Multi-Objective Optimization Based Location and Social Network Aware Recommendation

Makbule Gül cin Özsoy
Department of Computer Engineering
Middle East Technical University
Ankara, Turkey
e1395383@ceng.metu.edu.tr

Faruk Polat
Department of Computer Engineering
Middle East Technical University
Ankara, Turkey
polat@ceng.metu.edu.tr

Reda Alhajj
Department of Computer Science
University of Calgary
Calgary, Alberta, Canada
alhajj@ucalgary.ca

***Abstract*— Social networks, personal blog pages, on-line transaction web-sites, expertise web pages and location based social networks provide an attractive platform for millions of users to share opinions, comments, ratings, etc. Having this kind of diverse and comprehensive information leads to difficulties for users to reach the most appropriate and reliable conclusions. Recommendation systems form one of the solutions to deal with the information overload problem by providing personalized services. Using spatial, temporal and social information on recommender systems is a recent trend that increases the performance. Also, taking into account more than one criterion can improve the performance of the recommender systems. In this paper, a location and social network aware recommender system enhanced with multi objective filtering is proposed and described. The results show that the proposed method reaches high coverage while preserving precision. Besides, the proposed method is not affected by the range of ratings and provides persistent results in different settings.**

I. Introduction

People normally purchase, rent, use, or subscribe to a variety of items or commodities. This could be achieved based on personal experience or recommendation by friends, family or experts. However, sometimes the task is completed with little or even no previous information to help in the decision making process. Actually, the more information and feedback a person acquires about an item the better decision he/she may take. Thanks to the advancements in Web technologies which allow users to provide or share their views and experience about various items and commodities. Searching the Web it is very common to witness a wide range of reviews reflecting opinions from the very negative to the very positive. Of course, every person has his/her own expectations and the review reported and shared reflects the particular item met his/her own expectations. Relevant information is provided by millions of users for many items on different platforms, such as social networks (Twitter, Facebook etc.), personal blog pages, on-line transaction web-sites (Amazon.com etc.), expertise web-pages (Slashdot.com, Advogato.org etc.) and location based social networks (Foursquare, Facebook Places etc.). Having this kind of wide spectrum of views leads to difficulties for a user to reach the most appropriate and reliable information/item. Recommendation systems form one of the solutions to deal with the information overload problem via providing personalized services, i.e., by identifying and highlighting posters who share similar expectations with the seeker.

Recommendation systems suggest items to a user by estimating the ratings that the user would give to that item. Recommendation can be given on any subject such as books, movies, news, jokes, vacations, etc. The process of estimating ratings can be performed by using heuristics and machine learning approaches. In the literature, there are three basic approaches to give recommendations, namely content based approaches, collaborative filtering approaches and hybrid approaches.

Location based social network (LBSN) is a new concept that allows users to share their location (e.g., check-in) with their on-line friends, comment on certain locations (e.g., restaurants), connect with friends and find other people who are nearby. Traditional recommender systems consider neither location nor social networks. With the help of LBSNs, e.g., Foursquare, Facebook Places, it is possible to embed both location and social network information into recommender systems and hence increase the performance. Most of the recommender systems are based on single criterion, such that they aim to evaluate a given item based on overall rating. However, a user may consider more than one criteria while deciding to use/buy an item [2]. In order to give more accurate recommendations, a recommender system can take advantage of considering multiple criteria.

In this paper, a location and social network aware recommender system enhanced with multi objective filtering is proposed and described. Our contributions may be enumerated as follows:

- We propose a new recommendation method that combines ratings, location and social network information. These features are used in a multi objective optimization setting to select neighbors. Afterwards, past preferences of these neighbors are used to give future recommendations to the target users.
- We introduce several different settings, namely Base, Weight Based, Rate Based and Rate and Weight Based. In Base setting, we use binary check-in data and weight neighbors equally. In Weight Based setting, neighbors are assigned different weights. In Rate Based setting, binary data is converted into ratings and the related calculations are performed on them. Rate and Weight Based setting is the combination of the previous two.

Reda Alhajj is also affiliated with the Department of Computer Science, Global University, Beirut, Lebanon

- We evaluate the results on a Foursquare dataset, namely Checkin2011 dataset [6]. We consider a subset of this dataset and used data in January for training and data in February for testing. The evaluation results show that the proposed method increases coverage while preserving precision. Besides, it can handle different rating ranges and provides persistent results.

The rest of this paper is organized as follows. Information on recommendation systems that use location and/or social network information or multi-objective optimization methods is given in Section II. In Section III the proposed recommendation system is explained. In Section IV, the evaluation settings and the results are explained. Also, the comparison of the proposed method to the state-of-the-art methods is given. The paper is concluded in Section V.

II. Related Work

The goal of recommendation systems is suggesting to a user the items that might be of interest to him/her [16]. The most common form of giving recommendation is estimating rating of un-seen/un-rated items by the user and choosing items with the highest estimation values [22]. In the literature, there are three basic approaches to give recommendations, namely content based, collaborative filtering and hybrid approaches. While the content based approach uses item similarity to give recommendations, collaborative filtering uses past preferences of the users to decide which item to recommend. On the other hand, hybrid methods basically combine the previous two approaches to give recommendations.

The data to be processed by a recommendation system has basically three elements, which are *user*, *item* and *rating*. In most of the algorithms, these elements are represented by a matrix or a graph. Traditional recommender systems consider neither location nor social networks for users or items. Today many applications provide both location and social network information to the researchers. With the help of these applications, e.g., Foursquare, Facebook Places, etc. it is possible to embed both location and social network information into recommender systems for more realistic and concrete outcome.

In subsections II-A and II-B related work on location aware and social network aware recommendation systems are explained, respectively. Then, a brief description on multi-objective recommender systems is given in subsection II-C.

A. Location Aware Recommender Systems

Analysis show that users from the same spatial region prefer different items than users in another region (preference locality) and users prefer location based items (e.g., restaurants) which are in limited/shorter distance (travel locality) [11]. These findings reveal that recommendation systems should consider spatial information such that recommendations should be given by users who live in the same region and recommended items should be close to the target user.

