descent (SGD) for collaborative filtering [14]. We use other models that learn item and user similarities based on a distance metric and Nearest Neighbour methods [14], as well as ones based on item popularity.

2.4 Kernel Density Estimation

As mentioned in [4] and [27], Kernel Density Estimation (KDE) is used in several LBSNs recommender systems. KDE is calculated using the equation

$$f(x) = \sum_{i=1}^{n} K(x, x_i; h), \tag{1}$$

where X is a set that contains samples $x_1, x_2, \ldots, x_n$ from the corresponding probability distribution. K is the kernel and h is the smoothing parameter called bandwidth. In our experiments, K is the Gaussian Kernel

$$K(x, x_i; h) \sim \exp\left(-\frac{(x - x_i)^2}{2h^2}\right). \tag{2}$$

3 DATA

We require a dataset with the LBSN activity of users both when they are alone and when they are in a group. The datasets used in [3], [8], [11], [5] and [20] contain information about the activity of individual users and their friends, but no group information – and some lack detailed information about the venue.

To create such a dataset, we collect data related to the popular LBSN Swarm, a platform that enables users to indicate the venue they are *checking-in*. On Swarm, users are able to mention with whom they visit a venue and share publicly their check-ins in other social networks, like Twitter. This gives us the opportunity to collect data related to both individual and group activity.[2]

Towards that end, we deploy a crawler that uses the Twitter API to search for public tweets that contain group check-ins. Then, in snowballing fashion, we collect the latest 200 tweets of each user that is mentioned in a group check-in and extract their public check-ins contained therein. Figure 1 is a visualization of this recursive process.

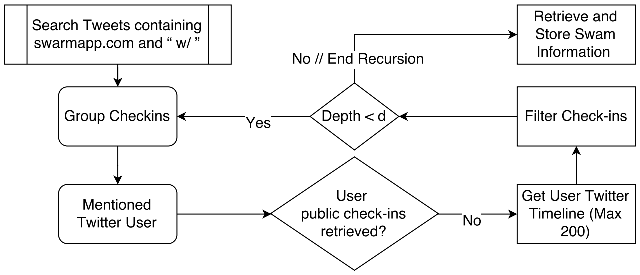

Search Tweets containing swarmapp.com and " w/ "
Retrieve and Store Swam Information
No // End Recursion
Group Checkins
Yes
Depth < d
Filter Check-ins
Mentioned Twitter User
User public check-ins retrieved?
No
Get User Twitter Timeline (Max 200)

Figure 1: The crawling process of Swarm check-ins. Additionally, the lookup endpoint of the Twitter API allow us to constrain the search to a specific location by defining the geographical center and a radius.

Our recursive crawl is constrained by a stopping condition that specifies the maximum depth d the crawler can reach from the original group check-in. Depth 0 corresponds to the check-ins retrieved in the first pass over tweets, depth 1 to the check-ins of the users who are mentioned in a group check-in from depth 0, depth 2 to

[2] In what follows, we'll be using the term 'group' to refer to sets of at least two (2) users – and distinguish it from the term 'individual', which refers to a singleton set (one user).

the check-ins of the users who are mentioned in a group check-in from depth 1, and so on.

We completed two crawls with no location constraint at depth 2 and 3. Subsequently, we identified the city with most check-ins for each country and performed a crawl constrained to the geographical coordinates of the city. To do that, we used the geographic coordinates that are associated with tweets and indicate the location of the user the moment when they generated the tweet. Since not all tweets are tagged with such geographic coordinates (due to the different privacy choices of Twitter users), for many cities we were not able to retrieve a sufficient number of tweets – and thus neither check-ins.

At the end of all crawls, we had a global dataset with approximately 143 K users, 522 K venues, 780 categories, 453 K groups, 5.6 M check-ins and 1 M group check-ins. Figure 2 shows a map with the check-ins around the globe. The data collection was done between September and October 2016.

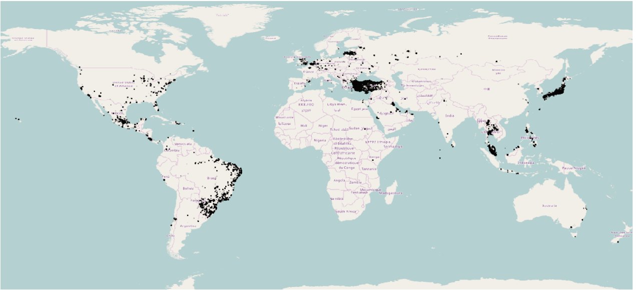


Figure 2: Map of the check-ins of all the collections together.

The top 10 cities from the collection are presented in Table 2. The names of the cities were obtained by assigning to each check-in the closest city from the Geonames database.[3] For this purpose, we used R-Trees [12].

City	Total Check-ins	Total Venues	Total Categories	Group Check-ins	Group Venues	Group Categories
Istanbul	483,214	25,953	402	43,096	8,072	297
Izmir	369,627	16,306	378	37,105	4,865	263
Mexico City	95,422	15,805	354	12,612	4,839	271
Kuala Lumpur	69,861	12,376	359	3,553	1,843	203
Bursa	59,931	4,465	283	5,459	1,218	164
Aydn	58,864	5,386	305	6,127	1,562	181
Izmit	45,575	2,961	255	4,189	818	128
Antalya	41,408	3,855	277	6,495	1,245	164
Mugla	40,148	3,520	276	5,121	1,129	173
Mytilene	40,027	2,662	219	3,220	796	131

Table 2: Top 10 cities in the data collection.

To improve the quality of our experiments, we remove possible biases caused by bots and very active users. Bots and very active users have a big geographical dispersion in their check-ins. We filtered the dataset by removing the last quartile of the users according to the standard deviation of their geographical mobility. Also, we removed approximately 2.3 M check-ins of irrelevant categories to our research (i.e., *Residence*, *States & Municipalities*, *Professional & Other Places* and *Event*, *College & University*, *Travel & Transport*).[4]

4 EXPLORATION OF GROUP BEHAVIOR

In this section, we provide an exploration of the dataset, in terms of statistics that describe various aspects of group behavior.

