

Advances in Algorithms for Time-dependent Recommender Systems

Pavlos Kefalas
Department of Informatics
Aristotle University
54124 Thessaloniki, Greece
kefalasp@csd.auth.gr

Yannis Manolopoulos
Department of Informatics
Aristotle University
54124 Thessaloniki, Greece
manolopo@csd.auth.gr

Abstract— Nowadays, Online Social Networks have given the opportunity to users to share their interests. Moreover Location-Based Social Network added the location factor giving a new perspective to users' check-ins in POIs through smartphones. There are three main parameters characterizing these networks: mobility, proximity and periodicity. Here, we argue that periodicity is a significant upcoming trend in recommender systems. In particular, we present an extended comparison among 9 recommendation frameworks and their structural components. Moreover, we examine whether they provide personalized recommendations or not, the recommendation type they support, the data factors/features they use, the preferred methodology with which they model the problem and the data representation model they have chosen. By gathering this information we give an overview of the techniques and the features used and define new trends in this domain. The main factor is time that refines the final recommendation revealing relations among entities, which can increase accuracy of the proposals.

I. INTRODUCTION

Nowadays, smart devices are becoming gradually more and more popular as they offer capabilities to users to declare their interest. Thus, many Location-Based Social Networks (LBSN) adopt Recommender Systems (RS) to facilitate the users in decision making. In typical Online Social Networks (OSN), a user is a member of a huge network where s/he can exploit many capabilities. S/he can share information, s/he can make friends or even s/he can accept friend or item recommendations. Thus, users are a vital component of these systems. Many OSNs like Facebook¹, foursquare², Swarm³, Yelp⁴ not only allow users to share their interest (i.e. photos, text, music) but, in addition, they allow users to share their location through their check-ins. Since the mobility factor gave this new perspective of the location, systems have adapted their functionality based on that factor. In particular, they provided POIs (Points of Interest), friend and item recommendations based on proximity of a location. This way, a user benefits from the location dependent recommendations, which aim to facilitate the user when exploring different location during a day.

Despite the fact that the systems assisted the users, they had to face the problems of 'big data' and 'sparsity', which

made the recommendations a difficult task. Many approaches have been introduced in literature to tackle this problem. Moreover, the difficulty is becoming even larger when the recommendations are time-dependent. We argue that periodicity gives a new direction to solve these problems. User's daily movement in different locations has a periodical repetition. For example, a user checks-in near his/her work from Monday till Friday from 8 a.m. till 5 p.m. Similarly, s/he checks-in near his/her house the same days from 6 p.m. till 6 a.m. His/her schedule differs at weekends while resting (i.e. s/he probably may go for a daily trip, or go clubbing etc.) Thus, the time factor is significant and can affect the nature of the recommendations by adding this new direction. People want different recommendations in different time periods. Similar to the previous example, the periodicity may differ not only on a daily basis but also on a weekly or monthly basis or even on yearly basis. For example, a person often goes for vacations in July or August, so s/he should be treated in a proper way based on his/her past history periodicity. Thus, the recommendations should be time-dependent to satisfy his/her needs.

To the best of our knowledge, there are very few papers (9) dealing with time-aware location-based recommendations. In this report, we aim to set a framework for the area of the time-dependent recommendations. In particular, we present an extended comparison among these 9 algorithms, we highlight their respective advantages and drawbacks, and, finally, we give a general overview of the area. The remainder of this paper is organized as follows. In Section II we present a detailed comparison among the algorithms and all their used structural components. Section III describes the new perspectives in this domain and the new potentials. Finally, Section IV concludes the paper.

II. TIME-AWARE CROSS ALGORITHMS COMPARISON

In this section we will present a comparison among algorithms using the notion of time to provide recommendations. There is a distinction among the algorithms that use time in a preprocessing state vs. the algorithms which actually use time to provide time-dependent recommendations. We focus only on the latter category of these algorithms; this is due to the fact that we argue that time is an important factor and affects seriously the final recommendations. Moreover, we are presenting an extended comparison among these algorithms and we are

