Time Preference aware Dynamic Recommendation Enhanced with Location, Social Network and Temporal Information

Makbule Gulcin Ozsoy and Faruk Polat
Department of Computer Engineering
Middle East Technical University,
Ankara, Turkey
Email: makbule.ozsoy, polat@ceng.metu.edu.tr

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

***Abstract*—Social networks and location based social networks have many active users who provide various kind of data, such as where they have been, who their friends are, which items they like more, when they go to a venue. Location, social network and temporal information provided by them can be used by recommendation systems to give more accurate suggestions. Also, recommendation systems can provide dynamic recommendations based on the users' preferences, such that they can give different recommendations for different hours of the day or different days of the week. In this paper, we propose a recommendation system which considers the users' temporal preference to give dynamic recommendation. The recommendation method uses multi-objective optimization approach and gives point of interest (POI) recommendation using several different criteria, namely past check-in locations, hometown of users, time of check-ins, friendship and influence among users.**

I. Introduction

Social networks (e.g.Twitter, Facebook, Foursquare) have many active users who provide various kind of data, such as where they have been, who their friends are, which items they like more. For example, up to January 2016, Twitter has 320 million monthly active users, Facebook has 1.55 billion monthly active users, Foursquare and Swarm have more than 50 million monthly active users. Nowadays, researchers use the social network data for different purposes, one of which is recommendation systems.

The traditional recommendation systems generally make use of overall rating, a single criterion, but do not utilize all information provided by the social networks. Location, social networks and time information in Location Based Social Networks (LBSNs) can be effectively incorporated in recommendation systems to derive much accurate recommendations. Furthermore, performance of recommendation systems can be improved by considering user preferences. In this work, we focused on giving point of interest (POI) recommendations using LBSN data by considering two folds of time information usage: considering check-in times as an additional criterion to decide the neighbors to the target users and giving dynamic recommendations based on users' temporal preferences.

Firstly, we focused on using the temporal information available in the LBSN data to find out the best neighbors. Even if two users visit the same venue frequently, we should differentiate them if they visit that place in different time of the day or day of the week. For example, lets consider a scenario where target user is *userA*, who has many common check-ins with *userB*, but less common check-ins with *userC*. However, *userA* and *userC* usually check in the morning (e.g. morning person), and *userB* checks in at night (e.g. night owl). Traditional collaborative filtering techniques, not using any temporal information, will choose *userB* as the most similar neighbor. However, when we consider the time information, it is better to choose *userC*. In the literature there are few works, [22], [11], [23], that use temporal information in their recommendation method. None of them use all the available information, namely historical preferences, spatial, social and temporal information. In this work, we aim to combine all four kinds of information to give better POI recommendations.

Secondly, we focused on giving dynamic recommendations to a target user based on the given time preference of the user. Traditional recommendation systems generally produce the same recommendations for a target user given the data with or without temporal preference. However, it is shown in the literature that users' behavior differs depending on the hour of the day (daytime vs. night) and day of the week (weekdays vs. weekend)([11], [3], [4]). For example, for a breakfast a user may want to visit a venue which serves brunch on a weekend morning, but prefers to visit a coffee shop on a weekday morning. To our knowledge, in the literature, there are few works, [23], that consider target time of the recommendation and our work is one of the first recommendation systems that considers users' temporal preference and gives dynamic recommendations based on these preferences.

The works in the literature which aim to combine a subset of the above-mentioned kinds of information; namely historical preferences, spatial, social and temporal information; usually use a linear fusion model or Gaussian mixture models. In this work, we combine all four kinds of data using a multi-objective optimization setting, by extending our previous work [16]. In [16], past preference of users, location and social network information are already used and we extend that work

IEEE/ACM ASONAM 2016, August 18-21, 2016, San Francisco, CA, USA 978-1-5090-2846-7/16/$31.00 2016 IEEE

by considering temporal information and introducing dynamic recommendation. We compare performance of the new method with those of traditional collaborative filtering based methods and our previous work.

Before giving information on related work and explaining the proposed methodology, in subsection I-A we explain our motivation on why we propose to use time information in a multi-objective optimization setting and why we aim to give dynamic recommendations. In subsection I-B, the organization of the rest of this paper is presented.

A. *Motivation*

Motivated by the analysis on human behavior in the literature and our intuition, we aim to add temporal information to the recommendation process. As mentioned in related work section, it has been shown that humans tend to behave differently depending on the time of the day (daytime vs. night) or day of the week (weekdays vs. weekends) ([11], [14], [3], [4]).

