A Graph-based Taxonomy of Recommendation Algorithms and Systems in LBSNs

Pavlos Kefalas, Panagiotis Symeonidis and Yannis Manolopoulos,

Abstract—Recently location-based social networks (LBSNs) gave the opportunity to users to share geo-tagged information along with photos, videos, and SMSs. Recommender systems can exploit this geographic information to provide much more accurate and reliable recommendations to users. In this paper, we present and compare 16 real life LBSNs, bringing into surface their advantages/disadvantages, their special functionalities, and their impact in the mobile social Web. Moreover, we describe and compare extensively 43 state-of-the-art recommendation algorithms for LBSNs. We categorize these algorithms according to: personalization type, recommendation type, data factors/features, problem modeling methodology and data representation. In addition to the above categorizations which cannot cover all algorithms in an integrated way, we also propose a hybrid k-partite graph taxonomy to categorize them based on the number of the involved k-partite graphs. Finally, we compare the recommendation algorithms with respect to their evaluation methodology (i.e. datasets and metrics) and we highlight new perspectives for future work in LBSNs.

Index Terms—Recommender systems, Location-based recommendations

1 Introduction

Nowadays, social media flood Internet allowing users to communicate and share their interests with others. The location factor gave a new perspective during the procedure of sharing geo-tagged information along with notes, photos, videos, SMSs and so on. This subset of On-line Social Networks (OSNs) is known as Location-based Social Networks (LBSNs). Location merges the physical layer with the digital one giving the opportunity to detect users' preferences from their behaviors and their choices. Recently, the need for more accurate recommendation leads researchers to combine the digital with the physical world using information with location data. Wireless technologies via smartphones and smart devices gave a new perspective arising from users' mobility behavior. Users doing their daily schedule (going to work, to the gym etc.) use smart devices through wireless technologies and share geospatial data relating their location to their interests. This merge brings into surface new dimensions/factors in the problem of recommendation that haven't been searched before.

These factors can be summarized in the following two points:

Location factor. Location contains information about users' ratings for items, venues, places, etc. This information can reveal relationships among location on the one hand and, on the other hand, users, media, items, check-in frequency, events, activities, sessions and, finally, groups.

Time factor. Time is a factor that can help in providing more accurate recommendations. At different times of the day users want different services according to their location and their mood. Therefore, time must be taken thoroughly into consideration. The relationship between time and location gives us hidden ties as well.

Last years researchers have proposed many different kinds of methodologies and algorithms to handle the recommendation problem in LBSNs. Bao et al. [1] have recently published a survey of 22 recommendation algorithms in LBSNs, missing, however, to categorize real-life LBSNs. Moreover, their survey described three different taxonomies (recommendation type, methodology, and data factors) of algorithms in LBSNs, with no clear connection among them. That is, all three taxonomies are parallel to each other and algorithms are mixed up among the taxonomies.

In this paper, we survey 43 state-of-the-art algorithms in LBSNs, plus we survey 16 real life systems in LBSNs presented or published during the past 5 years. In addition, we propose a novel taxonomy of algorithms in LBSNs, which categorizes all algorithms in an integrated way, acting as a hyper-taxonomy of other proposed taxonomies [1], as will be presented in Section 4.

The highlights of this contribution can be summarized as follows:

- Survey and categorization of LBSNs websites,
- Survey and categorization of algorithms in LBSNs,
- Provision of a new taxonomy to categorize the algorithms taking into account the different kinds of information used,
- Provision of an extended 4-D Hybrid Explanation style of recommendations,
- Survey of the evaluation methods used, and
- New perspectives for future work.

- *P. Kefalas, P. Symeonidis and Y. Manolopoulos are with the Department of Informatics, Aristotle University of Thessaloniki, 54124 Greece. E-mail: {kefalasp,symeon,manolopo}@csd.auth.gr*