¹<http://www.facebook.com/about/location>

²<https://foursquare.com>

³<https://www.swarmapp.com>

⁴<http://www.yelp.com>

TABLE I: Cross algorithm comparison

Algorithm	Personalization		Recommendation types					Data factors/ features					Methodology of Problem Modelling							Data representation					
	Generic	Personalized	Friend	Location	Activity	Event (Local)	Event (Non-local)	Time	Activity/tags	User Profile	Trajectories	Locations	Group profiles	Collaborative Filtering	Factorization	Random Walks	Hybrid	Semantic	Probabilistic	Classification	Clustering	Matrix-based models	Graph-based models	Tensor-based models	Hybrid data models
1 LRT [2]	-	✓	-	✓	-	-	-	✓	-	-	-	✓	-	✓	✓	-	-	-	✓	-	-	✓	-	-	-
2 UTE [11]	-	✓	-	✓	-	-	-	✓	-	-	-	✓	-	✓	-	-	-	-	✓	-	-	-	-	✓	-
3 STG [10]	-	✓	-	✓	-	-	-	✓	-	-	-	✓	-	-	✓	-	-	-	✓	-	-	-	-	✓	-
4 MetaFac [6]	-	✓	-	-	✓	-	-	✓	✓	-	-	✓	-	-	✓	-	-	-	✓	-	✓	-	-	✓	-
5 Hoodsquare [12]	-	✓	-	✓	-	-	-	✓	✓	✓	-	✓	-	-	-	✓	✓	-	✓	-	-	-	-	✓	-
6 gSCorr [3]	-	✓	-	✓	-	-	-	✓	-	-	-	✓	-	-	-	-	-	-	✓	-	-	-	-	-	✓
7 Marinho et al. [8]	-	✓	-	✓	-	-	-	✓	-	-	-	✓	-	-	-	-	-	✓	-	-	-	-	-	-	✓
8 FEOR [4]	✓	-	-	-	-	✓	-	✓	✓	-	-	✓	-	-	-	✓	✓	-	✓	-	-	-	-	-	✓
9 PTR [7]	-	✓	-	-	-	✓	-	✓	-	-	-	✓	-	✓	-	✓	-	-	✓	-	✓	-	-	-	✓

emphasizing the emerging trends in this area. In the subsequent subsections we focus in five important characteristics for a recommendation (i.e. Personalization, Recommendation types, Data factors/features, Methodology of the Problem Modeling and Data representation).

A. Personalization

The first examined characteristic is whether the algorithms provide personalized or generic recommendations. This is shown in the third column of Table I. On the one hand, generic recommendations do not count the users’ past history. This kind of recommendations is called generic as it is the same for anyone based on his/her current position. On the other hand, personalized recommendations take into account the users’ history of check-ins and the respective information to provide refined recommendations. For example, if a user’s current location is near the ‘White Tower’ in Thessaloniki, then a generic recommendation for all users in this location could be: *I recommend you to go to the ‘Aristotle Theater’ because it has the highest rating.* Notice that, in this case the highest rating is the factor that defines the nature of the recommendations. Another factor could be the proximity. For the same example, a personalized recommendation could be: *I recommend you to go to the ‘Aristotle Theater’ because your friends had been there in the past.* In this case, there are social ties *and* a check-in history, which affect the final recommendations.

These examples are constrained only from social, proximity and rating factors; however, the examined algorithms add the notion of time as an extra dimension. Thus, an example of a time-aware recommendation could be: *I recommend you to go to ‘Tsimiski str’ because during 4:00-6:00 a.m. five of your friends had been there in the past.* Notice that, in this case the recommendation is a location, which is important for the user’s friends during a specific time period (i.e. 4:00-6:00) proximate to this location.

B. Recommendation types

The second characteristic is the type of recommendations. There are four basic type of recommendations (i.e. Friend, Location, Activity, Event (local and not-local)) as shown in the fourth column of Table I. It is noticeable that the algorithms

support only one kind of recommendation type, which means that they combine only one characteristic with time (i.e. locations with time, activity with time and event with time). Moreover, none of them supports friend recommendations depended on time.

The first type of recommendation is ‘Friend recommendation’. In this case algorithms provide candidate friends to the user. As an example, ‘Nikos Papas’ is the best candidate for our target user because they have four common friends as shown in the second row of the table in the Figure 1. Time-depended friend recommendations could recommend users which have been co-located with our target user several times in the past. In the same example if ‘Nikos Papas’ were co-located with our target user four times in the past, s/he could be the first candidate to be recommended.