Similar to the analysis results, our intuition is that we should differentiate users who check in at similar locations at similar times than users who check in at similar locations at different times, as well as users who check in at different locations. We give Figure 1 as an example. In the figure, there are four users, namely $u1$, $u2$, $u3$ and $u4$, and two criteria, $c1$ and $c2$, which are the location and time of the check-ins, respectively. The locations are represented by their ids, e.g. $L1$, and temporal information is given in terms time of the day, namely daytime (D) and night (N). In the example, the target user, to whom we want to make recommendations, is $u1$. Just by looking at the input data, we observe that $u3$ and $u4$ have visited the similar places as the target user, $u1$. However, we observe that $u3$ and $u1$ have been at the same place at the same time more often than $u4$ and $u1$. From these observations we can conclude that $u3$ is the most similar user (the neighbor) for the target user $u1$.

u1					
Check-in Loc.	L1	L2	L 3	L 4	L 5
Check-in Time	D	N	N	N	D

u2				
Check-in Loc.	L 3	L 4	L 5	L 6
Check-in Time	D	D	D	D

u3						
Check-in Loc.	L 1	L 2	L 3	L 4	L 5	L 6
Check-in Time	N	N	D	N	D	D

u4						
Check-in Loc.	L1	L2	L3	L4	L5	L6
Check-in Time	N	D	D	D	D	D

Fig. 1. Example check-in and time information of users

Different approaches using temporal information behave differently while choosing the neighbors and giving recommendations. The following figures show how the similarities are calculated and used to decide on the best neighbor for the target user, $u1$, based on our example given in the Figure 1.

The first method is a collaborative filtering based recommendation method which only takes into account the past preferences (the check-in venues). In Figure 2, the check-in based matrix is created and the similarities among users are calculated. Based on the similarities, the users who are most similar to the target user are $u3$ or $u4$, and one of them is selected as the best neighbor.

Check-in Similarities Based

Users vs. Check-in Loc.	**L1**	**L2**	**L3**	**L4**	**L5**	**L6**
u1	1	1	1	1	1	0
u2	0	0	1	1	1	1
u3	1	1	1	1	1	1
u4	1	1	1	1	1	1

Size: |U| x $|C_1|$

Similarities to
the target user (u1):
u1-u2: 0.670
u1-u3: 0.912
u1-u4: 0.912

u3 = u4 > u2

Fig. 2. Neighbor selection: Check-in similarity based

The second method is a collaborative filtering based recommendation method which only takes into account the temporal information of the check-ins. In Figure 3, the time based matrix is created and the similarities among users are calculated. Based on the similarity values, $u3$ is selected as the neighbor as he/she has the highest similarity to the target user.

Temporal Similarities Based

Users vs. Time	**D**	**N**
u1	2	3
u2	4	0
u3	3	3
u4	5	1

Size: |U| x $|C_2|$

Similarities to
the target user (u1):
u1-u2: 0.554
u1-u3: 0.980
u1-u4: 0.707

u3 > u4 > u2

Fig. 3. Neighbor selection: Time similarity based

The third method is a collaborative filtering based recommendation method which takes into account both the location and the temporal information of the check-ins. In Figure 4, both check-in and time are considered together as a combination and the related matrix is created. Based on the matrix, the similarities among users are calculated and based on the results $u3$ is selected as the neighbor. This approach is not applicable since the size of the combination and the number of users are large in the real world data.

The fourth method uses the proposed multi-objective optimization based recommendation method and takes into account both the location and the temporal information of the check-ins. In Figure 5, the similarities among users based on location and temporal information are given on the left and the related dominance matrix is given on the right. The similarities are calculated as shown in the Figures 2 and 3. The details on

Check-in & Temporal (Combination) Similarities Based

Users vs. Check-in Loc. & Time	L1 D	L1 N	L2 D	L2 N	L3 D	L3 N	L4 D	L4 N	L5 D	L5 N	L6 D	L6 N
u1	1	0	0	1	0	1	0	1	1	0	0	0
u2	0	0	0	0	1	0	1	0	1	0	1	0
u3	0	1	0	1	1	0	0	1	1	0	1	0
u4	0	1	1	0	1	0	1	0	1	0	1	0

Size: $|U| \times (|C_1| \cdot |C_2|)$

Similarities to the target user (u1):
u1-u2: 0.223
u1-u3: 0.547
u1-u4: 0.182

u3 > u2 > u4

Fig. 4. Neighbor selection: Check-in&Time similarity based

how dominance matrix is created is not given in the figure, but it is explained in the Section III-B2. In this method, the non-dominated user is selected as the neighbor and for this example it is $u3$.

Multi-Objective Optimization Based

Users vs. Criteria	Check-in Loc Similarity	Temporal Similarity
u2	0.670	0.554
u3	0.912	0.980
u4	0.912	0.707

Size: $(|U|-1) \times |C| + |U| \times (|C_1| + |C_2|)$

Dominance Matrix	u2	u3	u4
u2	0	0	0
u3	1	0	1
u4	1	0	0
Total:	2	**0**	1

Fig. 5. Neighbor selection: Multi-objective optimization based

Previous works in the literature, our intuition and the example analyzed in this section show that temporal information provides information on the users behavior and can increase the performance of the recommendation system, by assisting to choose neighbors more effectively. While using the temporal information, other important features, such as social network and geographical information which are already shown to be effective in recommendation performance, should not be discarded. In order to combine all these kinds of information effectively, we believe that the multi-objective optimization based approach is a good choice.