LARS [11] is a framework that produces location-aware recommendations. The authors introduce three types of settings which use user location information only, item location information only, and both user and item locations. For the first setting, they perform user portioning using an adaptive pyramid structure. For the second setting, they use travel penalty and avoid recommending distant items. For the last setting, the previous two approaches are combined together to give recommendations. Also in LARS* [20], the proposed methods are improved for better scalability and efficiency.

The work described in [19] proposes a map-based recommendation system that collects context information, location, time, weather and user request. Using these features, they model user preferences by Bayesian network, and inferred the most appropriate item to suggest and show it on map.

The work described in [25] uses GPS history of users to give location recommendations (e.g., Where should I go?) and activity recommendations (e.g., What else can I do there?). After modelling users' past location and activity preferences and mining information on locations and activities (e.g. correlations, features), they apply matrix factorization method.

The work described in [21] proposed a method, named Extended Feature Combination (EFC), to deal with data containing user, location, activity and rating information. The author constructs 2D data structure from higher dimensional data and introduces a general data combination approach.

The work described in [24] proposes a point of interest (POI) recommendation method that combines location, social network and past ratings. With the help of these information, the authors created three different ranking lists and used a linear fusion framework to integrate them into one.

The work described in [5] proposes a recommender system that recommends successive personalized POI. They propose a novel matrix factorization method, FPMC-LR, which uses both location and temporal information. They showed that POI recommendation is very time-critical and its performance is related not only to past preferences but also to current location.

B. Social Network Aware Recommender Systems

Online social network sites brought a new trend to recommender systems, which relies on the assumption that friends share similar tastes. Using this information, data sparsity problem can be handled more efficiently and the quality of recommendations can be increased [12].

The work described in [7] proposes a social-historical model to explore users' check-in behavior in LBSNs. Their model integrates users' past preferences and considers social ties. For historical tie analysis, they introduce a language model, after observing common features of language processing and location-based social networks. For social ties, they investigate that friends have higher check-in similarity. To model both effects, they add user's social ties as a regularization term to historical ties.

SoCo [12] is a context-aware recommender system that incorporates social network information. The authors applied random decision tree algorithm to partition the input matrix into subgroups of similar contexts. For predicting ratings, matrix factorization method is used, where social network information was used as a regulator. Also, in this paper a context-aware Pearson correlation coefficient calculation method is proposed.

The work described in [23] aims to exploit both social and geographical information existing in LBSNs to support location recommendation services. The authors proposed a friend-based collaborative filtering (FCF) approach and its geo-measured version (GM-FCF). In FCF only friends of the target user are considered when performing collaborative filtering. The researchers also observe that nearby friends share common places more than geographically distant friends. Using this information, they propose GM-FCF, where friend similarity was also affected by geographical location.

The work described in [13] aims to provide a general method that incorporates social networks to recommender system. The authors introduce two different regularization terms to matrix factorization. The first regularization, namely average-based regularization, uses the assumption that target user's taste is close to the average of the his/her friends' taste. While taking the average, they weighted the taste based on target user-friend similarity. The second regularization, namely individual-based regularization, uses friends' taste individually.

The work described in [17] proposes a new similarity metric based on user's past preferences and his/her social activities. Besides, they propose a new method. For the similarity calculations they consider similarity based on common tags on common items, friendship, and membership to the same group. The proposed algorithm applies both item similarity and user similarity.

C. Multi Objective Recommender Systems

Most of the recommender systems are based on single criterion, such that they aim to evaluate a given item based on overall rating. A user may consider more than one criteria while deciding to use an item. For example, while choosing a movie to watch, a user may not only consider the overall rating, but also the genre, the actors/actresses, the director, etc. In order to give more accurate recommendations, a recommender system can take advantage of considering multiple criteria. The importance of multi-criteria recommender systems is highlighted in [2], [3], [1] and [14].

The common techniques described in the literature are taking a linear combination of multiple criteria, finding Pareto optimal solutions, optimizing for only the most important criterion and consecutive optimization for one criterion at a time.

Taking a linear combination of multiple criteria technique combines the results which are obtained in prediction step for each criterion. In [9], marginal utility value for each criterion is used while taking the sum of prediction scores. Similarly, in [15] the sum of prediction scores is calculated. Also, in the same paper another approach which considers weights assigned by users to indicate importance of criteria is used for getting the sum. After combining the multiple criteria ratings, the items that maximize the value are recommended to the user.

The technique of finding Pareto optimal solutions discovers several good items among large number of candidates [2]. This approach does not require weights priorly-assigned to the criteria. The approach described in [10] is used for a restaurant recommendation system where users indicate their preference. For example, in this system a user may look for cheap and Italian food, and the system tries to find the optimal restaurant based on these criteria. In [2], it is stated that these systems may suffer from scalability problem when the number of criteria increases.

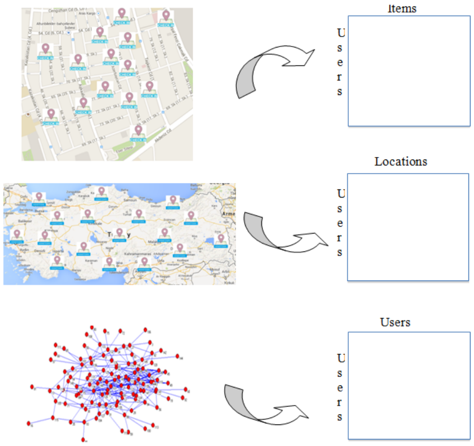


Fig. 1. Data with multiple features

Optimizing for only the most important criterion technique filters recommendations using a single criterion, which is indicated by the user. In the fourth technique, consecutive optimization for one criterion at a time, the ranked list of criteria is used for filtering the recommendations.

III. Location and Social Network Aware Multi Objective Recommendation System

Traditional recommender systems do not consider location or social network information. Besides they are usually based on a single-criterion, namely overall rating for an item. In this study, we propose a method that considers not only users, checkins (as items) and ratings, but also hometown(as location), friendship and social network information (Figure 1). Most of the research on these kinds of data, like containing multiple criteria, aggregate the preferences into a utility function, usually by getting the weighted sum. Unlike previous works, we combine all the criteria into a vector representation and decide on the results using multi-objective optimization methods.