Photo	First Name	Last Name	Email address	Explanation		Add as a friend
				Number of common friends	Names of common friends	
	Nikos	Papas	npa@csd.auth.gr	4	Nick John Petro North Maria Downs Paul Manos	<input type="button" value="Add"/>
	Petros	Johns	jhn@csd.auth.gr	3	Petro North Maria Downs Kostas Papas	<input type="button" value="Add"/>
	Maria	Down	md@csd.auth.gr	2	Petro North Kostas Papas	<input type="button" value="Add"/>
	Pavlos	Doe	doe@csd.auth.gr	2	Maria Downs Nick John	<input type="button" value="Add"/>

Fig. 1: Friend recommendation

The second type of recommendation is ‘Location recommendation’. In this case, algorithms provide candidate locations to the user based on his/her current location. As an example, the ‘White Tower’ of Thessaloniki is the best candidate for Mr. Kefalas because it has 14 total check-ins and the higher average rating as shown in the first row of the table in Figure 2. Time-depended location recommendations could recommend locations in which a user has been co-located with other users in the past and are proximate to his/her current position.

The third type of recommendation is ‘Activity recommen-

We recommend the following locations for Mr. Kefalas.

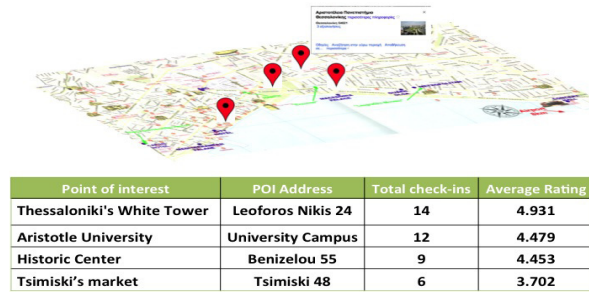


Fig. 2: Location recommendation

We recommend the following events for Mr. Kefalas.

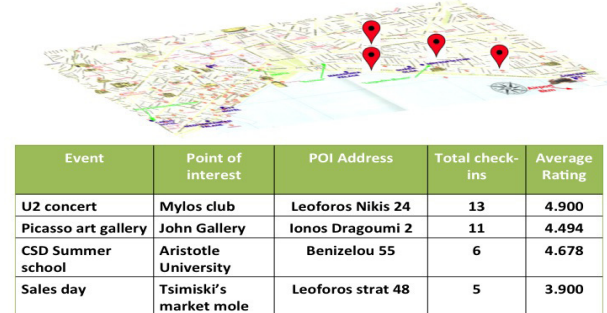


Fig. 4: Event recommendation

dation'. In this case, algorithms provide candidate activities to the user based on his/her activity history. As an example, 'Dinner' at 'Alex Restaurant' is the best activity for Mr. Kefalas because it has 12 total check-ins and higher average rating as shown in the first row of the table in Figure 3. Similar to the previous examples, a time-dependent activity recommendation could be depended on users or his/her friends' activity history at a particular location.

We recommend the following activities for Mr. Kefalas.

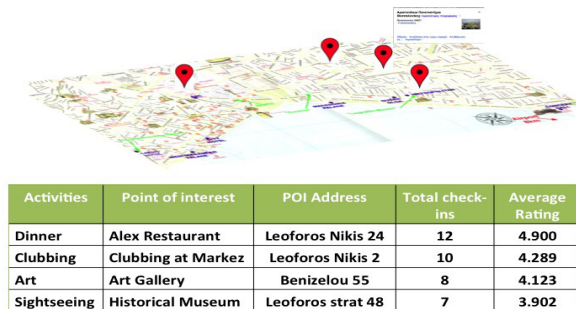


Fig. 3: Activity recommendation

Finally, the fourth type of recommendation is 'Event recommendation'. In this case, we may have a Local or a Non-Local event. The difference between them is the fact that the Local event takes place in a physical place (i.e. in a location), whereas the Non-Local event takes place on the internet (i.e. webinar, online meeting etc.). As an example, 'U2 concert' at 'Mylos club' is a candidate event to be recommended to Mr. Kefalas because it has 13 total check-ins and higher average rating as shown in the first row of the table in Figure 4. A time-dependent event recommendation could be based on the periodical visitation of a user in a location (i.e. if a user has been going for vacation to Naxos island for the last 5 years and this summer a concert is taking place at this location then it is a candidate event recommendation).

Notice that, six to nine algorithms focus on POIs recommendation implying that there is a trend in locations based on mobility. Moreover, none of them makes friend recommendations, which is a drawback since there is a strong correlation among users.