4.1 Group Size and Activity Dispersion

We investigate group sizes as well as the distance and time between their check-ins. Figure 3 shows the group size frequency. We filter the groups with maximum 12 participants. Figure 4 presents the time and distance between check-ins for users and groups. We removed the last quartile of time and distance.

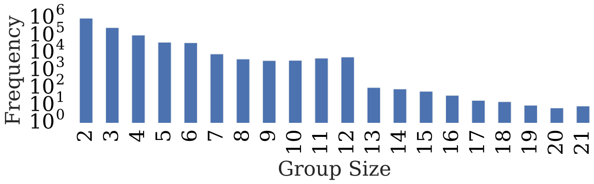


Figure 3: Group size check-in frequency (logarithmic scale).

4.2 Category Preferences

Next, we analyze the preferences of individuals and groups for venues of different categories and identify differences between the two. For category preferences we construct one ranked list of the preferred categories for individuals and one groups – and compare them using the Kendall-tau ranking correlation coefficient.

We compare the preferences at three levels: global, per city, per user. By comparing the most frequent categories for users and groups globally, the global Kendall-tau is 0.82. Figure 5 presents the Kendall-tau at the city and user level. In Figure 6 we give examples of a city and a user by parallel coordinates.

For the categorical information we use the Foursquare category tree to compute the similarity. This helps us find similar venues among different depths of the category hierarchy. For instance, if a venue is a “mexican restaurant” and another is “mediterranean restaurant” there will be some similarity between the venues because both are in the “food” main category.

4.3 Location Preferences

Figure 4 suggests that individual and group check-in behavior is different in frequency and distance spread. Next we investigate if the areas where individuals usually check-in are the same for the group.

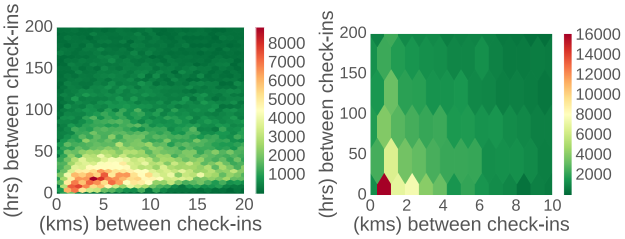


Figure 4: Frequency of time and distance between check-ins for users (left) and groups (right).

[3] http://download.geonames.org/export/dump/

[4] More information about the categorization of venues can be found in the Foursquare documentation. https://developer.foursquare.com/categorytree

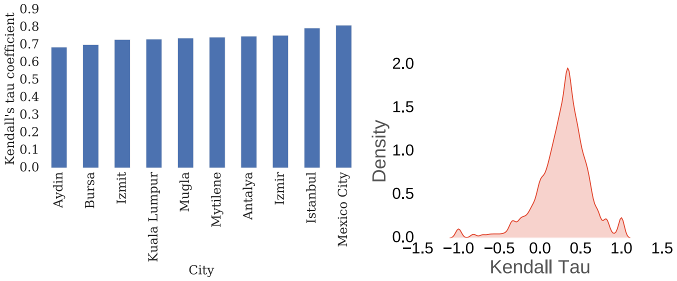


Figure 5: Kendall-tau for user and group category preferences. Left: average by city; Right: distribution on users.

To measure how much users travel to meet with a group, we first need to identify the areas where users and groups are. A well-known technique that enables us to do this is DBSCAN [10]. We use the *Vincenty distance* as the metric to identify the clusters. Figure 7 shows the user and group check-in clusters as well as their distance.

With the identified clusters, we can define a weighted average of movement for user to the groups. The following are the steps required for the whole analysis. In Step 1, for each user, we compute the centers c_u of the geographical clusters of their check-ins and the total check-ins per cluster w_u. In Step 2, for each group, we compute the centers c_g of the geographical clusters of their check-ins and the total check-ins per cluster w_g. In Step 3, we compute the weighted average distance as

$$d(c_u, c_g) = \frac{\sum_{c_{ui}} \sum_{c_{gj}} w_{ui} w_{gj} d_{vinc.}(c_{ui}, c_{gj})}{\sum_{c_{ui}} \sum_{c_{gj}} w_{ui} w_{gj}}. \qquad (3)$$

We want to allow clusters to form where the POIs are at maximum 1.5 km (i.e., about 10 blocks) away from each other – and, if a POI is too far away, we consider it as an independent cluster. We used an *epsilon* of 1.5 km and *minimum points* of 1 as the parameters for the DBSCAN clustering.

Figure 8 shows the KDE for the weighted average traveling distance that users need to move to meet with the groups.

To use the geographical feature in the recommender systems we projected the latitude and longitude from the World Geodetic System (i.e., WGS84 Model) to a sphere in the Cartesian coordinate system.

5 RECOMMENDATION ALGORITHMS

We experiment with a large set of recommender systems from the literature that we use as baselines. Then, for each recommender system we create variants that differ along three dimensions. Firstly, they differ in whether we include a pre-processing step that filters POIs near the areas where the group has already been. Secondly, they differ in whether we include as features the category and location of the POI to the recommender system. Thirdly, they differ in whether the recommender system is trained using individual user check-ins or group check-ins. The variants that include the pre-processing step, category and location features and are trained on the group check-ins are collectively reffered to as *Geo-Group-Recommender* (GGR)