In section III-A, the proposed recommendation system is explained. Section III-B includes implementation details based on the characteristics of the input data.

A. Location and Social Network Aware Multi Objective Recommendation System

The proposed system is composed of the following steps:

- Similarity calculations using check-in, hometown, friendship and influence information
- Selection of the most similar users(neighbors) to the target user by multi-objective optimization methods
- Selection of items using traditional approach used in user-based collaborative filtering

We considered four different settings:

- Basic: Input rating information is considered to be binary. For example, a user likes a place or not. In this setting, the selected neighbors are considered to have same level of effect on the target user.
- Weight Based: In this setting, the neighbors are given different weights, so that they have different level of effect on the target user. For example, if first neighbor has weight value 0.4 and the second neighbor has weight value of 0.8; then the recommendations from the second neighbor will be considered to be more (twice) important than the first one. In weight based setting, the input rating information is considered to be binary.
- Rate Based: In this setting, the input rating information is given in a range, either directly using the range of the input set or by assigning ratings later. Similar to the basic setting, in this setting the neighbors are considered to have same level of effect on the target user.
- Rate and Weight Based: This setting is a combination of rate-based and weight-based methods. In this setting, the input ratings are given in a range and the neighbors are assigned different weights.

The details of the steps and the different settings are given in the following subsections.

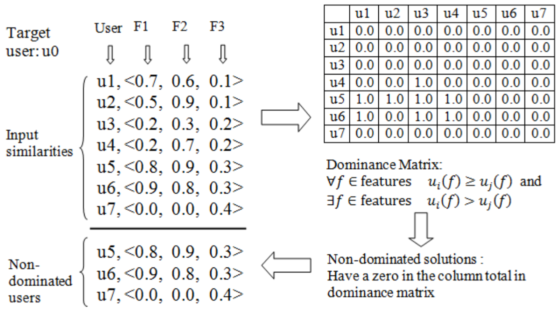
1) Similarity calculations: The first step of the method is to calculate user-user similarities based on several different contexts, such as location, friendship, and past preferences. The calculation of similarities can be based on Cosine similarity, Pearson correlation, Jaccard similarity, etc. Besides the similarities of the mentioned criteria, different criterion can be introduced to the system, such as hometown, gender, age, etc.

The equations for the common similarity metrics to calculate user similarities, namely Pearson correlation, Cosine similarity, Adjusted Cosine similarity and Jaccard similarity, are given in Equations 1, 2, 3 and 4, respectively.

In the equations the following notation is used. Users are denoted as *u* and *v* and the items are denoted as *i*. *rat(u,i)* indicates the rating of item *i* given by user *u*. $rat_{avg}(u,.)$ indicates the average rating given by the user *u*. $rat_{avg}(u,*)$ and $rat_{avg}(v,*)$ are used to denote the average rating given by user *u* to the common items rated by user *v*, or vice versa. *sim(u,v)* indicates the similarity of user *u* to user *v*.

In this step, one can choose to use Base or Rate Based settings. As explained previously, Base setting uses binary data. So for the similarity calculations, it is appropriate to use Cosine similarity or Jaccard similarity. Rate Based setting uses rating information. In this setting, using Pearson Correlation or Adjusted Cosine similarity measures are suitable for the calculations.

2) Neighbor selection: Having the similarities for each user to the others, the next step is to decide on the most similar users to the target user. For this purpose, non-dominated users are determined by finding the Pareto optimal front. We propose that non-dominated users are the ones that will affect the target most. An example of multi dimensional data is given in Figure 2. In this example, for different dimensions, namely $F1$, $F2$ and $F3$, similarities of seven users to the target user are given. The F_i values are the calculated similarities, such as check-in, hometown and friendship similarity.



	u1	u2	u3	u4	u5	u6	u7
u1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
u2	0.0	0.0	0.0	0.0	0.0	0.0	0.0
u3	0.0	0.0	0.0	0.0	0.0	0.0	0.0
u4	0.0	0.0	1.0	0.0	0.0	0.0	0.0
u5	1.0	1.0	1.0	1.0	0.0	0.0	0.0
u6	1.0	0.0	1.0	1.0	0.0	0.0	0.0
u7	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Fig. 2. Example input and non-dominated solutions

In order to find non-dominated users, the first step is to create the dominance matrix. In a dominance matrix, each entry indicates whether the user in the row dominates the user in the column. The formula to fill the entries is given in Equation 5, where f indicates the features used; in the example figure, Figure 2, the features are $F1$, $F2$ and $F3$. Based on the example similarities, the output dominance matrix is given in the figure.

$$dom(u,v) = \begin{cases} 1.0 & \forall f u(f) \geq v(f) \text{and} \exists f u(f) > v(f) \\ 0.0 & \text{otherwise} \end{cases} \tag{5}$$

Having the dominance matrix, non-dominated users are decided by looking at the column sums of the dominance matrix. Users whose column sum is equal to 0.0 are the non-dominated users.

In order to collect as many neighbors as given, an iterative process of neighbor collection can be applied. First, we apply the method of finding non-dominated users explained previously. If the number of non-dominated users is less than the given neighbor count, we remove the selected users from the data representation and re-apply the method to find non-dominated users. We continue this process until the predefined number of neighbors count is reached.

3) Item selection: The last step is to give recommendation of items. Items preferred by neighbors are considered as candidate recommendations. The more neighbors recommend an item, the more the score of the candidate item is. At the end, the *top-n* items with the highest score are suggested to the user.