TABLE 1: Selected location-based Social Networks

		Platform		Persona-lization		System features					Recommendation types					Explanation			Impact
	Systems	Web interface + applet	Web interface	Generic	Personalized	Cross-system connectivity	Wish list	Duplicates correction	Map visualization	Check-ins	Location/Route	Friend	Followee (User or Community)	Activity/Tag	Event	User	Activity	Location	Number of registered Users
1	foursquare[1]	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	-	60 M
2	yelp[2]	✓	-	✓	-	✓	-	-	✓	-	✓	-	✓	✓	✓	-	✓	-	50 M
3	getnowapp[3]	✓	-	✓	-	✓	-	-	✓	-	✓	-	-	✓	✓	-	-	-	3 M
4	everplaces[4]	✓	-	✓	-	✓	-	-	✓	-	✓	-	✓	-	-	-	✓	-	4 K
5	fieldtripper[5]	✓	-	✓	-	✓	-	-	✓	-	✓	-	✓	-	-	-	-	-	NA
6	tagwhat[6]	✓	-	-	✓	✓	-	-	✓	✓	✓	-	✓	-	-	-	-	-	NA
7	zagat[7]	✓	-	✓	✓	-	-	-	✓	-	✓	-	-	-	-	-	✓	-	4 M
8	raved[8]	✓	-	-	✓	✓	✓	-	✓	-	✓	-	✓	✓	-	-	-	-	8 M
9	snoox[9]	✓	-	✓	✓	✓	-	-	✓	✓	✓	-	✓	✓	-	✓	-	-	NA
10	google+[10]	-	✓	-	✓	✓	-	-	✓	✓	✓	-	-	-	-	✓	✓	✓	540 M
11	geosocialrec[11]	-	✓	-	✓	-	-	-	✓	✓	✓	✓	-	✓	-	✓	✓	✓	30 K
12	wisdom[12]	✓	-	✓	-	✓	-	-	✓	✓	✓	✓	-	✓	✓	✓	-	✓	NA
13	Facebook Places[13]	✓	-	✓	-	-	-	✓	✓	✓	✓	✓	✓	-	✓	✓	-	✓	1.15 G
14	Sindbad[14]	✓	-	✓	✓	-	-	-	✓	✓	✓	-	✓	-	-	✓	-	✓	NA
15	flickr[15]	✓	-	✓	✓	✓	-	-	✓	✓	✓	✓	✓	-	-	✓	-	-	87 M
16	Twitter[16]	✓	-	✓	✓	✓	-	-	-	✓	✓	✓	✓	✓	✓	✓	-	-	302 M
	Systems supporting the feature	14	2	12	10	12	2	2	15	10	16	6	11	8	5	9	6	5	2.2 G

The sequel is organized as follows. In Section 2, we present and categorize 16 real-life LBSNs. Section 3 presents the categorization of 43 LBSNs algorithms using several criteria (i.e. personalization type, recommender type, data features, methodology of problem modeling and data representation used). Then, in Section 4 we present a new taxonomy of algorithms based on hybrid k-partite graphs. Section 5 provides a comparison of the evaluation used. Finally, Section 6 illustrates new trends and perspectives for future work.

2 Real Life Location-based Social Networks

In this Section, we present 16 selected real-life LBSNs that provide recommendations. We categorize them in many different ways, i.e. platform that they use, their personalization type, their system features, the recommendation type and provided explanation types as shown in Table 1. Our goal is to discover the strengths and the weaknesses of these systems.

Firstly, we divide these systems based on the fact of having an applet for tablets/mobiles or just having a web interface (see third column of Table 1). As it can be seen, most of these systems support both access via a web browser and via a specific applet interface. It is obvious, that all LBSNs should provide both interfaces to their users because this gives a better functionality and a user-friendly environment.

Moreover, we categorize recommender systems based on the fact that they support personalization or not (see fourth column of Table 1). It can be seen that there is a balance between the numbers of systems that support generic or personalized recommendations. It is notable, that only 6 systems support both. Generic recommendations do not exploit any knowledge about the user. On the other hand, personalized recommendations are based on the users' profile such as log history, friend's suggestions etc. Systems should be able to provide recommendations to new/unregistered users. Thereupon, it is a significant advantage for LBSNs to support both types.

Next, we examine the internal features supported by the recommender systems. As shown in the fifth column of Table 1, there are 5 features (i.e. cross-system connectivity, wish list, duplicates correction, map visualization, and check-ins). In the following, we discuss each feature in detail.

Cross-systems connectivity allows users to connect to other OSNs, by using the credentials of another network. For example, Facebook Connect [17] in cooperation with Netflix allows users to carry their friendship network from Facebook to Netflix. Thus, a user can take movie recommendations by using the movies his/her friends posted in Facebook, which they liked.