C. Data factors/features

The third characteristic is the data factors/features used during the process of recommendations. There are 6 factors/features (i.e. time, activity/tags, user profiles, trajectories, locations, group profiles) which are commonly used as shown in the fifth column of Table I.

The first feature is 'time'. This feature is related to the time-awareness of the recommendations. All the aforementioned works [2]–[4], [6]–[8], [10]–[12] argue that time is a crucial dimension, which can affect the nature of the recommendations. Periodicity is referred to actions users repeat during a time period [1], [9]. As an example, imagine a user's daily schedule as shown in Figure 5. Red dots represent locations a user checks-in close to his/her house, whereas blue dots represent locations a user checks-in close to his/her work. On the right, there is a clock showing the times of the day that these check-ins were performed. These check-ins are repeated from Monday to Friday on a weekly basis. Thus, this person goes to work at 6 p.m. till 3 a.m. and visits locations proximate to his/her work, while from 3 a.m. till 6 p.m. s/he stays at home or checks-in locations close to his/her house. Notice that, this behavior indicates that there is a pattern of activities/visits during a week. Similarly, there is a pattern during a month or even a year. Therefore, recommendations should be time-dependent.

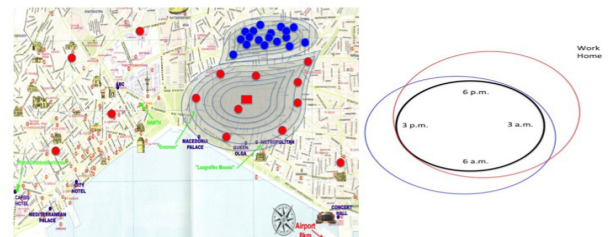


Fig. 5: Time example

The second feature is 'Activity/tags'. This feature is related to activities that a user performs in a location. This way, s/he correlates a location with one or more activities (i.e. jogging, sightseeing, restaurant, bar etc.). Similarly to activities, a user

can tag labels on multimedia, which also relate to a location. For example, Maria uploads a video with the label sightseeing in a location as shown in Figure 6. On the other hand, Anna uploads a text (i.e. ‘what a beautiful view’) in the same location and may correlate this location with a different activity. Notice that, there are many combinations of correlating activities with locations and tags for each user. Thus, each location is handled in a different way among users.

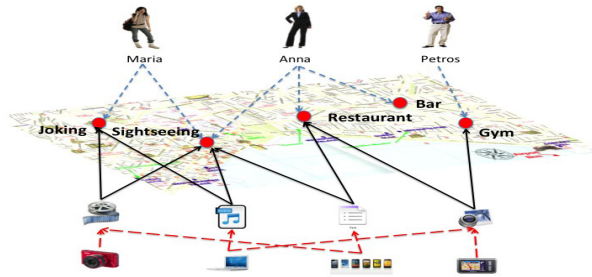


Fig. 6: Activity/tags examples

The third feature is ‘User profile’. This feature is related to the information that a user has in his/her profile on a social network site/app. Most of the times, users provide personal indication (i.e. name, country, town, birthday etc.), which is very useful to find other users similar to them. For example, ‘Nikos Papas’ was born in Thessaloniki in 1980 as shown in Figure 7. By using this information we can find similar users with Nikos close to his/her area (i.e. schoolmates etc.). Information from users’ profiles is used widely and contributes to the accuracy of the recommendations.


	Name: Nikos Papas Country: Greece Town: Thessaloniki Birth day: 1 January 1980 Email : papas@csd.auth.gr Language: Greek, English, German
Friends: Nick John, Petro North Maria Downs, Paul Manos, Maria Pyr.	

Fig. 7: User profile example

The fourth feature is ‘Trajectory routes’. This feature is related to the sequence of check-ins that a user performs. This way, we can define the route from a starting point to an ending point. For instance, a user starting from ‘City Hotel’ and ending at ‘Metropolitan Hotel’, as shown in Figure 8, has the following route: ‘City Hotel’ → ‘White Tower’ → ‘ABC Hotel’ → ‘Macedonia Palace Hotel’ → ‘Queen Olga Hotel’ → ‘Metropolitan Hotel’. In this example, obviously a user is searching for a hotel, thus the system could provide him recommendations with hotels close to his/her position.