Beside considering temporal information as a source for similarity calculations, we aim to use this information as a tool for giving dynamic recommendations. A user may ask the recommendation system to make recommendation for a specific hour of a day or day of a week. For example, to have a breakfast, a user may prefer to visit a venue which serves brunch on a weekend morning, but prefers to visit a coffee shop on a weekday morning. Also a user may require a recommendation independent from the current time (the time of asking for the recommendation). For example, on a weekday evening a user may look for brunch locations for the weekend. Traditional recommendation systems generally produce the same recommendations for a target user even if the user indicates his/her preference of time. Few recent works in the literature use the users' temporal preferences in recommendation process and our work is one of the first of these systems.

B. Organization

The rest of this paper is organized as follows: Literature review of the recommendation systems that use location, social network and/or temporal information or multi-objective optimization methods is given in Section II. The proposed recommendation method is explained in Section III. In Section IV, the evaluation settings and the results are given. The paper is concluded in the Section V.

II. Related Work

In the recommendation systems literature, there are some works that combine historical preferences of users, location, social network or time information. However, to our knowledge, none of them has combined all of the above-mentioned information. Besides, only few of them aim to give dynamic recommendations based on users' preferences.

Analysis on LBSN data shows that users tend to visit locations periodically [3] and that their behavior differs depending on the hour of the day (daytime vs. night) and day of the week (weekdays vs. weekend) ([11], [3], [4]). [22] aims to recommend POIs to the target user by incorporating the temporal information. They perform temporal analysis by splitting the time into hourly slots. They consider the defined check-in time slots to find out the user similarities. [6] studies on temporal effects on LBSNs and models user mobile behavior. They propose a framework that analyses and models the temporal cyclic patterns of human behavior (temporal preferences) and their relationship with spatial and social data (temporal correlations). They conclude that temporal and spatial information complements each other and improves the location prediction performance. [11] gives category aware point-of-interest (POI) recommendation using matrix factorization (MF) technique. In their method they divide the time into four by considering working hours and leisure time of a day and weekdays and weekends of a week. Their approach first models the preference transition of users in terms of categories by taking into account both category information and temporal effects. Then they predict the preferred categories of the target user and recommend the locations in the predicted categories. [23] gives time-aware POI recommendations by considering both geographical and temporal influences. They first create a graph model for check-ins, locations and time information and then use this graph model to propagate the preferences with their proposed algorithm. In their approach they split time into hourly slots. [19] gives route suggestions by considering spatial and temporal behavior of the visitors of a theme park. After creating sequence of Location-Item-Time (LIT) information based on the visitors' behavior analysis, they mine the frequent LIT sequences by their proposed algorithm. [24] proposes a method that uses Temporal Influence Correlations (TIC) to make time-aware recommendations and to recommend time-to-visit that location. It combines user-based and item-based (location-based) correlations. [27] learns context similarities to make context-aware recommendation, where one of the contexts can be time information. The idea of the proposed method is that the simialrity among contextual situations should produce similar recommendation lists. In this work, various similarity calculation methods are used, such as Independent Context Similarity (ICS), Latent Context Similarity (LCS), Weighted Jaccard Context Similarity (WJCS) and Multidimensional Context Similarity (MCS). [25] aims to recommend the successive locations to the user based on the user's current location. For this purprose they proposed a new model called location and time aware social collaborative retrieval model (LTSCR). In order to make recommendation, it combines the current location of the target user, the friendship relations among users and the time information. [13] aims to predict next POI of a tourist based on his/her past preferences by employing supervised learning techniques. Their feature set is composed of 68 features which are designed for tourism-related data. They included time-based features to model the behavior of tourists about how they spend their available time and when they visit the POIs. The experimental results show that the proposed method outperforms the state-of-the-art methods in recommendation and trail prediction for tourism.

Besides temporal information, location and social network information are used by many recommendation methods. Based on the analysis, [9] states that recommendations should be given by users who are living in the same or similar regions and recommended items should be close to the target user. LARS [9], [26], [21] and [2] are some example recommendation systems that uses location to improve their recommendation performances. [10] states that use of friendship relationships among users, data sparsity problem can be handled more efficiently and quality of recommendations can be increased. [7], SoCo [10], [20] and [12] are some example works that use social relations in the recommendation process. In order to combine multiple criteria in recommendation process, multi-objective optimization methods can also be used. [15] and [16] are some example works that use Pareto optimal points method to decide on the most representative neighbors. Our previous work, [16], uses Pareto dominance to find most representative neighbors to the target user in a location and social network aware setting. In that work we consider not only rating (check-in) but also location(hometown) and friendship information and show that the method preserves precision while increasing coverage. We also adopted the work proposed in [16] to other problems, i.e. prediction of the structure of gene regulatory networks (GRNs) [17], using multiple social networks to model user and and to make recommendations[18].