The second feature is the 'wish list'. By using the wish

[1] https://www.foursquare.com
[2] http://www.yelp.com
[3] http://www.getnowapp.com
[4] https://www.everplaces.com
[5] http://www.fieldtripper.com
[6] http://www.tagwhat.com
[7] http://www.zagat.com
[8] http://www.raved.com
[9] https://www.snoox.com
[10] http://www.google.com/+/learnmore/local
[11] http://delab.csd.auth.gr/geosocialrec
[12] http://www.wisdom.com
[13] https://www.facebook.com/about/location
[14] http://sindbad.cs.umn.edu
[15] https://www.flickr.com
[16] https://twitter.com
[17] https://blog.facebook.com/blog.php?post=41735647130

list users can keep notes of places they want to visit or events they want to attend. This feature helps the recommendation engine to provide location recommendations by inter-relating common events/places of users.

The third feature is the 'duplicates correction'. In LBSNs, millions of users check-in at different locations, but tag them with the same word (synonimity). For example, one of the terms often used to tag a location is the word 'home'. LBSNs have many locations tagged with this word, which makes it difficult to distinguish. To solve this problem, some LBSNs have adopted the duplicates correction feature. By using this feature, a user can determine if a given tag is correct or not. Otherwise, s/he proposes a correction. Thus, with time, the system is purified from duplicated names.

The fourth feature is the 'map visualization' and is supported by all systems. This feature allows users to visually locate their current location. Thus, it supports users' mobility, which is essential in LBSNs.

Finally, the fifth feature is the 'check-in'. This feature allows users to declare their location, by using also geo-tagged information (i.e. photo, text, video etc.). By using this feature, users are able to check-in at points of interest (POIs). Also, LBSNs keep information about users' preferences, their activities, and the events they attend.

Furthermore, we divide systems based on the type of recommendation that they provide (i.e. location/route, friend, followee, activity/tag, event), as shown in Table 1. In the following, we provide some basic characteristics of each recommendation type:

Friend recommendation. This type of recommendation suggests possible friends to the target user. Friend recommendations appeared in Symmetric Social Networks (SSNs) such as Facebook and Google+. SSNs include reciprocal relationships among users (i.e. via the underlying undirected graph).

Followee recommendation. This type of recommendation suggests to a target user other "users to follow". This is also known as asymmetric recommendation, since it happens with Asymmetric Social Networks (ASNs), such as Twitter. ASNs include non-reciprocal relationships among users (i.e. the underlying graph is directed). Please notice that a user can be recommended to follow either a user or a group of users, as shown in the fourteenth column of Table 1.

Location/Route recommendation. The location factor used in social media has given new opportunities in users' daily behavior during the last years. For example, a user in a foreign country may want to visit the most important POIs, like archeological sites or museums. In such a case, locations are very useful as guidance. This kind of recommendation uses information about current user's position, his/her location history and his/her friends' history. Please notice that a sequence of recommended POIs can be considered as a special type of recommendation, known as route recommendation.

Activity/Tag recommendation. Activities reflect users' daily life choices. Jogging, clubbing and reading may be some activities suitable for a specific place. A user usually asks the recommender system for an activity recommendation near-by his place of presence. Then, the recommender system suggests to him an activity to perform. For instance, it may recommend him to go for dinner at Alex restaurant, because it is lunch time, it is very close and it has many users' check-ins. Please notice that many times a user (after performing an activity) wants to put a tag about his experience, so that he can retrieve it later more easily. The choice of a suitable tag for a specific activity in a location is not always trivial. Thus, many systems provide users with tag recommendations, helping them to choose a suitable tag for their activity.

Event recommendation. Event recommendation suggests a user what s/he may attend either psychically or on the web. Examples of this type of recommendation could be a concert taking place in 'Mythos club', an internet lecture (i.e. webinar) and/or an on-line auction.

The seventh column of Table 1 introduces the explanation styles of recommendation (i.e., 'User', 'Activity' and 'Location'). Explanations are the heart of each LBSN because they reflect in a transparent way the logic behind a recommendation. Recommendations should be justified to users, so that they can understand the reason and trust the recommendation. Most of LBSNs support user and activity explanation to justify their recommendations. However, distance proximity is the most important factor and should be also considered in explanations.