The score calculation is performed according to Equation 6. In the equation, *s(u,i)* is the predicted score of item *i* that target user *u* will give, *v* is a user who is chosen as a neighbor to user *u*. *sim(u,v)* is the similarity of users *u* and *v* and *rat(v,i)* is the rating given to item *i* by user *v*.

$$s(u,i) = \sum_{v \in Neighbours} sim(u,v) rat(v,i) \tag{6}$$

$$sim(u,v) = \frac{\sum_i (rat(u,i) - rat_{avg}(u,*))(rat(v,i) - rat_{avg}(v,*))}{\sqrt{\sum_i (rat(u,i) - rat_{avg}(u,*))^2}\sqrt{\sum_i (rat(v,i) - rat_{avg}(v,*))^2}} \quad (1)$$

$$sim(u,v) = \frac{\sum_i rat(u,i) rat(v,i)}{\sqrt{\sum_i rat(u,i)^2}\sqrt{\sum_i rat(v,i)^2}} \quad (2)$$

$$sim(u,v) = \frac{\sum_i (rat(u,i) - rat_{avg}(u,.))(rat(v,i) - rat_{avg}(v,.))}{\sqrt{\sum_i (rat(u,i) - rat_{avg}(u,.))^2}\sqrt{\sum_i (rat(v,i) - rat_{avg}(v,.))^2}} \quad (3)$$

$$sim(u,v) = \frac{|rat(u,.) \cap rat(v,.)|}{|rat(u,.) \cup rat(v,.)|} \quad (4)$$

The calculation of the score is dependent on the settings.

- Base setting: All neighbors are considered to be equal, such that the similarity value of each neighbor is assigned to 1.0. Besides, in this setting rating score is also assigned to 1.0. Instead of assigning the rating score to 1.0, a value which is calculated from the data can also be used, such as number of times a check-in is done in the related location by the related neighbor.
- Weight Based setting: Neighbors are assigned different weights according to their similarities to the target user. The weight of a neighbor is assigned by taking the average of similarities that are used in the related method (Equation 7).

$$w(u,v) = \frac{\sum_{sim_k \in Similarities} sim_k(u,v)}{|Similarities|} \quad (7)$$

- Rate Based setting: The rating scores are used in the range that is either given in the input data or calculated afterwards. The similarity values of neighbors are assigned to 1.0 as in the Base setting.
- Rate and Weight Based setting: Both the input rating scores and the calculated weights of neighbors are used.

B. Details of the Implementation

We used the Checkin2011[1] dataset [6] during implementation and testing. The data contains information about users' check-in, friendship and hometown. The check-in information is binary by definition, indicating whether a user checked-in in a location or not. In order to apply rate based calculations, we introduced check-in ratings to the system. This is detailed in subsection III-B1.

Using the input information, we first calculate several different similarity measures among users (Detailed in subsection III-B2). Then we use these similarities to decide on the non-dominated users as explained in the previous section. Besides, we implement the state-of-the-art methods to be able to compare our proposed method (Detailed in subsection III-B3).

1) Converting Binary Check-in Data to Rated Check-in Data: Analysis of the input data showed us that there are many locations where a user checked-in only once, but there are some others where users checked-in several more times. If a user checks-in at the same location very frequently, this shows that the user prefers that location more than a location where he/she checks in less frequent. So, we assigned rating scores to the check-ins depending on the number of check-ins for a location.

TABLE I. RATING ASSIGNMENT TO BINARY CHECKIN DATA

Range	Checkins	Rating score
x ≤ 1	64510	1.0
1 < x ≤ 5	17753	2.0
5 < x ≤ 11	2688	3.0
11 < x ≤ 23	1052	4.0
23 < x	372	5.0

We start by assigning 1.0 as the rating score to the check-ins which the user visits only once. Then we take the average of the check-in counts by discarding the one timers, and assigned 2.0 as the rating score. We continue in the same manner until we reach 5.0 as the rating score. Table I shows the related ranges, number of check-ins in those ranges and the assigned rating scores.

Looking at the table we observe that the number of check-ins which have 3.0 or more as ratings are less than the number of check-ins with ratings 1.0 and 2.0. So we decide to give rating 3.0 to all ratings which were previously assigned to 3.0 or more. We end up with a rating scale between 1.0-3.0, and thresholds 1 and 5.

2) Similarity Criteria: We used four different similarity criteria which are based on check-in, hometown, friendship and social influence features.

The first criterion is user-user similarity based on user check-in matrix. The assumption is that similar users prefer to check in at similar places. For user check-in based similarity calculations, we used Cosine similarity metric for the case of binary data, and Adjusted Cosine similarity metric for the case of rated data.

The second criterion is user-user similarity based on hometown. The assumption is that users from the same hometown prefer similar locations to check-in. This similarity is set to 1.0 if the users are from the same hometown, and set to 0.0 otherwise.

The third criterion is user-user similarity based on friendship. The assumption is that friends prefer check in at similar locations. This similarity is set to 1.0 if the users are friends; it is set to 0.0 otherwise.

[1] http://www.public.asu.edu/~hgao16/dataset.html

The last criterion is related to the influence of the users on each other. Given a social network, global influence of users can be calculated by several different algorithms. The most intuitive one is to use the count of connected friends, such that if many users are connected to a single user (count of inward edges is high), then that user is considered to have more influence on the social network. Another method to calculate influence is to use PageRank [18]. PageRank is proposed to decide on the importance of Web-pages and rank them. Similarly, finding out influencers is the process of finding important users/nodes in the social network and ranking them. However, the PageRank method needs global information of the input network, which leads to scalability problems in a large dataset. Instead, there is a need for influence analysis on local information.

In this work, we used a local influence model that uses friendship information only. We modelled influence of a user on the target user by finding out common friends. The idea is that if a user has many common friends with the target, this user will be able to influence the target more. For the influence calculation an Adjusted Cosine similarity metric on friendship data is used.

3) The Implemented Methods: Besides several different versions of the proposed multi-objective optimization based method, we implemented different versions of traditional collaborative filtering based methods:

- Collaborative Filtering on User x Check-in (CF-C): Traditional user-based collaborative filtering algorithm is used to give recommendations. The most similar users based on past check-in information are defined as neighbors.
- Collaborative Filtering on Friends(CF-F): Friends are defined as neighbors.
- Collaborative Filtering on Influence(CF-I): Users who have most influence on the target user are defined as neighbors.
- Collaborative Filtering on Hometown(CF-H): Users from the same hometown as the target user are defined as neighbors.
- Multi-Objective User Similarity: Multiple context and/or similarity measures are used to decide on non-dominated users. These users are considered as neighbors. We used different combination of contexts to observe their effects:
    - ○ Check-in & Influence (MO-CI): The criteria are check-in similarity and influence.
    - ○ Check-in & Friends & Influence (MO-CFI): The criteria are check-in similarity, friendship and influence.
    - ○ Check-in & Friends & Influence & Hometown (MO-CFIH): The criteria are check-in similarity, friendship, influence and hometown.
    - ○ Check-in & Hometown (MO-CH): The criteria are check-in similarity and hometown.