III. Time Preference aware Dynamic Recommendation Enhanced with Location, Social Network and Temporal Information

Nowadays, recommendation algorithms consider not only users, items and ratings, but also location, friendship, social network and temporal information. Most of the research using multiple criteria aggregate these criteria into a utility function by getting the weighted sum. Unlike them, [16] combines the criteria using a multi-objective optimization method. However, it considers only historical preferences of users, location and social network information. Also, to our knowledge, only few works in the literature aim to give different recommendations using the preference of each user. In this work, first, we add temporal information to [16] to get into account the time based preferences of users. Then we take the target users' temporal preference into account to make dynamic recommendations.

In Section III-A, information on how we model the temporal information is given. In Section III-B, the proposed recommendation system is explained and the implementation details based on the characteristics of the input data is given. Lastly, in Section III-C, the methods that are implemented in this work are summarized.

A. Modelling Temporal Information

In our work we divided time into eight different slots, such as the combination of four partitions of the day (i.e. morning, afternoon, evening, night) and two partitions of the week (i.e. weekdays and weekend). The weekends are assigned as Saturday and Sunday, and weekdays as the rest of the week. We assigned the hours in between 06.00 - 11.59 (6.00am - 11.59am) as morning, 12.00 - 17.59 (12.00pm - 5.59pm) as afternoon, 18.00 - 23.59 (6.00pm - 11.59pm) as evening and lastly 00.00 - 05.59 (00.00am - 5.59am) as night.

Using these time slots we believe that we can differentiate users who socialize in different times of a day and a week; such as in the morning (i.e. morning person) or at the night (i.e. night owl). For this purpose the temporal preference based similarity among users is used. Also in the proposed system, the time slots can be used by the target users to indicate their temporal preferences to get recommendations. As a result, our proposed method will give recommendations of venues that can be visited specifically in the given time slot. The process of decision of the neighbors and how they are used to give dynamic recommendations are detailed in the Section III-B.

B. The Proposed Method

The proposed system is composed of the following steps: similarity calculations, neighbor selection and item selection. These steps are the same as the ones presented in [16]. However, there are some differences in the similarity calculations and item selection step. The details of the steps and the differences between [16] and this work are as follows::

1) Similarity calculations: In the similarity calculations step, user-user similarities are calculated based on several different criteria. The choice of the criteria to be used is dependent on the data characteristics. In this work, we used the same data as in [16], namely Checkin2011 dataset [5]. The data contains users' check-ins, friendship and hometowns. The check-in information contains time-stamps as well as the location ids and its longitude-latitude information. Using this dataset, we calculated the following similarities: Check-in, hometown, friendship, influence and check-in time.

- Check-in: The assumption is that similar users prefer to check in at similar places. For this calculation, we use Cosine similarity metric.
- 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.

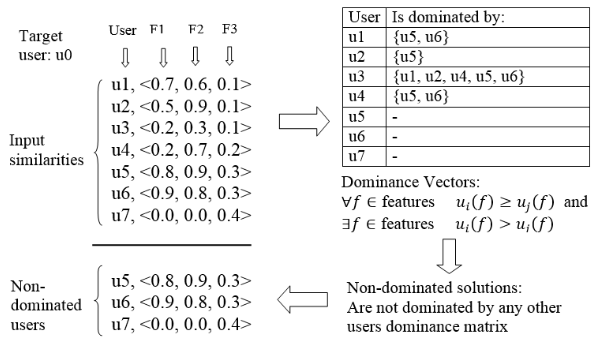

Fig. 6. Example input and non-dominated solutions

- Friendship: The assumption is that friends prefer check in at similar locations. This similarity is set to 1.0 if the users are friends, and set to 0.0 otherwise.
- Influence: We use the same local influence model explained in [16]. The influence model uses friendship information only and 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 calculations Cosine similarity metric on friendship data is used.
- Check-in time: This similarity measure is not used in [16] and added as a new feature in this work. We map the time of previous check-ins into eight different categories, which are the combination of four partitions of the day (i.e. morning, afternoon, evening, night) and two partitions of the week (i.e. weekdays and weekend). After mapping the check-in times of the users to the time slots, the similarities of users based on their frequency of check-ins in the related time slot is calculated. For the similarity calculations Cosine similarity metric is used.

2) Neighbor selection: [16] explains that the neighbors should be the ones who will affect the target most and the non-dominated users are the ones that have this property. The non-dominated users are obtained by finding the Pareto optimal points using the calculated similarities in the previous step. In Figure 6, an example of multi-dimensional data is given. In this example, the similarity values of seven users to the target user, $u0$, are given for three different criteria, namely F_1-F_3. To make the example more concrete, one can assume that these similarities are check-in, hometown and friendship similarity, which are calculated in the previous step.