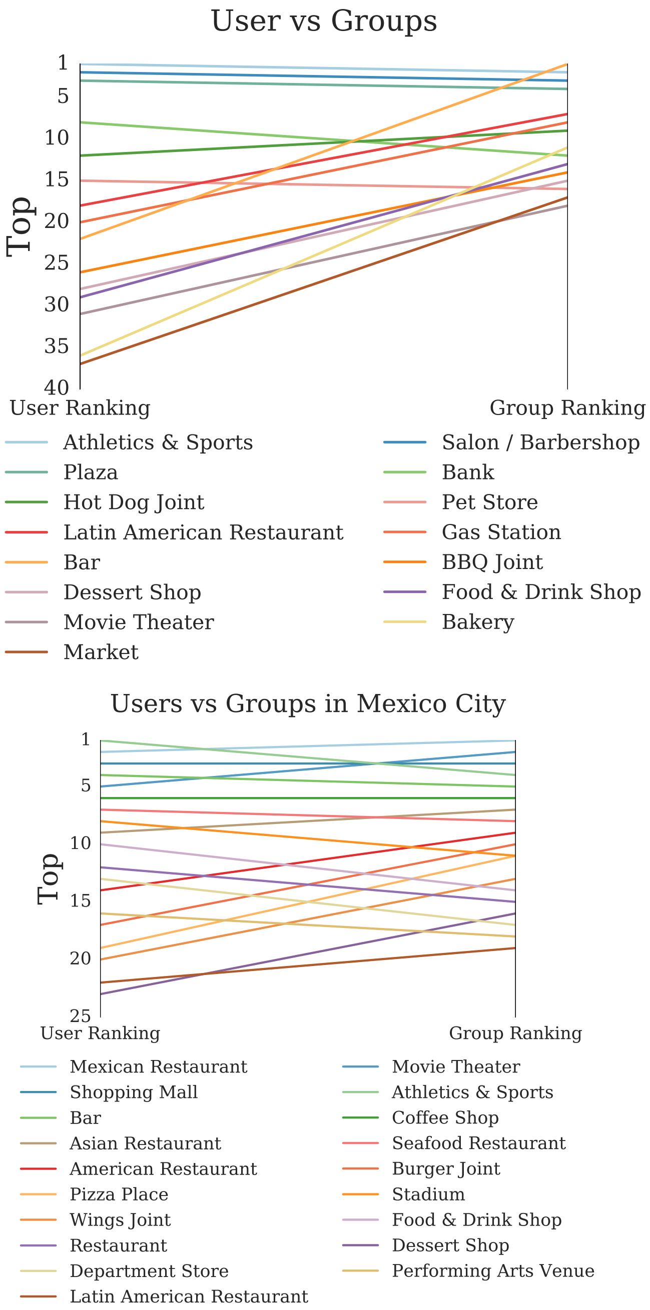


Figure 6: Category comparison using the check-ins of a single user and her groups. The Kendall-tau is 0.1(top). Total check-ins count for all the users and groups in Mexico City. The Kendall-tau is 0.8(bottom).

The motivation for the first type of variant is that, based on the inter-distance distribution of Figure 4, we know that groups do not travel much between their check-ins. Therefore, it is natural to narrow geographically the recommendations for the groups, we do this by fitting a Gaussian KDE. KDE helps us to differentiate the dense areas for a group based on the geographical check-in

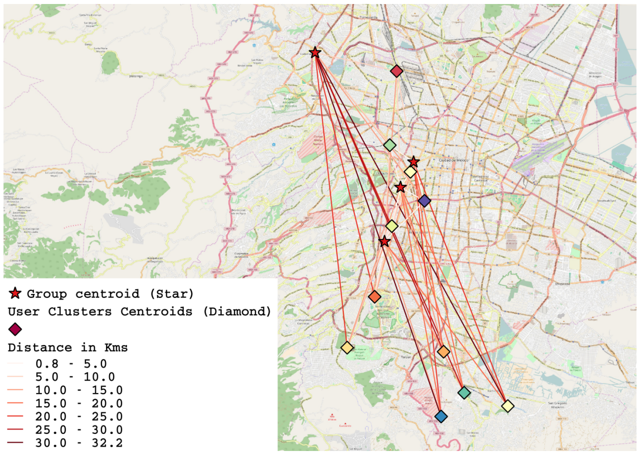


Figure 7: A map of Mexico City with an example of a user and her groups location preferences. The *diamond* shape are the centroids of the user check-in clusters. The *star* shape are the centroids of the group cluster.

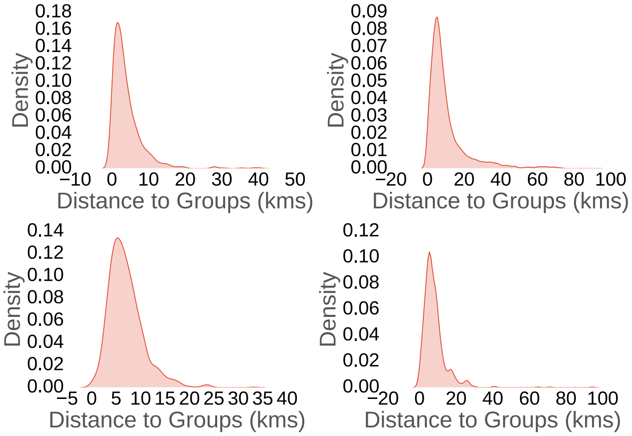
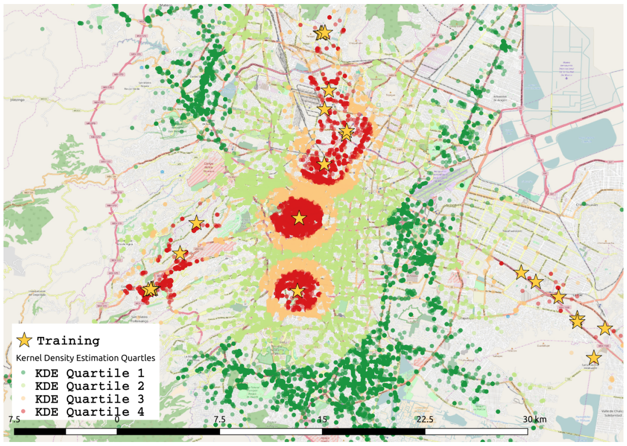


Figure 8: The KDE of users individual location preference vs. group location preference in four major cities. Lower means that the user had to travel less to meet the groups. From left to right and top to bottom: Istanbul, Izmir, Mexico City and Kuala Lumpur.

distribution, as shown in Figure 9 for one group. Specifically, we fit a KDE for each group using the check-ins in the training dataset. Subsequently, we compute a density score at the location of each venue, and keep as candidates for recommendation only the venues in the highest (densest) quartile. These POIs are then passed to the recommender system for ranking. The variants that include this pre-processing contain the keyword *KDE* in their name.