IV. Evaluation Results

The Checkin2011 dataset contains 11326 users, 187218 locations, 1385223 check-ins, and 47164 friendship links. The data was collected from the Foursquare web-site in between January 2011 and December 2011. Since the size of the data is large, we obtained a sub-sample of the data by extracting check-ins made in January 2011, which we named as CheckinsJan. In CheckinsJan data, there are 8308 users, 49521 locations and 86375 check-ins.

Using the CheckinsJan data, we aimed to recommend a list of checkin locations to the users that they will check-in during the next month, February. There are 7187 users who have checked-in both in January and February. In the evaluation step, we considered only those users. Before discussing the evaluation results in Section IV-C, we explain the evaluation metrics in Section IV-A and we cover the configuration in Section IV-B.

A. Evaluation Metrics

We used several different metrics, namely Precision, Ndcg (Normalized discounted cumulative gain), Hit rate and Coverage to analyze the performance of the implemented methods.

The Precision metric is presented in Equations 8. In the equation tp_k denotes true positives and fp_k denotes false positives in the given output list with size k. True positives are the ones which are recommended in the output list and are actually true, such that they are visited by the target user in the future. False positives are the recommendations listed in the output list, but are not actually visited in the future by the target. While giving the evaluation results, we presented the average of the $Prec_k$ values.

$$Prec_k = \frac{tp_k}{tp_k + fp_k} \tag{8}$$

The Ndcg (Normalized discounted cumulative gain) metrics decides on the relevance of the listed items depending on their rank. It is calculated by Equation 9. The Dcg (Discounted cumulative gain) value is calculated by Equation 10. In the equation k is size of the returned list and j is the item's position in the list. Idcg (Ideal discounted cumulative gain) is the Dcg value in the ideal case, where the resulting list is sorted by relevance. While giving the evaluation results, we presented the average of the $Ndcg_k$ values.

$$Ndcg_k = \frac{Dcg_k}{Idcg_k} \tag{9}$$

$$Dcg_k = rel_1 + \sum_{j=2}^{k} \frac{rel_j}{log_2 j} \tag{10}$$

The Hit rate metric shows the ratio of the users who are given at least one true recommendation. It is calculated by Equation 11. In the equation, M is the set of target users, and m is one of those users. $HitRate_m$ is a number whose value is set to 1.0 if the output list contains at least one true recommendation and to 0.0 otherwise.

$$HitRate = \frac{\sum_{m \in M} HitRate_m}{|M|} \tag{11}$$

User coverage is defined as the ratio of the users who are given any recommendation by the system. As stated in [4], some of the algorithms in the literature lose coverage in order to gain more accuracy. These algorithms usually suffer from giving poor recommendations to cold start users. In [8] it is stated that coverage and accuracy should be analyzed together.

B. Configurations

We aim to predict future check-ins of each user using the CheckinsJan data. In the prediction step, we need to limit the number of neighbors, N, and the output list size, k. In order to decide on N and k values, we perform the following analysis.

In order to decide on N, we fixed the value of k and performed the recommendation process using the Base setting. For the analysis, we set k in the range of 10-30 with 5 increments, and we set N in between 10-80 with 10 increments. Then, we analyzed the results in terms of precision and hit rate. Figures 3 and 4 show the results when k is set to 10 and Figures 5 and 6 show the results when it is set to 30. In all the figures, we observe that increasing N increases the precision and hit rate performance. However, the acceleration of the increase reduces after a certain value of N. This value can be assigned as N in the rest of the analysis. Using the outputs of our tests, we assigned N as 30, for this evaluation configuration.

After deciding on the value of N, the next step is to decide the value of k. For this purpose, we used N as 30 and gave k different values in the range 10-30 with 5 increments, as we did previously. The precision and hit rate results for this setting are given in Figures 7 and 8. From the results, we observe that the precision is affected by k significantly and reduces sharply as it increases. The hit rate increases, as expected, when the value of k increases. However, the ratio of the change is not as sharp as in the precision. Considering all these, we decided to assign the value of k as 10.

C. Evaluation Results

The upper-bounds of the related metrics are as follows: Ndcg, Hit rate and Coverage metrics' upper-bounds are 1.0. The upper-bound for the Precision metric is 0.489. For the upper bound calculations, only the check-in locations that are seen in the CheckinsJan data are considered. For all of the metrics, the users are also limited to the ones who check in both in January and February. The number of users with precision upper-bound values are given in Table II. Most of the users have 0.1 or 0.2 precision upper-bound values, which shows that these users check in only once or twice during February, i.e., during our test period. These users are the most challenging ones to give recommendation.

In Table III, the results for Base setting are given. According to the table, the best two methods are CF-F and MO-CH in terms of Precision, Ndcg and Hitrate. The best Coverage is obtained by the MO-CFIH method. We observed that in CheckinsJan data with Base settings, the check-in and the hometown features carry more information than the friendship and influence features. Combining all the features increases the coverage close to 1.0, to the upper bound. The lower precision value in the combination of the features, namely MO-CFIH, may depend on the coverage. This method (and the other multi-objective based methods) can give recommendations even to the cold start users. However, giving recommendation to these users is challenging and may lead to decrease in the precision.