The last column of Table 1 presents the impact of each LBSN. It can be seen that there are over 2.2 billion registered users. That is, LBSNs are almost everywhere and attract the interest of more than 1/3 of the world population. Additionally, the last row of Table 1 points out the features which have more attractive power towards people. In particular, features which are very popular are adopted by multiple systems in contrast to features which are unpopular and have not been adopted from many systems.

Finally, similar systems to the LBSNs presented in Table 1 are the *Cyclopath*[18] and the *bikely*[19] where users can upload their bicycle routes into the database. In these systems, bikers can find interesting bike routes from the existing ones to follow. However, both systems could be extended to provide route recommendations to their users, based on their similarity with other users.

3 Categorizing Recommendation Algorithms in LBSNs

Here we present the state-of-the-art recommendation algorithms in LBSNs. As mentioned, we categorize these algorithms according to several criteria such as:

[18]http://cyclopath.org
[19]http://www.bikely.com

personalization type, recommendation type, data factors/features, methodology and models and data representation (see Table 2). According to these aspects, we introduce five different taxonomies to categorize recommendation algorithms in LBSNs, as shown in Figure 1. Our goal is to detect the strengths and weaknesses of these algorithms, and at the same time to identify the domain tendency. Next we will analyze the categorization of Table 2.

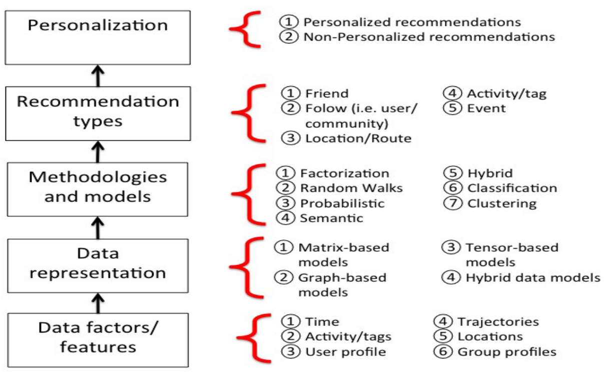


Fig. 1: Overview of algorithms taxonomies.

3.1 Data factors/features

In this Section, we discuss in detail each of the 6 main data features/factors (i.e. time, activity/tags, user profiles, trajectories, locations, group profiles), which are used by algorithms to provide recommendations, as shown in the third column of Table 2 .

Time. Time-aware recommendations require the existence of the time dimension. For example, an algorithm can exploit information about the time that a user sends a post in a LBSN. That is, besides the geographic latitude and longitude of a user's check-in, an algorithm can also exploit the time that the user check-ins a place to leverage the quality of recommendations.

Only few of the algorithms presented in Table 2 try to take advantage of the time dimension. We distinguish between frameworks that exploit time in a preprocessing state vs. frameworks that exploit time to make time-aware recommendations. Frameworks exploiting time at prepossessing state can be found in [2], [6], [21], [25], [36], [37], [49], [56], [57]. They use time to extract stay regions and additional information about users. On the other hand, frameworks that exploit time to make time dependent recommendations use time as a reference point [10]–[12], [14], [27]–[30], [48], [54], [55].

Activity/tags. This feature takes advantage of the different activities that users usually perform and the posted tags. This way, recommendations based on this feature are close to users' behavior and interests. For instance, imagine people who use smart devices and post geo-social information (i.e. photos, activities etc.). These posts include tags, which are words or phrases added in posts, describing them so that it is easier to found them in the future. Also, tags are used to categorize these posts in clusters with other entities with which they have something in common. It can be seen that 14 frameworks take advantage of these features to provide more accurate recommendations [7], [10], [13], [14], [16], [17], [19], [24], [26], [28], [39], [41], [52], [53], [55].

User profiles. Users post information about them (i.e. personal details, likes/dislikes, check-ins, ratings, social relations with other users etc.) in their profile. This kind of information can be exploited to cluster them in groups and reveal relations with other members of the same cluster. Please notice that most of algorithms, which take advantage of the information from user profiles, use the Collaborative Filtering technique.

Trajectories. A sequence of POIs in users' daily schedule composes the trajectory information. This information tracks the user's routes collected from GPS (Global Positioning System) devices. This information contains data about stay point's duration, location row dependence in users' schedule, different paths followed, velocity and acceleration. Algorithms such as [6], [9], [25], [31], [42], [52], [56]–[58] exploit users' routes to provide new location recommendations.