The second type of variant includes also category and/or geolocation features. These variants are denoted with *GEO* and *CAT* in their acronym.



(a) A Map showing the POIs that could be recommended in Mexico City colored by the KDE score for a particular group. The *star* shapes represent the check-ins used to fit the KDE. In our experiments, we used a fixed bandwidth of 0.2 and picked venues at the 4th quartile as POIs candidates.

Figure 9: Example of pre-processing of geographical information to calculate the KDE for a particular group. Please view in color print.

Model	Acronym
IALS Matrix Factorization [22]	IALS
KDE ∩ IALS Matrix Factorization	KDE IALS
SGD Matrix Factorization [14]	SGD
KDE ∩ SGD Matrix Factorization	KDE SGD
Item To Item [24]	ITEM-ITEM
KDE ∩ Item To Item	KDE ITEM-ITEM
Popularity Recommender [24]	POP
KDE ∩ Popularity Recommender	KDE POP
Content Based Recommender [24]	CB
KDE ∩ Content Based Recommender	KDE CB

Table 3: Models used to generate recommendations.

Among the third type of variants we distinguish three subtypes depending on the aggregation function of individual preferences. Specifically, the recommender systems generate a rating $r_{u,i}$ for each pair of user u and candidate venue i that are combined with one of the following aggregation schemes [24]:

- average individual ratings (AIR), which considers the average rating of each item,

$$\hat{r}(G,i) = \frac{\sum_{u \in G} r_{u,i}}{|G|}, \tag{4}$$

- average without misery (AWM), which assigns to items the average of their individual ratings under a certain threshold,

$$\hat{r}(G,i;s) = \frac{\sum_{u \in G; r_{u,i} > s} r_{u,i}}{|G|} \tag{5}$$

- and average least misery (ALM), which considers the minimum of their individual ratings,

$$\hat{r}(G,i) = \frac{\sum \min_{u \in G} r_{u,i}}{|G|}. \tag{6}$$

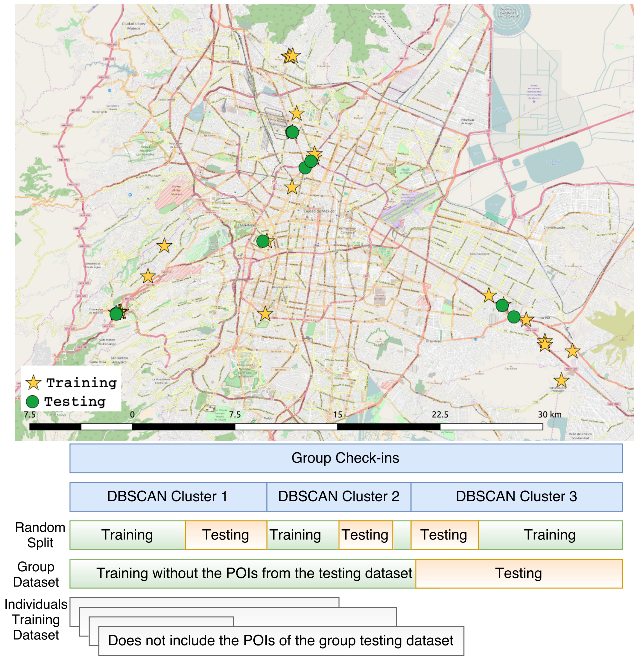


Figure 10: Random split per group and cluster. An example of the group check-ins split is shown at the top and the data split at the bottom.

Table 3 includes the names of the recommender systems that differ along the first two dimensions. Recommender systems that are trained on individual user check-ins have one of the three acronyms (i.e. AIR, AWM, ALM) appended to their name.

Evaluation Methodology

Our experiments focus on the cities with most of the group check-ins – i.e., Istanbul, Izmir and Mexico City. We split the check-ins per group cluster (i.e. clusters detected by DBSCAN) to create two datasets. The training set contains (apprx. 70%) of check-ins at the group cluster and the remaining comprises the testing set (apprx. 30%). Group clusters with size lower than the median were added to the training set. We combined all the group cluster splits to create one global training and one testing dataset. Figure 10 describes this process.

We used Turi's GraphLab Create[5] implementation of the recommender systems listed in Table 3. We used GraphLab's built-in function for tuning the parameters of the models. For this purpose, we used 5% of the training dataset as a validation set. The experiments were conducted in a single machine with 40 cores and 200 GB of RAM and ran for a day.

For performance metrics we use precision and recall at different cutoffs K (i.e. 5, 10, 20, 30, 40, 50),

$$\text{Precision@}K = \frac{\text{Visited POIs in Cluster} \cap \text{Recommended POIs}}{\text{Recommended POIs}},$$

and

$$\text{Recall@}K = \frac{\text{Visited POIs in Cluster} \cap \text{Recommended POIs}}{\text{Visited POIs in Cluster}}.$$

6 RESULTS

Building upon the discussion of Sections 4 and 5, we now provide answers to the research questions we set in the beginning of this work.

6.1 RQ 1: Group Behavior

Observation 1: *Groups move less than users and their check-ins are less frequent.*

Based on the analysis of time and distance between check-ins we observed that 75% of the user check-ins occur between 2.5 days and within a distance of 10 kms. However, 75% of the groups check-ins happen between 8 days and within 5 kms. 50% of the groups move just 1 km between their check-ins.