Non-dominated users are founded by determining users who dominate some other users and then selecting the users who are never been dominated by any other. As the first step, a dominance matrix, as in [16], or inverted vector representation of dominance, as in the Figure 6, can be created. In the inverted vector representation of dominance, the dominated users and who dominates them are collected. The equation 1 is used to decide who dominates who. In the equations f indicates the features used, e.g. F_1, F_2 and F_3. Comparing the scores of users based on that feature, the dominance is decided. User au dominates user v, if u's scores (similarities in our example) are greater than or equal to those of v's, and there exists at least one feature score of u that is greater than that of v's. The second step is selecting the non-dominated users. They are the users who have an empty dominance vector. In the example, $u5$, $u6$ and $u7$ are selected as the non-dominated users.

$$dom(u,v) = \begin{cases} 1.0 & \forall f u(f) \geq v(f) \text{ and } \exists f u(f) > v(f) \\ 0.0 & \text{otherwise} \end{cases} \tag{1}$$

Neighbor selection step can be terminated in one iteration or in multiple iterations. If it is terminated in a single step, the neighbors' count can be less than expected. In [16] it is observed that having less neighbors does not produce good results for recommendation. To solve this issue, they introduced an iterative process that collects pre-defined number of neighbors. Following their approach, we implemented the iterative process which collects as many neighbors as predefined.

3) Item selection: In this step, the recommendation of items are given using the neighbors' previous preferences. The items previously preferred by the neighbors are considered as candidates. The scores of the items are calculated such that the more neighbors recommend an item, the more the score of the candidate item is. Also in this step, the target user's time category preference is taken into account. If the candidate item has never been visited in the given time slot by any of the neighbors, then it is removed from the candidate list. At the end, *top-k* items with the highest score in the requested time slot are suggested to the user.

The score calculation is performed according to the Equation 2. In the equation, $s(u,i)$ is the preference score of the target user u for item i. v is a user who is chosen as a neighbor to the user u. $sim(u,v)$ is the similarity of users u and v and $rat(v,i)$ is the rating given to the item i by the user v. In this work, the selected neighbors are considered to have the same level of effect on the target user, such that the similarity value of each neighbor is assigned to 1.0. Besides, rating score is also considered to be binary and its value is assigned to 1.0.

$$s(u,i) = \sum_{v \in Neighbors} sim(u,v) * rat(v,i) \tag{2}$$

C. Methods

Several different methods that use temporal information together with different combination of features are implemented. We implemented not only non-dynamic versions of the methods but also the dynamic versions. The dynamic versions are based on time category preference of the target users. While presenting the evaluation results, the methods with dynamic recommendation process are initialized with Dynamic Temporal Preference (DTP). The methods that use collaborative filtering or multi-objective optimization are abbreviated by CF or MO, respectively. The criteria that are used are abbreviated as follows: Checkin (C), Friends (F), Influence (I), Hometown (H), Time (T).

IV. Evaluation Results

We use the same dataset as in [16], namely CheckinsJan data which is a subset of Checkin2011 dataset [5]. The Checkin2011 dataset is collected from Foursquare web-site in between January 2011 - December 2011. It contains 11326 users, 187218 locations, 1385223 check-ins and 47164 friendship links. The subset CheckinsJan data is in range January and February, where January data is used as training set and February as test set. In CheckinsJan data, there are 8308 users, 49521 locations and 86375 check-ins. Our aim is to give a list of venues or point of interests (POIs) as recommendations using the CheckinsJan data, as previously done in [16].

We explain the evaluation metrics in Section IV-A and the evaluation results in Section IV-B

A. Evaluation Metrics

We used the same metrics *Precision*, *Ndcg*, *Hitrate* and *Coverage*, to analyze the performance of the implemented methods.

The Precision metric measures the ratio of true predictions, tp_k, among all the predictions, $tp_k + fp_k$. It is calculated by Equations 3. The tp_k represents true positives and fp_k represents false positives in the given output list with size k. In the evaluation results section, the overall average of Precision results is given.

$$Prec_k = \frac{tp_k}{tp_k + fp_k} \tag{3}$$

The Ndcg (Normalized discounted cumulative gain) metric also measures the truthfulness of the prediction, but it also takes the rankings of the true predictions into account. For example, if a true prediction is ranked upper in a list, its Ndcg value becomes larger. It is calculated by $Ndcg_k = \frac{Dcg_k}{Idcg_k}$ and the Dcg (Discounted cumulative gain) value is calculated by Equation 4. In the equation, k is the size of the returned list and j is the item's position in the list. The Idcg (Ideal discounted cumulative gain) is the Dcg value in the ideal case. In the evaluation results section, the overall average of Ndcg results is given.