Locations. This feature indicates the places that a user performs his/her activities. Usually users expose their current location to receive other recommendations nearby for performing an activity or two inform their friends about their position. It is noticeable, that most of the frameworks explore location records, as shown in Table 2.

Group profiles. Groups are composed by members with similar features/interests. Users choose to be members of a group based on their common features/interests. As an example, imagine a user in an OSN such as Facebook, who joins a group under the name 'AUTH DELAB team'. By deciding to join this group, s/he indicates of having similar interests with the members of this group. In this case, additional information is provided by the user. Notice that only 5 frameworks ([2], [9], [25], [28], [45]) take advantage of it, as shown in Table 2.

3.2 Data representation

In this section we present the data representation used in frameworks (see fourth column of Table 2). You can observe that there are 4 main types of data representation in use: (i) Matrix-based, (ii) Graph-based, (iii) Tensor-based, and (iv) Hybrid.

3.2.1 Matrix Representation

Several approaches use the structure of matrix as a choice of data representation in the field of recommendation in LBSNs. For example, factorization on matrix structures has been applied by Zheng et al. [56]. After appropriate preparing of the data set, they produced three two dimensional matrices (i.e., activity-activity, location-activity and location-feature matrices).

TABLE 2: Cross algorithm comparison

		Data factors/ features						Data representation				Methodologies and models							Recommendation types					Persona-lization	
	Algorithm	Time	Activity/Tags	User Profile	Trajectories	Locations	Group profiles	Matrix-based	Graph-based models	Tensor-based	Hybrid data models	Factorization	Random Walks	Probabilistic	Semantic	Hybrid	Classification	Clustering	Friend	Followee (User or Community)	Location/Route	Activity/Tags	Event	Non-Personalized algorithms	Personalized algorithms
1	CLAF [56]	-	-	-	✓	✓	-	✓	-	-	-	✓	-	-	-	-	-	-	-	-	✓	✓	✓	✓	-
2	IFC [42]	-	-	-	✓	✓	-	✓	-	-	-	✓	-	-	-	-	-	-	-	-	✓	✓	-	✓	-
3	RMF [2]	-	-	✓	-	✓	✓	✓	-	-	-	✓	-	-	-	-	-	-	-	-	✓	-	-	-	✓
4	diffeRS [8]	-	-	✓	-	✓	-	✓	-	-	-	✓	-	-	-	-	✓	-	-	-	✓	-	-	-	✓
5	LFBCA [47]	-	-	✓	-	✓	-	-	-	-	✓	-	✓	✓	-	-	-	-	-	-	✓	-	-	-	✓
6	LRT [11]	✓	-	-	-	✓	-	✓	-	-	-	✓	-	✓	-	-	-	-	-	-	✓	-	-	-	✓