TABLE II. NUMBER OF USERS WITH THE GIVEN PRECISION UPPER-BOUND VALUE

Upper-bound Prec.	No. of Users	Ratio of Users
0.1	1152	16.0%
0.2	1048	14.6%
0.3	877	12.2%
0.4	753	10.5%
0.5	634	8.8%
0.6	491	6.8%
0.7	421	5.9%
0.8	322	4.5%
0.9	262	3.6%
1.0	1227	17.1%

TABLE III. RESULTS FOR BASE SETTING

Method	Precision	Ndcg	HitRate	Coverage
CF-C	**0.183**	**0.242**	**0.621**	0.955
CF-F	0.138	0.064	0.212	0.810
CF-I	0.145	0.067	0.226	0.873
CF-H	0.156	0.132	0.435	0.965
MO-CI	0.179	0.213	0.572	0.993
MO-CFI	0.178	0.213	0.577	0.993
MO-CFIH	0.177	0.218	0.596	**0.999**
MO-CH	0.182	0.227	0.616	0.996

TABLE IV. RESULTS FOR WEIGHT BASED SETTING

Method	Precision	Ndcg	HitRate	Coverage
CF-C	**0.188**	**0.250**	**0.630**	0.955
CF-F	0.138	0.064	0.221	0.845
CF-I	0.147	0.068	0.229	0.873
CF-H	0.156	0.132	0.435	0.965
MO-CI	0.176	0.218	0.597	**0.999**
MO-CFI	0.183	0.225	0.607	0.996
MO-CFIH	0.178	0.207	0.549	0.993
MO-CH	0.169	0.193	0.522	0.993

In Table IV, the results for Weight Based setting are given. According to the table, the best two methods are CF-F and MO-CFI in terms of Precision, Ndcg and Hitrate. The best Coverage is obtained by the MO-CI method. Combining multiple features via multi-objective optimization methods leads to increase in the coverage more than 3%. Comparison with the Base setting results shows us that using weights of neighbors in the calculations increases the performance of the methods.

In Table V, the results for Rate Based setting are given. According to the table, the best Precision result belongs to CF-C. It is followed by the multi-objective optimization methods. In terms of Ndcg, Hitrate and Coverage, the best results are obtained by MO-CH and MO-CFIH. Introducing rates to the system decreases the performances of the methods which use check-in feature. We observe that multi-objective optimization methods are not affected as much as the other ones from the introduced rates, since they consider multiple features at once.

In Table VI, the results for Rate and Weight Based setting are given. According to the table, the best Precision result is obtained by CF-C. The best Ndcg, Hitrate and Coverage results belong to MO-CFIH. Combining multiple features via

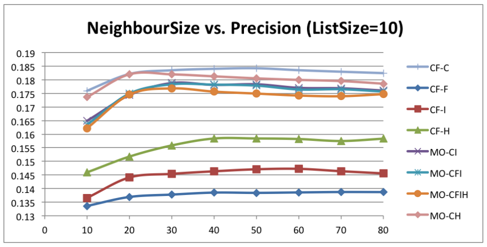


Fig. 3. N vs Precision when $k = 10$

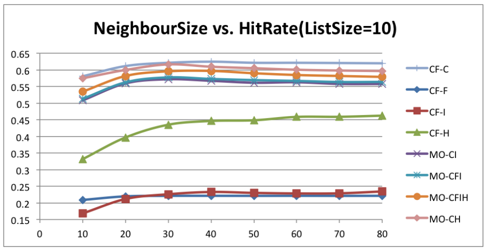


Fig. 4. N vs Hitrate when $k = 10$

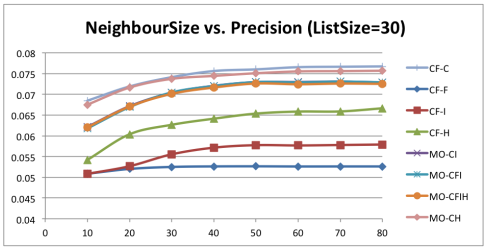


Fig. 5. N vs Precision when $k = 30$

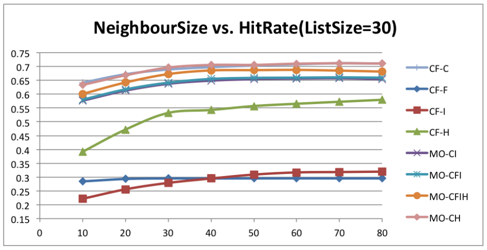


Fig. 6. N vs Hitrate when $k = 30$

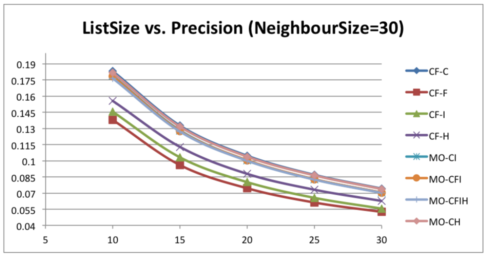


Fig. 7. k vs Precision when $N = 30$

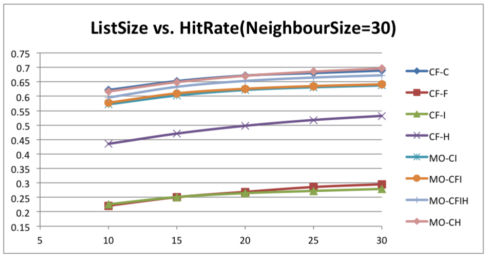


Fig. 8. k vs Hitrate when $N = 30$

TABLE V. RESULTS FOR RATE BASED SETTING

Method	Precision	Ndcg	HitRate	Coverage
CF-C	**0.188**	0.173	0.429	0.590
CF-F	0.141	0.074	0.241	0.845
CF-I	0.145	0.063	0.205	0.873
CF-H	0.156	0.136	0.439	0.965
MO-CI	0.171	0.165	0.440	0.948
MO-CFI	0.170	0.169	0.455	0.951
MO-CFIH	0.167	0.186	0.516	**0.997**
MO-CH	0.172	**0.190**	**0.537**	0.983

TABLE VI. RESULTS FOR RATE AND WEIGHT BASED SETTING

Method	Precision	Ndcg	HitRate	Coverage
CF-C	**0.190**	0.178	0.434	0.590
CF-F	0.141	0.074	0.241	0.845
CF-I	0.146	0.064	0.209	0.873
CF-H	0.156	0.136	0.439	0.965
MO-CI	0.173	0.174	0.450	0.948
MO-CFI	0.166	0.171	0.457	0.951
MO-CFIH	0.169	**0.196**	**0.540**	**0.997**
MO-CH	0.174	0.193	0.539	0.983

multi-objective optimization methods leads to high coverage performance, more than 95%. Comparing the results to the rate based setting, we observe that the performances of the methods that are using check-in information increases slightly.