Observation 2: *Groups in LBSNs are small.*

Most of the check-ins are made of groups of two people. Groups with size greater than 12 people are rare.

6.2 RQ 2: Individual vs. Group Preferences

Observation 3: *Group prefer other areas than their members.*

In Figure 8 we can observe that users needs to travel to parts of the city that they are not usually going. The KDE of the average weighted distance saturates between 5-10 kms.

Observation 4: *Groups prefer other types of venues than their members.*

In Figure 5 (right) we observe that top categories for users are different than groups. The Kendall-tau most dense part is around 0.4.

6.3 RQ 3: POI Recommendation

Our main result is the comparison of recommender system algorithms in different large cities. GGR models are top performers and the types of recommender systems perform differently among cities. Figure 11 shows the results for Istanbul, where KDE IALS performs best for both precision and recall. Figure 12 shows the results for Izmir, where in contrast, KDE SGD GEO performs best for both precision and recall. Finally for Mexico City (Figure 13), KDE IALS again performs best for recall@5-10 while KDE SGD CAT for recall@20-50. Best performing methods are the same for precision as well.

Observation 5: *Training recommender systems for groups works better than combining individual recommendations.*

By answering **RQ1** and **RQ2** we show that the behavior of users and groups is different. This is the reason why combining individual recommendations by averaging under-performs the group models. A better approach is to train a model based on groups information only. The results for comparing iALS for groups vs. aggregating for individual users in Figure 14 show the superiority of group-based over individual recommendations.

Observation 6: *A Geographical KDE improves the performance of new POIs recommendation in the area where the group check-ins.*

Using the geographical KDE prioritize new POIs around the group preferred areas. This improves the models for all the cities, as seen in Figs. 11–13.

Observation 7: *Geography and Categorical Features are important.*

[5] https://turi.com/

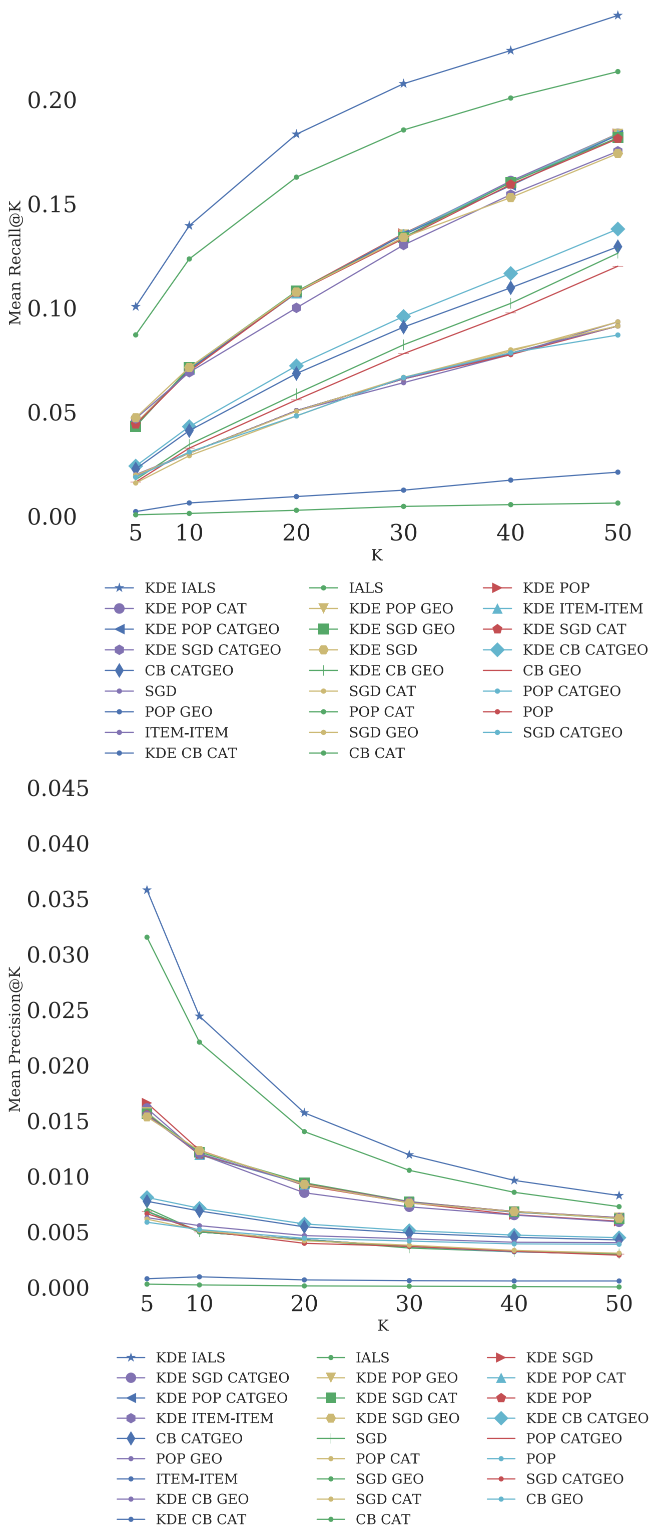


Figure 11: Istanbul Recall@K(top) and Precision@K(bottom). The legends are sorted by the best performance models from left to right and top to bottom.