$$Dcg_k = rel_1 + \sum_{j=2}^{k} \frac{rel_j}{log_2 j} \tag{4}$$

Precision@k value is dependent on the list size. For example, assume that there is only one true prediction in our output list. If the output list size, k, is 10, the Precision will be 0.1. If k is 2, the Precision will be 0.5. In order to have an independent metric from output list size, we use the Hitrate metric. It shows the ratio of the users who are given at least one true recommendation. It is calculated by Equation 5. In the equation, M is the set of target users, and m is one of those users. $HitRate_m$ is set to 1.0 if the output list contains at least one true recommendation and to 0.0 otherwise.

$$HitRate = \frac{\sum_{m \in M} HitRate_m}{|M|} \tag{5}$$

TABLE I
Results for non-dynamic methods using time similarity and its comparison to other methods

Method	Prec.	Ndcg	HitRate	Covrg.
CF-C	**0.114**	**0.242**	**0.621**	0.955
CF-F	0.030	0.064	0.221	0.845
CF-I	0.033	0.067	0.226	0.873
CF-H	0.068	0.132	0.435	0.965
CF-T	0.012	0.019	0.096	**1.000**
MO-CI	0.102	0.213	0.572	0.993
MO-CFI	0.103	0.213	0.577	0.993
MO-CFIH	0.105	0.218	0.596	**0.999**
MO-CH	**0.112**	**0.227**	**0.616**	0.996
MO-CT	0.105	0.213	0.576	**1.000**
MO-CHT	0.107	0.220	0.599	**1.000**
MO-CHFT	**0.108**	**0.221**	0.603	**1.000**
MO-CHFIT	0.107	**0.221**	**0.608**	**1.000**

User coverage is the ratio of the users who are given any recommendation by the system. Some of the algorithms in the literature loose coverage in order to gain more accuracy [1]. For example, they do not give any recommendation to challenging users, such as cold start users, at all. In [8] it is stated that coverage and accuracy should be analyzed together.

B. Evaluation Results

In this section, the performance results of the proposed recommendation systems that use temporal similarity in non-dynamic and dynamic settings are presented. The necessary parameters that need to be pre-defined; namely the number of neighbors, N, and the output list size, k, are assigned to the same values that are used in [16], such that $N = 30$ and $k = 10$.

1) Performance Results for Non-Dynamic Methods: We give performance results of the proposed method, which uses temporal information as well as other features, in different settings. The recommendations in this section do not consider the dynamic nature of user preferences. As in the previous experiments, only the locations that are seen in the CheckinsJan data and the users who checked in both in training and test periods are taken into account. The upper-bounds of the performance metrics are as follows: for Ndcg, Hitrate and Coverage, the upper-bound is 1.0. The upper-bound for Precision metric is 0.489.

In Table I, the results for recommendation methods that use time similarity as a criterion are given. According to the table, using only temporal similarity leads to very low performance results for Precision, Ndcg and Hitrate. However, for Coverage it is very informative and makes the recommendation system to be able to give recommendations to any user. Adding historical check-in information to the time information provides a huge jump in all metrics. The methods that perform best are the ones that uses check-in, hometown, friendship, influence and time information. All of the methods which use temporal information can give recommendation to any user, so the Coverage performance is 1.0.

In Table, I, we presented the comparison of these methods to others in order to observe how time criteria affected the performance of the recommendation system. In the table, the

TABLE II
UPPER BOUND OF THE METRICS BASED ON THE GIVEN TEMPORAL PREFERENCE

Temp.	# Users	Precision
WE_M	1526	0.208
WE_A	2803	0.217
WE_E	3914	0.246
WE_N	3297	0.238
WD_M	1994	0.297
WD_A	3902	0.338
WD_E	4756	0.343
WD_N	4117	0.318

first group gives the results of the traditional collaborative filtering based recommendation systems which use a single criterion. The results of this group shows that using check-in information provides better performance in terms of Precision, Ndcg and Hitrate. However, use of temporal information has the ability of covering all of the users. These results show that the use of temporal information together with historical check-in information is promising. The second group in the table includes the results of multi-objective optimization based methods presented in [16]. These methods use combination of historical check-ins, hometown of users, friendship and influence relation among users, but not temporal check-in similarity. These results show that combining multiple criteria together increases the performance, especially for the ones that use friendship only or influence only cases. The last group in the table belongs to the multi-objective optimization based method using temporal information. We observe that use of time information do not always increase the performance. For example, while there is about 1.2% increase in Hitrate performance comparing the methods MO-CFIH and MO-CHFIT, there is about 0.8% decrease in Hitrate performance comparing the methods MO-CH and MO-CHT. However, the use of temporal similarity is always useful to cover/to make recommendation to all of the users.

2) Performance Results for Dynamic Methods: In this section, we give performance results of the proposed method, which uses all the available criteria and gives dynamic recommendations based on the target user's temporal preferences. Similar to the previous results, we only considered the check-in locations that are seen in the CheckinsJan data. Also, we limit the users to the ones who checked in the given temporal preference slot during the test interval. For example, we did not take into account a user who asked for a recommendation for a weekend afternoon, but never checked in on that time slot during the test period.