In order to observe how different settings have affected the results, we presented Figures 9 to 16. These figures show the behavior of different methods on different settings. For the CF-C method, adding weight information slightly increases performance. However, introducing rating information to the system leads to sharp decrease, especially for the Hitrate and Coverage metrics. This shows that CF-C is highly affected by the range of the input rating. For the CF-F, CF-I and CF-H methods, the results seem not to be affected by the weight or rate information. For multi-objective optimization based methods, namely MO-CI, MO-CFI, MO-CFIH and MO-CH, the performance increases with weight usage and decreases with rate usage. Unlike CF-C, these methods are not affected sharply from the different settings. Also, these methods achieved nearly as good as Precision and Ndcg performance while achieving nearly 1.0 coverage, which is the upper-bound.

In summary, multi-objective optimization based methods use the information provided by different features. This leads to increase in coverage while preserving precision. Also, these methods are not affected by the rating range and have persistent results.

V. CONCLUSION

Recommendation systems suggest items to the user by estimating the ratings that user would give to them. In the literature, there are three main approaches to give recom-

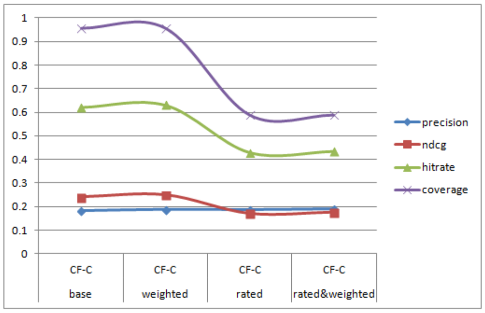


Fig. 9. Results for different settings: CF-C

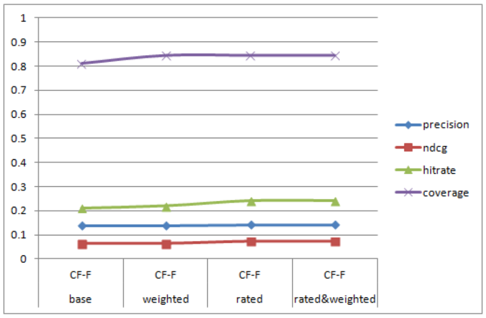


Fig. 10. Results for different settings: CF-F

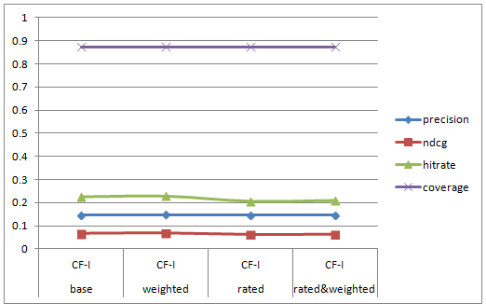


Fig. 11. Results for different settings: CF-I

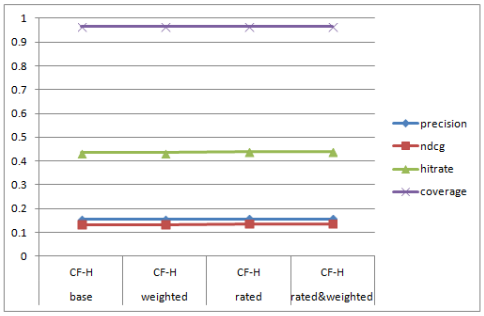


Fig. 12. Results for different settings: CF-H

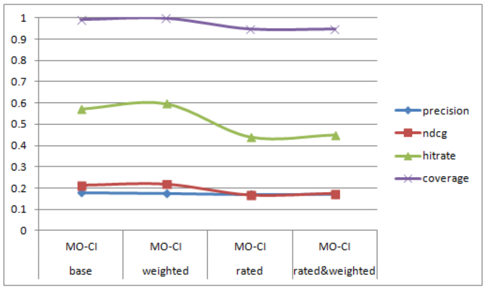


Fig. 13. Results for different settings: MO-CI

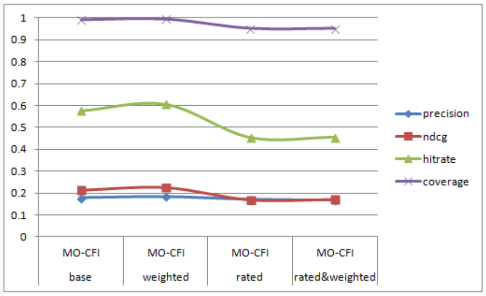


Fig. 14. Results for different settings: MO-CFI

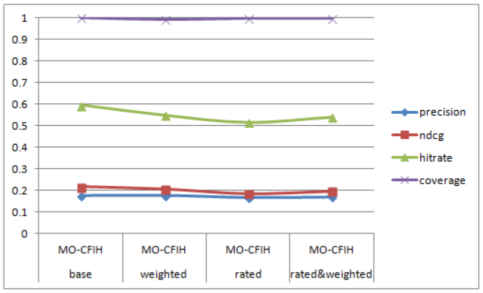


Fig. 15. Results for different settings: MO-CFIH

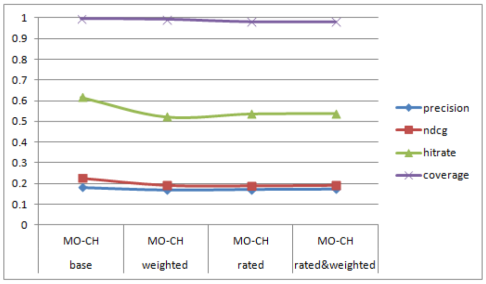


Fig. 16. Results for different settings: MO-CH

mendations, namely content based, collaborative filtering and hybrid approaches. Traditional recommender systems consider neither location nor social networks for users or items. Today many applications provide both location and social network information to the researchers. With the help of these applications, it is possible to embed both location and social network information into recommender systems. Most of the recommender systems are based on single criterion, such that they aim to evaluate the item based on overall rating. In order to give more accurate recommendations, a recommender system can take advantage of considering multiple criteria.