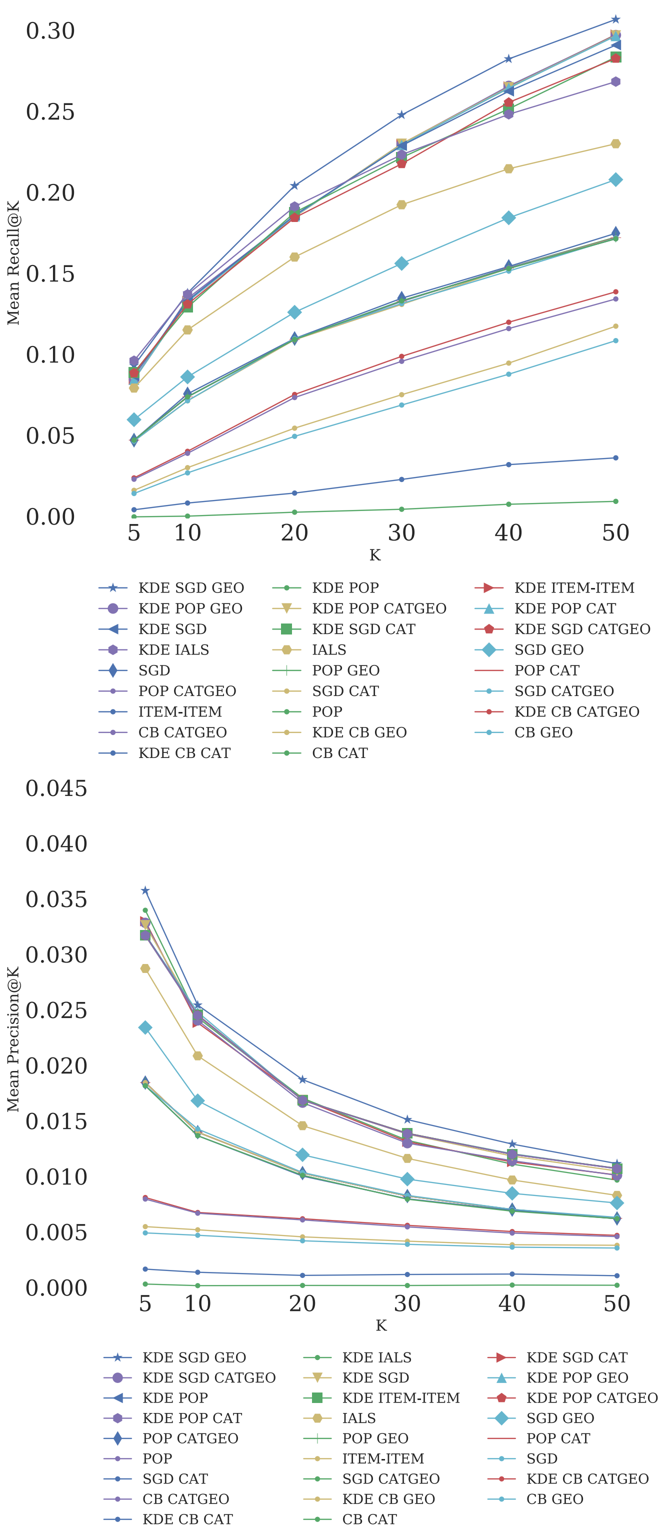


Figure 12: Izmir Recall@K(top) and Precision@K(bottom). The legends are sorted by the best performance models from left to right and top to bottom.

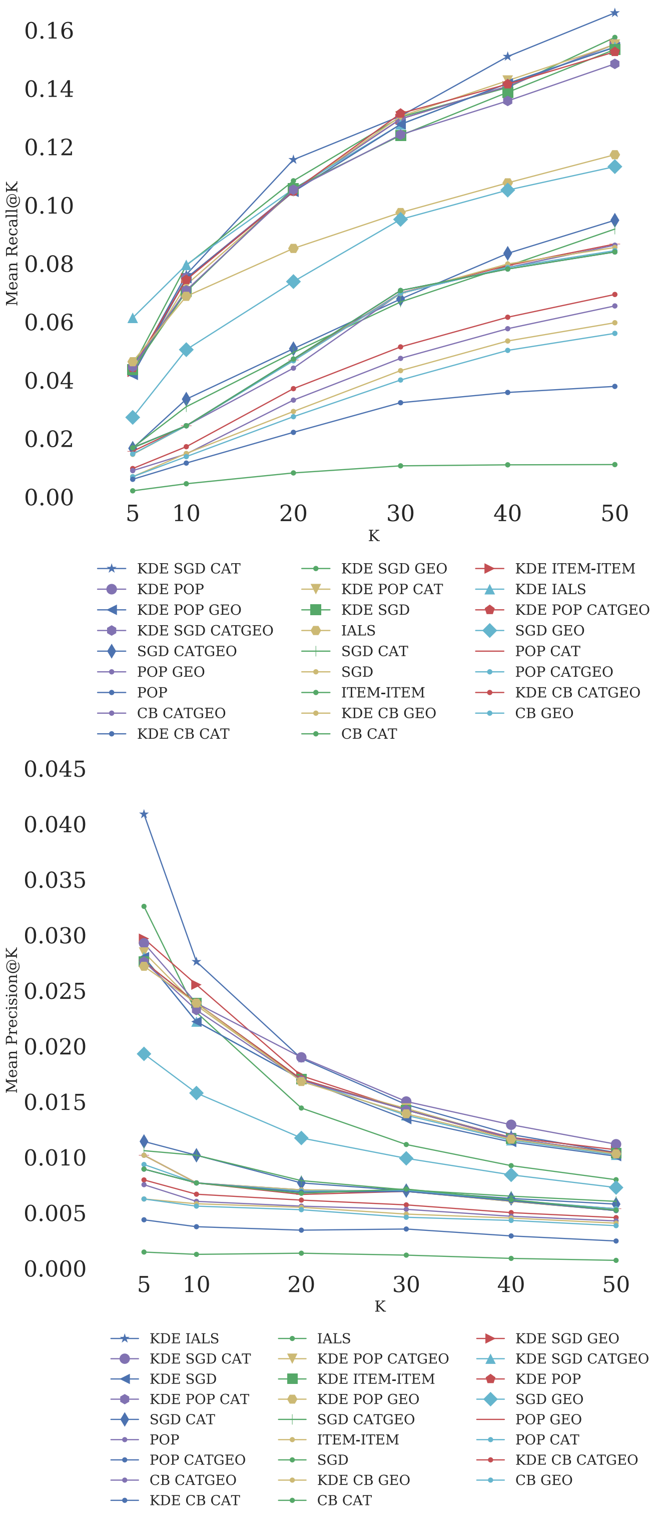


Figure 13: Mexico City Recall@K(top) and Precision@K(bottom). The legends are sorted by the best performance models from left to right and top to bottom.