The fifth feature is ‘Location’, which is also known as POIs. This feature is related to user’s check-ins. These loca-



Fig. 8: Trajectory route example

tions indicate the places where a user spends time. Thus, the knowledge of a user’s location history gives a direct connection between the physical and the non-physical layer. Then, we can find similar places to the ones the user hangs out or similar activities in other locations with the ones s/he performs in these locations (see Figure 9).

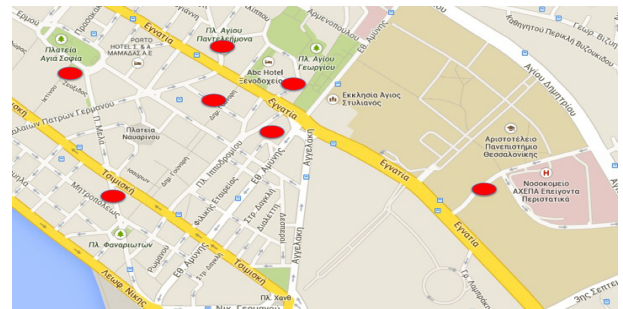


Fig. 9: Location example

Finally, the sixth feature is ‘Group profile’. This feature is related to the groups that a user belongs to as a member. When a user joins a group, this means that s/he has several things in common with the other members of this group. For example, since ‘Yannis Manolopoulos’ is a member of the ‘Delab’ group in a social network, he has many things in common with other members of this group, i.e. same research interests, nearby offices, etc. (see Figure 9). Notice that, this is a direct user-based clustering, which enriches the knowledge we have about him. Thus, there is higher possibility to recommend him someone from the group, who is not friend with him yet.

D. Methodology of Problem Modeling

The fourth characteristic is the methodology used to model the problem. Notice that, there are 8 different kinds of methodologies (i.e. Collaborative Filtering (CF), Factorization, Random Walks, Hybrid, Semantic, Probabilistic, Classification and Clustering) as shown in the sixth column of Table I.

Collaborative Filtering models belong to the memory-based algorithms, which are supposed to be lazy since they do not create a model beforehand for future searches. They create relation matrices in pairs and they match each entity by using clustering or classification techniques with entities similar to the training set. These models are very popular in LBSNs and seem to work fine for modeling time-based problems as they have been used three times [2], [7], [11].








	Group: delab csd auth Country: Greece Town: Thessaloniki	Members:   Pavlos Kefalas Giannis Manolopoulos
	About: Delab conducts research in the areas of databases, data mining and sensor networks in the Department of Informatics at Aristotle University. Our research is focused, but not limited to, on web mining, spatial and temporal databases, music databases, information retrieval, wireless sensor networks (indexing, caching, broadcasting, routing scheduling) and streaming data.	  Apostolos N. Papadopoulos Anastasio Gounaris
Upcoming events: <ul style="list-style-type: none"> • 26th International Conference on Advanced Information Systems Engineering (CAISE), Thessaloniki, Greece, June 2014 • 18th East European Conference on Advances in Databases and Information Systems (ADBIS), Ohrid, FYROM, Sept 2014 	  Kostas Tsihlias Panagiotis Katsaros	

Fig. 10: Group profile examples

Factorization models minimize the size of a huge matrix or a tensor to smaller matrices to reduce the computations cost. By multiplying these smaller matrices we return to the original matrix or tensor. Even though factorization techniques are very popular in LBSNs, they have been used only twice [2], [10] in time-dependent, which indicates that they are not good enough to model time.

Random Walks models propagate a graph in n -steps to rank the possibility of an initial node to jump to another node. Despite the fact that Random walks can model graphs very well, they are not very popular in frameworks dealing with time-dependent recommendations. Notice that only Xiang et al. [10] use a RWR approach to model the problem.

Hybrid models gradually gain importance because they are flexible and allow the researchers to combine all the methodologies presented in this subsection. Notice that they have been used three times [4], [7], [12], which means that they allow researchers to model time data very well.

Semantic models use ontologies to model the problem. In particular, the information underlying in each nested layer inherited to the layers below. This information enriches our knowledge for the entities and reduces the computation cost. There are only two research approaches in literature [4], [12], which provide time-aware recommendations.