Motivated by the challenges of combining recommendation systems, LBSNs, and multi objective methods, a location and social network aware recommender system is proposed in this paper. The proposed method is used for giving check-in recommendations, using check-in, hometown, friendship and influence information. Also, we introduce different settings, namely base, weight based, rate based and rate and weight based, to predict the future check-in locations of the target users. Combination of different features leads to increase in coverage while preserving precision. Also, it is not affected by the rating range, which leads to persistent results.

In the future we want to add several different features to the system to increase the performance. Besides we want to consider clustering techniques in combination with the proposed multi-objective based method.

Acknowledgement

This work is supported by TÜBİTAK-BİDEB 2214/A program.

References

[1] G. Adomavicius and Y. Kwon, "New recommendation techniques for multicriteria rating systems," *IEEE Intelligent Systems*, vol. 22, no. 3, pp. 48–55, May 2007.

[2] G. Adomavicius, N. Manouselis, and Y. Kwon, "Multi-criteria recommender systems," in *Recommender Systems Handbook*, 2011, pp. 769–803.

[3] G. Adomavicius and A. Tuzhilin, "Towards the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734–749, 2005.

[4] A. Bellogín, I. Cantador, F. Díez, P. Castells, and E. Chavarriaga, "An empirical comparison of social, collaborative filtering, and hybrid recommenders," *ACM Trans. Intell. Syst. Technol.*, vol. 4, no. 1, pp. 14:1–14:29, Feb. 2013. [Online]. Available: http://doi.acm.org/10.1145/2414425.2414439

[5] C. Cheng, H. Yang, M. R. Lyu, and I. King, "Where you like to go next: Successive point-of-interest recommendation," in *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, ser. IJCAI'13. AAAI Press, 2013, pp. 2605–2611.

[6] H. Gao and H. Liu, "Location-based social network data repository," 2014. [Online]. Available: http://www.public.asu.edu/~hgao16/dataset.html

[7] H. Gao, J. Tang, and H. Liu, "Exploring social-historical ties on location-based social networks," in *ICWSM*, J. G. Breslin, N. B. Ellison, J. G. Shanahan, and Z. Tufekci, Eds. The AAAI Press, 2012.

[8] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 5–53, Jan. 2004. [Online]. Available: http://doi.acm.org/10.1145/963770.963772

[9] K. Lakiotaki, S. Tsafarakis, and N. Matsatsinis, "Uta-rec: A recommender system based on multiple criteria analysis," in *Proceedings of the 2008 ACM Conference on Recommender Systems*, ser. RecSys '08. New York, NY, USA: ACM, 2008, pp. 219–226.

[10] H.-H. Lee and W.-G. Teng, "Incorporating multi-criteria ratings in recommendation systems." in *IRI*. IEEE Systems, Man, and Cybernetics Society, 2007, pp. 273–278.

[11] J. J. Levandoski, M. Sarwat, A. Eldawy, and M. F. Mokbel, "Lars: A location-aware recommender system," in *Proceedings of the 2012 IEEE 28th International Conference on Data Engineering*, ser. ICDE '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 450–461.

[12] X. Liu and K. Aberer, "Soco: A social network aided context-aware recommender system," in *Proceedings of the 22Nd International Conference on World Wide Web*, ser. WWW '13. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2013, pp. 781–802. [Online]. Available: http://dl.acm.org/citation.cfm?id=2488388.2488457

[13] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender systems with social regularization," in *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, ser. WSDM '11. New York, NY, USA: ACM, 2011, pp. 287–296.

[14] N. Manouselis and C. Costopoulou, "Analysis and classification of multi-criteria recommender systems," *World Wide Web*, vol. 10, no. 4, pp. 415–441, Dec. 2007.

[15] N. Manouselis and C. Costopoulou, "Experimental analysis of design choices in multiattribute utility collaborative filtering." *IJPRAI*, vol. 21, no. 2, pp. 311–331, 2007.

[16] P. Massa and P. Avesani, "Trust-aware recommender systems," in *RecSys*, 2007, pp. 17–24.

[17] S. Naseri, A. Bahrehmand, C. Ding, and C.-H. Chi, "Enhancing tag-based collaborative filtering via integrated social networking information," in *ASONAM*, 2013.

[18] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web," in *Proceedings of the 7th International World Wide Web Conference*, Brisbane, Australia, 1998, pp. 161–172. [Online]. Available: citeseer.nj.nec.com/page98pagerank.html

[19] M.-H. Park, J.-H. Hong, and S.-B. Cho, "Location-based recommendation system using bayesian user's preference model in mobile devices," in *Proceedings of the 4th International Conference on Ubiquitous Intelligence and Computing*, ser. UIC'07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 1130–1139.

[20] M. Sarwat, J. J. Levandoski, A. Eldawy, and M. F. Mokbel, "Lars*: An efficient and scalable location-aware recommender system," *IEEE Transactions on Knowledge and Data Engineering*, vol. 99, no. PrePrints, p. 1, 2013.

[21] M. Sattari, "Hybrid geo-activity recommendation system using advanced feature combination and semantic activity similarity," Master's thesis, Middle East Technical University, Ankara, Turkey, 2013.

[22] M. Tavakolifard and K. C. Almeroth, "Social computing: an intersection of recommender systems, trust/reputation systems, and social networks," *IEEE Network*, vol. 26, no. 4, pp. 53–58, 2012.

[23] M. Ye, P. Yin, and W.-C. Lee, "Location recommendation for location-based social networks," in *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ser. GIS '10. New York, NY, USA: ACM, 2010, pp. 458–461.

[24] M. Ye, P. Yin, W.-C. Lee, and D.-L. Lee, "Exploiting geographical influence for collaborative point-of-interest recommendation," in *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '11. New York, NY, USA: ACM, 2011, pp. 325–334. [Online]. Available: http://doi.acm.org/10.1145/2009916.2009962

[25] V. W. Zheng, Y. Zheng, X. Xie, and Q. Yang, "Collaborative location and activity recommendations with gps history data," in *Proceedings of the 19th International Conference on World Wide Web*, ser. WWW '10. New York, NY, USA: ACM, 2010, pp. 1029–1038.