7	GeoSocialDB [7]	-	✓	✓	-	✓	-	-	-	-	✓	-	-	-	-	✓	-	-	-	-	✓	-	-	-	✓
8	wrwr [32]	-	-	-	-	✓	-	-	-	-	✓	-	✓	-	-	-	-	-	✓	-	✓	-	✓	-	✓
9	CPCT [39]	-	✓	-	-	✓	-	-	-	-	✓	-	-	✓	-	✓	-	-	-	-	✓	-	-	-	✓
10	UPOI-Mine [53]	-	✓	✓	-	✓	-	-	-	-	✓	-	-	✓	✓	-	-	-	-	-	✓	-	-	-	✓
11	LARS [26]	-	✓	✓	-	✓	-	-	-	-	✓	-	-	-	-	-	✓	-	-	-	✓	-	-	-	✓
12	MSSP [24]	-	✓	✓	-	✓	-	-	-	-	✓	-	-	-	-	-	✓	-	-	-	✓	-	-	-	✓
13	TraMSNET [9]	-	-	✓	✓	✓	✓	-	-	-	✓	✓	-	-	-	-	-	-	✓	-	✓	-	-	-	✓
14	ST-Unified [6]	-	-	-	✓	✓	-	-	✓	-	-	-	✓	✓	-	✓	-	-	-	-	✓	-	-	✓	-
15	CADC [25]	-	-	-	✓	✓	✓	-	✓	-	-	-	-	-	-	-	-	✓	-	-	✓	✓	-	-	✓
16	FriendSensing [36], [37]	-	-	✓	-	✓	-	-	✓	-	-	-	✓	-	-	-	-	-	✓	-	-	-	-	-	✓
17	GTS-FR [52]	-	✓	-	✓	✓	-	-	✓	-	-	-	✓	-	-	-	✓	-	✓	✓	-	-	-	-	✓
18	LBSNRank [20]	-	-	-	-	✓	-	-	✓	-	-	-	✓	-	-	-	-	-	-	-	✓	-	-	-	✓
19	SCLN [4]	-	-	-	-	✓	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	✓
20	ITR [45]	-	-	-	-	✓	✓	-	-	✓	-	✓	-	-	-	-	-	-	✓	-	✓	✓	-	-	✓
21	PCLAF [58]	-	-	-	✓	-	-	-	-	✓	-	✓	-	-	-	-	-	-	-	-	✓	✓	-	✓	-
22	RPCLAF [57]	-	-	-	✓	-	-	-	-	✓	-	✓	-	-	-	-	-	-	-	-	✓	✓	-	-	✓
23	Sindbad [41]	-	✓	✓	-	✓	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	✓
24	HRW [19]	-	✓	-	-	-	-	-	-	-	✓	-	✓	✓	-	-	-	-	-	-	-	✓	-	-	✓
25	UTP [54]	✓	-	-	-	✓	-	-	-	✓	-	-	-	✓	-	-	-	-	-	-	✓	-	-	-	✓
26	STG [48]	✓	-	-	-	✓	-	-	-	✓	-	-	✓	✓	-	-	-	-	-	-	✓	-	-	-	✓
27	MetaFac [28]	✓	✓	-	-	-	✓	-	-	✓	-	✓	-	-	-	-	✓	✓	-	-	-	✓	-	-	✓
28	Hoodsquare [55]	✓	✓	✓	-	✓	-	-	-	-	✓	-	-	-	✓	✓	✓	-	-	-	✓	-	-	-	✓
29	gSCorr [12]	✓	-	-	-	✓	-	-	-	-	✓	-	-	✓	-	-	-	-	-	-	✓	-	-	-	✓
30	Marinho et al. [30]	✓	-	-	-	✓	-	-	-	-	✓	-	-	✓	-	-	-	-	-	-	✓	-	-	-	✓
31	ST [16]	-	✓	✓	-	✓	-	-	-	✓	-	✓	-	-	✓	-	-	-	-	-	✓	-	-	-	✓
32	HMMs [31]	-	-	-	✓	-	-	-	-	✓	-	-	-	-	-	✓	-	✓	-	-	✓	-	-	-	✓
33	Eventer [21]	-	-	✓	-	-	-	-	-	-	✓	-	-	-	-	✓	-	-	-	-	-	-	✓	-	✓
34	SPG [43]	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	✓	✓	-	✓	-	-	-	-	-	✓
35	USG [50]	-	-	-	-	-	-	-	-	-	✓	-	-	✓	-	✓	-	-	-	-	✓	-	-	-	✓
36	SNAIR [44]	-	-	✓	-	✓	-	-	-	-	✓	-	-	-	-	✓	-	-	✓	-	✓	✓	-	-	✓
37	FEOR [14]	✓	✓	-	-	✓	-	-	-	-	✓	-	-	-	✓	✓	✓	-	-	-	-	-	✓	✓	-
38	PTR [29]	✓	-	-	-	✓	-	-	-	-	✓	-	-	-	-	✓	-	✓	-	-	-	-	✓	-	✓
39	TCL-K [49]	✓	-	-	-	✓	-	-	-	-	✓	✓	-	-	-	✓	-	-	-	-	✓	-	-	-	✓
40	Twittomender [13]	-	✓	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	✓	✓	-	-	-	-	✓
41	CEPR [27]	✓	-	-	-	✓	-	-	-	-	✓	-	✓	✓	-	✓	✓	✓	-	-	✓	-	-	-	✓
42	ST [17]	-	✓	-	-	✓	-	-	✓	-	-	-	-	✓	-	-	-	-	-	-	✓	-	-	-	✓
43	CAPRF [10]	✓	✓	✓	-	✓	-	✓	-	-	-	✓	-	✓	-	✓	✓	-	-	-	✓	-	-	-	✓