Probabilistic models simulate a class of entities assigning an associated probability to each one individually. They can be divided in three types. First, it is a Probabilistic distribution assigning a probability to each subset of possible outcomes of a data set. Second, it includes Machine Learning methods divided in Neural Networks and Support Vector Machines (SVMs). Finally, the third model is Probabilistic graphical models, which consists of Markov models, Hidden Markov models, Bayesian models and Stochastic grammars. The fact that they have been used five times shows that they are the most popular method to model time-dependent problems [2], [3], [8], [10], [11].

Classification models identify the category in which a new observation belongs to. They are quite popular modeling time problems because they use a training set with which categories have already known; this is why they have been used twice [6], [12]. Thus, it is much easier to have a prior knowledge of the exact number of the categories exist.

Clustering models are in a sense similar to classifications, however, with the difference that there is no prior knowledge for the number of categories. For this reason, they have been used only two times [6], [7] in modeling time problems.

Notice that, most researchers use more than one methodology to model the problem in question as shown in the sixth column of Table I. In particular, five of the frameworks [2], [4], [6], [7], [12] use three different methodologies, whereas two of them use two different methodologies [10], [11] and one methodology [3], [8] respectively to model the problem.

E. Data representation

Finally, the last examined characteristic is the data representation. There are four models for representing the data as shown in the seventh column of Table I (i.e. Matrix-based, Graph-based, Tensor-based and Hybrid data models).

Matrix-based (MB) models are very popular in recommender systems, known as Matrix Factorization techniques. Even though they are so popular, it seems that they are not good enough in handling time depended recommendations; this is the reason why it has been used only once as shown in the seventh column of Table I. In basic form they compose the latent factor model in pairs of each entity (user, location, activity, rating, session etc.). Each of these matrices holds a relation between two entities. This way, data can be modeled and reveal patterns. Additionally, these matrices can be combined to solve the sparsity problem.

Graph-based (GB) models, similarly to Matrix based ones, hold information at each level of a k -partite graph. In particular, in unipartite graphs exists information indicating the relation among same entities (i.e. user-user, location-location etc.); also, in bipartite graphs there is information indicating the relation between two different entities (i.e. user-location, user-activity etc.). Moreover, in tripartite graphs there is information indicating the relation among three different entities (i.e. user-location-activity, user-activity-session etc.). In the same direction there are k -partite graphs carrying information for each of the k entities. Even though these models are very popular in recommender systems, to the best of our knowledge there is no research work in literature using GB models.

Tensor-based (TB) models, and specifically tensor factorization models, are based on a multidimensional matrix analysis. In particular, a tensor holds information of three or more entities (i.e. user-location-activity etc.). The simplest form of a tensor is a cube where each dimension holds information about a specific entity. Moreover, this cube can provide three different inner products focusing on one dimension each time. In our example, the three inner products focus on user recommendations, location recommendations and activity recommendations. These models are quite popular and they have been used three times as shown in the seventh column of Table I.

Hybrid data (HD) are the most popular models handling time-dependent recommendations as shown in the seventh column of Table I. Each of the aforementioned models presented in this section has many weaknesses. Hybrid models seem as the best choice to overcome these problems as they combine many different models addressing all the facets of the problem.

III. FUTURE WORK

Extending our previous work [5], where we have presented the perspectives of time, privacy and explainability, in this paper we focus only on the first aspect. In the present section, we will introduce some new directions in this area. In particular, time-depended recommendations should not only focus on similarities among the entities but they should take into consideration relations underling in a network. For example, co-locations (i.e. a set of users hanging out in a location or a proximate locations during a time session), co-activities (i.e. a set of users performing activities in a location or a proximate location during a time session), co-events (i.e. a set of users taking part in events in a location or a proximate location during a time session) may contain extra information about the importance of the items/locations/friends/activities.

Another new perspective could be a constrainer which refines the final recommendations based on time and hinders some of them from being proposed. As an example, imagine a user who lives in a region with clubs and all his/her check-ins are proximate to such places. As a consequence, the system would recommend POIs or activities to proximate clubs. In a case scenario these recommendations could be provided during rush hours i.e. at 2:00 a.m., hours which night clubs are full of people. It is understood that receiving inaccurate recommendations during the night could be very annoying.