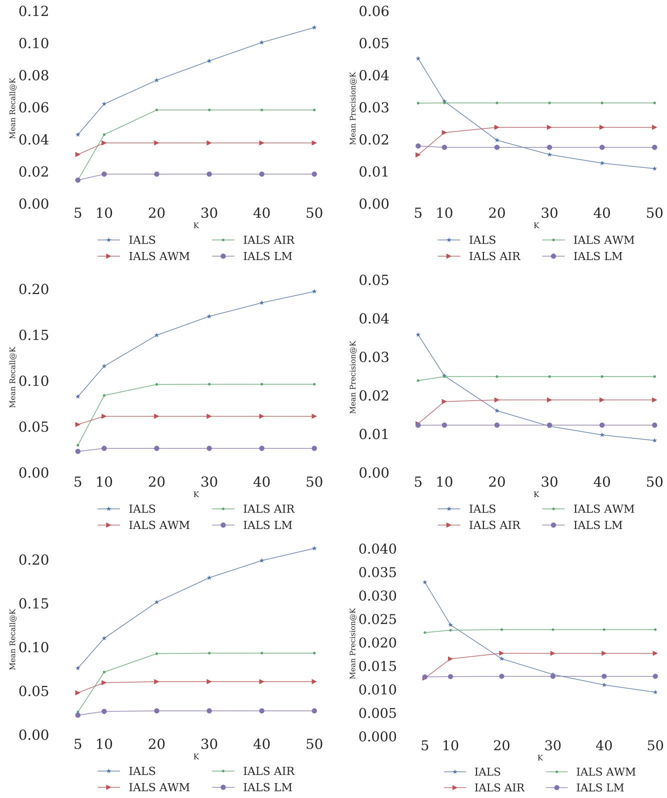


Figure 14: Combining individual iALS recommendations by equations(4–6) under-perform the group model. From top to bottom: Mexico City; Istanbul; Izmir. Left: Recall; Right: Precision.

In addition to geographical KDE, in our experiments the SGD, POP and CB models with either categorical or geographical information performed better than the same model without these features.

7 DISCUSSION

7.1 Implications

When groups decide where to go, they could save time if they receive a tailored top list of venues. The recommended list should be in-line with the group preferences in order to be useful and reasonable. Our findings suggest that this is feasible for the areas that we know a group has been to in the past. However, there are other possible POIs recommendations for groups. For example, we could recommend a new area in the city with venues that they might like. Or, we could recommend individuals to go together for the first time to a place.

7.2 Future Work

Our data collection was limited to publicly available data from Swarm and Twitter and the check-ins were extracted from the latest 200 tweets of the users. Our crawling strategy collected data from cities of Istanbul, Izmir and Mexico City. Other cities like New York where Swarm is very popular did not appear in our collection.

We could not retrieve the entire social graph for the users. Using such information (e.g. POIs popularity, areas and categories in the users ego network) would give rise to other ways to combine user preferences into group recommendations. Future work could be to understand the reasons why the models perform different for the different cities. Possible reasons are differences in the sizes of the cities, how easy is to move inside a city, the lack of data for groups (i.e. cold start problem) or even natural boundaries (e.g. rivers, mountains). Collecting more data could help to generalize our findings among different cultures, nations, urban or rural areas.

8 CONCLUSIONS

Our research work presents empirical findings on recommending new POIs to groups. To the best of our knowledge, this is the first study that uses group information in LBSNs without using specific assumptions and heuristics to detect the groups. Our experiments on over 5.6 M user check-ins and 1 M group check-ins show that users and groups prefer different geographical areas and categories. We show that recommending POIs near the areas were groups move is feasible. A major finding is that training a model using group profiles performs better than combining individual recommendations and that the GGR models generally are top performers.

9 ACKNOWLEDGMENTS

The publication was supported by the PIAC_13-1-2013-0205 project of the Research and Technology Innovation Fund, by the Momentum Grant of the Hungarian Academy of Sciences and by the Mexican Postgraduate Scholarship of the Mexican National Council for Science and Technology (CONACYT) and by the European Institute of Innovation and Technology (EIT) Digital Doctoral School. This work has been supported by the Academy of Finland project "Nestor" (286211) and the EC H2020 RIA project "SoBigData" (654024). Special thanks to Turi for the GraphLab Academic License.

REFERENCES

[1] Aris Anagnostopoulos, Reem Atassi, Luca Becchetti, Adriano Fazzone, and Fabrizio Silvestri. 2016. Tour recommendation for groups. *Data Mining and Knowledge Discovery* (2016), 1–32. DOI: http://dx.doi.org/10.1007/s10618-016-0477-7

[2] Linas Baltrunas, Tadas Makcinskas, and Francesco Ricci. 2010. Group Recommendations with Rank Aggregation and Collaborative Filtering. In *Proceedings of the Fourth ACM Conference on Recommender Systems (RecSys '10)*. ACM, New York, NY, USA, 119–126. DOI: http://dx.doi.org/10.1145/1864708.1864733

[3] Jie Bao, Yu Zheng, and Mohamed F. Mokbel. 2012. Location-based and preference-aware recommendation using sparse geo-social networking data.. In *SIGSPATIAL/GIS*, Isabel F. Cruz, Craig Knoblock, Peer Krger, Egemen Tanin, and Peter Widmayer (Eds.). ACM, 199–208. http://dblp.uni-trier.de/db/conf/gis/gis2012.html

[4] Jie Bao, Yu Zheng, David Wilkie, and Mohamed Mokbel. 2015. Recommendations in Location-based Social Networks: A Survey. *Geoinformatica* 19, 3 (July 2015), 525–565. DOI: http://dx.doi.org/10.1007/s10707-014-0220-8