The number of users and the related Precision upper bounds are given in the Table II, where WE and WD represent weekend and weekday, respectively, and M, A, E and N represent morning, afternoon, evening and night, respectively. From the table we observe that users tend to check in more on the weekdays and the number of users who check in increases from morning to afternoon and then decreases at the night. For the other metrics, namely Ndcg, HitRate and Coverage, the upper bounds are 1.0.

Table III show how temporal preference of users and dynamic recommendation affects the performance, The table is divided into three sections containing results of traditional methods using single criterion, multi-objective methods without time criterion and multi-objective methods with all the available criteria. The performance scores are the averages of the performance results calculated for each time category. According to the results, when we use a single criterion, the most effective methods are the ones that use check-in and hometown information. This observation follows the one made in [16] as well. We observe that including multiple criteria at once helps to increase coverage while preserving the performance for other metrics. When we compare these methods to the ones presented in Table I; non-dynamic methods; we observe that considering dynamicity while giving recommendation performs better in terms of Precision (note that for non-dynamic setting the Precision upper bound is 0.489 and for dynamic setting it is 0.276, on the average). However, the Ndcg and Hitrate results show that it is much harder to give at least one true recommendation to the target, when he/she indicates a temporal preference. This can be rooted from the fact that as we include more restriction on the recommendation, the available data becomes much sparser and the process of making recommendation becomes harder.

TABLE III
RESULTS FOR DYNAMIC METHODS USING TIME SIMILARITY

Method	Prec.	Ndcg	HitRate	Covrg.
DTP_CF-C	**0.025**	**0.106**	**0.217**	**0.961**
DTP_CF-F	0.006	0.025	0.055	0.708
DTP_CF-I	0.007	0.028	0.062	0.863
DTP_CF-H	0.015	0.060	0.135	0.947
DTP_CF-T	0.002	0.008	0.022	0.955
DTP_MO-CI	0.022	0.093	0.191	0.994
DTP_MO-CFI	0.022	0.094	0.193	0.994
DTP_MO-CFIH	0.023	0.096	0.200	**0.999**
DTP_MO-CH	**0.025**	**0.102**	**0.216**	0.994
DTP_MO-CT	0.024	0.097	0.203	0.975
DTP_MO-CHT	0.024	**0.100**	**0.211**	0.985
DTP_MO-CHFT	**0.025**	**0.100**	**0.211**	0.991
DTP_MO-CHFIT	0.024	**0.100**	0.208	**0.997**

V. CONCLUSION

Traditional recommendation systems consider neither location nor social networks nor time information. It is possible to improve the performance of recommendation systems with information from LBSNs. Furthermore, recommendation systems can provide dynamic recommendations based on the users preferences, such that they can make different recommendations for different hours of the day or different days of the week.

A new recommendation approach, that combines recommendation systems, LBSNs, multi-objective methods and gives dynamic recommendations, is proposed in this paper. To our knowledge, this is the first POI recommendation method that uses all types of information, namely check-in locations (historical preference information), hometown of users (geographical information), friendship and influence among users (social network information) and time of check-ins (temporal information), and one of the first recommendation systems that considers temporal preferences of users to give dynamic recommendations. We analyzed use of temporal information in both non-dynamic and dynamic settings. In non-dynamic setting, time category, such as weekend morning, of the check-ins are used to calculate the similarity of the users. Then, a multi-objective optimization based method is applied to make recommendations. In dynamic setting, the time category preference of the users are taken into account. This way, independent of the current time the users are able to ask for venue recommendation for any time category and the system can make different recommendations for different hours of the day or different days of the week. The evaluation results showed that use of temporal information increases the performance results when it is used together with other criteria. These systems has the ability of giving recommendation to any of the users. Besides, the evaluation results revealed that when dynamicity based on temporal preference is introduced to the system, it becomes more challenging to make recommendation, since data becomes sparser.

As a future work, we plan to add several different features, such as age and gender, to the system to increase the performance of recommendation. Also, we want to include user based profiles; such as having many friends/not, having many check-ins/not; to the system.

REFERENCES

[1] A. Bellogín, I. Cantador, F. Díez, P. Castells, and E. Chavarriaga, "An empirical comparison of social, collaborative filtering, and hybrid recommenders," *ACM TIST*, vol. 4, no. 1, p. 14, 2013.

[2] C. Cheng, H. Yang, M. R. Lyu, and I. King, "Where you like to go next: Successive point-of-interest recommendation," in *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, 2013.

[3] Z. Cheng, J. Caverlee, K. Lee, and D. Z. Sui, "Exploring millions of footprints in location sharing services." in *ICWSM*, L. A. Adamic, R. A. Baeza-Yates, and S. Counts, Eds. The AAAI Press, 2011.