Moreover, another new perspective could be a hybrid model combining the trending items/locations/events/words/text etc. based on their top peak hours. As an example, imagine a user who travels and checks-in locations. A time-depended hybrid model could provide recommendations based on events/articles/locations/friends etc. which are very popular at that moment close to him. For example, if the same user is sightseeing around the ‘White Tower’ of Thessaloniki in the morning 28th October (a day of national celebration), then s/he could be proposed to participate to these celebrations. Notice that, in the evening of the same day, s/he will receive different kinds of recommendation after the end of the celebrations. Similarly to the previous cases, the recommendation will differ from hour to hour and from day to day. Thus, the user must be treated in a proper way taking under consideration the notion of time.

Finally, our last proposal focuses on evolution of the Location-based Social Networks (LBSN). In particular, LBSNs distinguish the physical from the non-physical world and their recommendations depend on proximity. A new perspective could be recommendations of regular services or web services either we are referring to physical or in non-physical activities depended on the users’ check-in history. Imagine a user at a specific location who is seeking for a transportation means or that s/he is using his/her smart phone searching for a web service to rent a car and depart for a trip. Then the system should be able to combine the web services with the local transportation media and recommend the optimal solution. These recommendations should take into consideration the period dependencies and the schedule of the course.

IV. CONCLUSION

Nowadays, with the evolution of the LBSNs, users are able to share their interests with their friends simply by

using smart devices. The main parameters characterizing these networks are: mobility, proximity and periodicity. We argue that periodicity is an upcoming trend in recommender systems. In this paper, we present an extended comparison of 9 state-of-the-art recommendation frameworks and all the used structural components. In addition to these, we present new perspectives in this domain with respect to the notion of time.

REFERENCES

- [1] Eunjoon Cho, Seth A. Myers, and Jure 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 (KDD)*, pages 1082–1090, 2011.
- [2] Huiji Gao, Jiliang Tang, Xia Hu, and Huan Liu. Exploring temporal effects for location recommendation on location-based social networks. In *Proceedings of the 7th ACM Conference on Recommender Systems (RecSys)*, pages 93–100, 2013.
- [3] Huiji Gao, Jiliang Tang, and Huan Liu. gscorr: Modeling geo-social correlations for new check-ins on location-based social networks. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM)*, pages 1582–1586, 2012.
- [4] Shen-Shyang Ho, Mike Lieberman, Pu Wang, and Hanan Samet. Mining future spatiotemporal events and their sentiment from online news articles for location-aware recommendation system. In *Proceedings of the 1st ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems (MobiGIS)*, pages 25–32, 2012.
- [5] Pavlos Kefalas, Panagiotis Symeonidis, and Yannis Manolopoulos. New perspectives for recommendations in location-based social networks: Time, privacy and explainability. In *Proceedings of the 5th International Conference on Management of Emergent Digital EcoSystems (MEDES)*, pages 1–8, 2013.
- [6] Yu-Ru Lin, Jimeng Sun, Paul Castro, Ravi Konuru, Hari Sundaram, and Aisling Kelliher. Metafac: Community discovery via relational hypergraph factorization. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 527–536, 2009.
- [7] Eric Hsueh-Chan Lu, Ching-Yu Chen, and Vincent S. Tseng. Personalized trip recommendation with multiple constraints by mining user check-in behaviors. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems (SIGSPATIAL)*, pages 209–218, 2012.
- [8] Leandro Balby Marinho, Iury Nunes, Thomas Sandholm, Caio Nóbrega, Jordão Araújo, and Carlos Eduardo Santos Pires. Improving location recommendations with temporal pattern extraction. In *Proceedings of the 18th Brazilian Symposium on Multimedia and the Web (WebMedia)*, pages 293–296, 2012.
- [9] Daniel Preoțiuc-Pietro and Trevor Cohn. Mining user behaviours: A study of check-in patterns in location based social networks. In *Proceedings of the 5th Annual ACM Web Science Conference (WebSci)*, pages 306–315, 2013.
- [10] Liang Xiang, Quan Yuan, Shiwan Zhao, Li Chen, Xiatian Zhang, Qing Yang, and Jimeng Sun. Temporal recommendation on graphs via long- and short-term preference fusion. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 723–732, 2010.
- [11] Quan Yuan, Gao Cong, Zongyang Ma, Aixin Sun, and Nadia Magnenat Thalmann. Time-aware point-of-interest recommendation. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 363–372, 2013.
- [12] Amy X. Zhang, Anastasios Noulas, Salvatore Scellato, and Cecilia Mascolo. Hoodsquare: Modeling and recommending neighborhoods in location-based social networks. *CoRR*, page 15, 2013.