[5] Chlo Brown, Neal Lathia, Cecilia Mascolo, Anastasios Noulas, and Vincent Blondel. 2014. Group Colocation Behavior in Technological Social Networks. *PLOS ONE* 9, 8 (08 2014), 1–9. DOI: http://dx.doi.org/10.1371/journal.pone.0105816

[6] Lucas Augusto Montalvão Costa Carvalho and Hendrik Teixeira Macedo. 2013. Users' Satisfaction in Recommendation Systems for Groups: An Approach Based on Noncooperative Games. In *Proceedings of the 22Nd International Conference on World Wide Web (WWW '13 Companion)*. ACM, New York, NY, USA, 951–958. DOI: http://dx.doi.org/10.1145/2487788.2488090

[7] Chen Cheng, Haiqin Yang, Irwin King, and Michael R Lyu. 2012. Fused Matrix Factorization with Geographical and Social Influence in Location-Based Social Networks.. In *Aaai*, Vol. 12. 1.

[8] Eunjoon Cho, Seth A. Myers, and Jure Leskovec. 2011. Friendship and Mobility: User Movement in Location-based Social Networks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '11)*. ACM, New York, NY, USA, 1082–1090. DOI: http://dx.doi.org/10.1145/2020408.2020579

[9] Berardina De Carolis. 2011. *Adapting News and Advertisements to Groups*. Springer London, London, 227–246. DOI: http://dx.doi.org/10.1007/978-0-85729-352-7_11

[10] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proc. of 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*. 226–231.

[11] Huiji Gao, Jiliang Tang, Xia Hu, and Huan Liu. 2013. Exploring temporal effects for location recommendation on location-based social networks.. In *RecSys*, Qiang Yang, Irwin King, Qing Li, Pearl Pu, and George Karypis (Eds.). ACM, 93–100. http://dblp.uni-trier.de/db/conf/recsys/recsys2013.html

[12] Antonin Guttman. 1984. *R-trees: a dynamic index structure for spatial searching*. Vol. 14. ACM.

[13] Heung-Nam Kim, Mark Bloess, and Abdulmotaleb El Saddik. 2013. Folkommender: a group recommender system based on a graph-based ranking algorithm. *Multimedia Systems* 19, 6 (2013), 509–525. DOI: http://dx.doi.org/10.1007/s00530-012-0298-5

[14] Yehuda Koren, Robert Bell, Chris Volinsky, and others. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.

[15] Defu Lian, Cong Zhao, Xing Xie, Guangzhong Sun, Enhong Chen, and Yong Rui. 2014. GeoMF: joint geographical modeling and matrix factorization for point-of-interest recommendation. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 831–840.

[16] Yi Liao, Wai Lam, Shoaib Jameel, Steven Schockaert, and Xing Xie. 2016. Who Wants to Join Me?: Companion Recommendation in Location Based Social Networks. In *Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval (ICTIR '16)*. ACM, New York, NY, USA, 271–280. DOI: http://dx.doi.org/10.1145/2970398.2970420

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

[18] Joseph F. McCarthy and Theodore D. Anagnost. 1998. MusicFX: An Arbiter of Group Preferences for Computer Supported Collaborative Workouts. In *Proceedings of the 1998 ACM Conference on Computer Supported Cooperative Work (CSCW '98)*. ACM, New York, NY, USA, 363–372. DOI: http://dx.doi.org/10.1145/289444.289511

[19] Kevin McCarthy, Lorraine McGinty, Barry Smyth, and Maria Salamó. 2006. The Needs of the Many: A Case-based Group Recommender System. In *Proceedings of the 8th European Conference on Advances in Case-Based Reasoning (ECCBR'06)*. Springer-Verlag, Berlin, Heidelberg, 196–210. DOI: http://dx.doi.org/10.1007/11805816_16

[20] Anastasios Noulas, Salvatore Scellato, Cecilia Mascolo, and Massimiliano Pontil. 2011. An Empirical Study of Geographic User Activity Patterns in Foursquare.. In *ICWSM*, Lada A. Adamic, Ricardo A. Baeza-Yates, and Scott Counts (Eds.). The AAAI Press. http://dblp.uni-trier.de/db/conf/icwsm/icwsm2011.html

[21] Mark O'Connor, Dan Cosley, Joseph A. Konstan, and John Riedl. 2001. PolyLens: A Recommender System for Groups of Users. In *Proceedings of the Seventh Conference on European Conference on Computer Supported Cooperative Work (ECSCW'01)*. Kluwer Academic Publishers, Norwell, MA, USA, 199–218. http://dl.acm.org/citation.cfm?id=1241867.1241878

[22] István Pilászy, Dávid Zibriczky, and Domonkos Tikk. 2010. Fast als-based matrix factorization for explicit and implicit feedback datasets. In *Proceedings of the fourth ACM conference on Recommender systems*. ACM, 71–78.

[23] 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 (GeoCrowd '14)*. ACM, New York, NY, USA, 8–15. DOI: http://dx.doi.org/10.1145/2676440.2676442

[24] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2015. *Recommender Systems Handbook* (2nd ed.). Springer Publishing Company, Incorporated.

[25] Maria Salamó, Kevin Mccarthy, and Barry Smyth. 2012. Generating Recommendations for Consensus Negotiation in Group Personalization Services. *Personal Ubiquitous Comput.* 16, 5 (June 2012), 597–610. DOI: http://dx.doi.org/10.1007/s00779-011-0413-1

[26] Waldo R Tobler. 1970. A computer movie simulating urban growth in the Detroit region. *Economic geography* 46, sup1 (1970), 234–240.

[27] Jia-Dong Zhang and Chi-Yin Chow. 2015. GeoSoCa: Exploiting Geographical, Social and Categorical Correlations for Point-of-Interest Recommendations. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '15)*. ACM, New York, NY, USA, 443–452. DOI: http://dx.doi.org/10.1145/2766462.2767711