[4] E. Cho, S. A. Myers, and J. Leskovec, "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*, ser. KDD '11. New York, NY, USA: ACM, 2011, pp. 1082–1090.

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

[6] H. Gao, J. Tang, X. Hu, and H. Liu, "Modeling temporal effects of human mobile behavior on location-based social networks," in *Proceedings of the 22Nd ACM International Conference on Conference on Information and Knowledge Management*, ser. CIKM '13. New York, NY, USA: ACM, 2013, pp. 1673–1678.

[7] H. Gao, J. Tang, and H. Liu, "Exploring social-historical ties on location-based social networks," in *Proceedings of the Sixth International Conference on Weblogs and Social Media, Dublin, Ireland, June 4-7, 2012*, 2012.

[8] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 5–53, 2004.

[9] J. J. Levandoski, M. Sarwat, A. Eldawy, and M. F. Mokbel, "LARS: A location-aware recommender system," in *IEEE 28th International Conference on Data Engineering (ICDE 2012), Washington, DC, USA (Arlington, Virginia), 1-5 April, 2012*, 2012, pp. 450–461.

[10] X. Liu and K. Aberer, "Soco: a social network aided context-aware recommender system," in *22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013*, 2013, pp. 781–802.

[11] X. Liu, Y. Liu, K. Aberer, and C. Miao, "Personalized point-of-interest recommendation by mining users' preference transition," in *Proceedings of the 22Nd ACM International Conference on Conference on Information Knowledge Management*, ser. CIKM '13. New York, NY, USA: ACM, 2013, pp. 733–738.

[12] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender systems with social regularization," in *Proceedings of the Forth International Conference on Web Search and Web Data Mining, WSDM 2011, Hong Kong, China, February 9-12, 2011*, 2011, pp. 287–296.

[13] C. I. Muntean, F. M. Nardini, F. Silvestri, and R. Baraglia, "On learning prediction models for tourists paths," *ACM Trans. Intell. Syst. Technol.*, vol. 7, no. 1, pp. 8:1–8:34, Oct. 2015.

[14] A. Noulas, S. Scellato, N. Lathia, and C. Mascolo, "Mining user mobility features for next place prediction in location-based services," in *Proceedings of the 2012 IEEE 12th International Conference on Data Mining*, ser. ICDM '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 1038–1043.

[15] F. Ortega, J. L. Sánchez, J. Bobadilla, and A. Gutiérrez, "Improving collaborative filtering-based recommender systems results using pareto dominance," *Inf. Sci.*, vol. 239, pp. 50–61, 2013.

[16] M. G. Ozsoy, F. Polat, and R. Alhajj, "Multi-objective optimization based location and social network aware recommendation," in *10th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing, CollaborateCom 2014, Miami, Florida, USA, October 22-25, 2014*, 2014, pp. 233–242.

[17] M. G. Ozsoy, F. Polat, and R. Alhajj, "Inference of gene regulatory networks via multiple data sources and a recommendation method," in *The IEEE International Conference on Bioinformatics and Biomedicine (BIBM) 2015*. IEEE, 2015, pp. 661–664.

[18] M. G. Ozsoy, F. Polat, and R. Alhajj, "Modeling individuals and making recommendations using multiple social networks," in *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*, ser. ASONAM '15. New York, NY, USA: ACM, 2015, pp. 1184–1191.

[19] C.-Y. Tsai and B.-H. Lai, "A location-item-time sequential pattern mining algorithm for route recommendation," *Knowledge-Based Systems*, vol. 73, pp. 97–110, 2015.

[20] 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.

[21] M. Ye, P. Yin, W. Lee, and D. L. Lee, "Exploiting geographical influence for collaborative point-of-interest recommendation," in *Proceeding of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2011, Beijing, China, July 25-29, 2011*, 2011, pp. 325–334.

[22] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. M. Thalmann, "Time-aware point-of-interest recommendation," in *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '13. New York, NY, USA: ACM, 2013, pp. 363–372.

[23] Q. Yuan, G. Cong, and A. Sun, "Graph-based point-of-interest recommendation with geographical and temporal influences," in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. ACM, 2014, pp. 659–668.

[24] J. Zhang and C. Chow, "Ticrec: A probabilistic framework to utilize temporal influence correlations for time-aware location recommendations (to be published)," *Services Computing, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2015.

[25] W. Zhang and J. Wang, "Location and time aware social collaborative retrieval for new successive point-of-interest recommendation," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, ser. CIKM '15. New York, NY, USA: ACM, 2015, pp. 1221–1230.

[26] 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, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010*, 2010, pp. 1029–1038.

[27] Y. Zheng, B. Mobasher, and R. Burke, *User Modeling, Adaptation and Personalization: 23rd International Conference, UMAP 2015, Dublin, Ireland, June 29 – July 3, 2015. Proceedings*. Cham: Springer International Publishing, 2015, ch. Integrating Context Similarity with Sparse Linear Recommendation Model, pp. 370–376.