This way they managed to fill the missing values in location-activity matrix from auxiliary matrices. Their goal was to take advantage of information existing in these auxiliary matrices. In the same direction, Sattari et al. handled the same problem but using also an SVD approach [42]. More specifically, they merged three matrices X (Location-Activity), Y (Location-Feature) and Z (Activity-Activity) in one mixed matrix T, as shown by Equation 1. Then, they used SVD to reduce dimensions and reveal the latent semantic associations of the data.

$$T_{(l+a)\times(f+a)} = \begin{bmatrix} Y_{l\times f} & X_{l\times a} \\ 0_{a\times f} & Z_{a\times a} \end{bmatrix} \qquad (1)$$

3.2.2 *Graph-based*

Graphs model information in k-partite networks (i.e. user-user unipartite graph, location-user bipartite graph, location-user-activity tripartite graph and location-user-activity-event quadripartite graph) and correlate similarities among them.

Several approaches were proposed using a k-partite network. For example, Leung et al. [25] proposed a framework producing a tripartite graph based on users, locations and activities. Then, by using the Community-based Agglomerative-Divisive Clustering (CADC) algorithm they clustered users in similar groups of activities and locations using also their flow trajectories information.

Moreover, Jin et al. examined POIs separately to make recommendations [20]. At the beginning, they determine the location set S_p to be preprocessed. Then, they rank by score each location $p \in S_p$ and, finally, they compute the ranking of each person's closeness for that location.

Brown et al. [4] claimed that different factors on different platforms tend to play a crucial role for finding social communities (i.e. in Gowalla the location is the important factor for discovering communities, but in Twitter users and their posts are the main factor). Using *Louvain* algorithm they exploit social and places properties to discover different communities.

Friend recommendations were made by Quercia et al. using a graph-based model [36], [37]. In particular, they took advantage of technologies supported in smart phones, which remember if a phone has met anohter phone in the past and if theyhad been co-located in the same region during a time session. These records built a weighted social graph, which is traversed to provide personalised friend recommendation. The personalised friend recommendation list results as a combination of probability with link prediction algorithms (*PageRank*, *shortest path*, *KMarkovChain* and *HITS*).

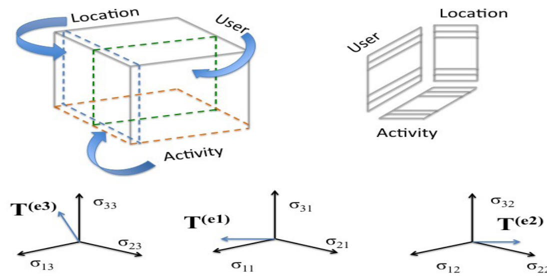
Followee recommendations were made by Hannon et al. [13] with Twittomender. This framework uses a TF-IDF score to assess how frequent a word t_i is of a target user's U_T profile in comparison to other user profiles. This way they give weights which correspond to profile similarity. The higher the profile weight value, the better candidate is for the target user.

3.2.3 Tensor-based Representation

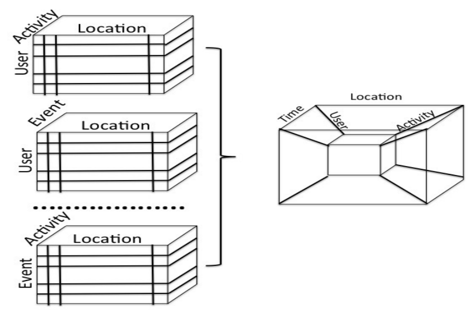
A tensor-based representation is a multi-dimensional array of geometric objects describing relations among scalars, vectors and matrices. In Figures 2(a)-2(b) examples are illustrated of a 3-dimensional tensor (location, user and activity), and a 4-dimensional tensor (location, user, activity and time), respectively. Each dimension holds information about its properties. In Figure 2(a), notice that information proximity in these three layers of the cube provides three different inner products, which focus in a different type of recommendation type each time (i.e. friend, location or activity). In Figure 2(b), notice that the alternating adaptation of this cube is based on time dimension. For example, if time is the factor we focus on, then the cube will slide from inside to outside revealing correlations based on time.

Zheng et al. [58] construct a 3rd order tensor A, which captures the relations among users X, locations Y, activities Z and location features U. They initially decompose tensor A to three low dimensional representations with respect to each tensor entity (i.e. users, locations and activities). Then, they reconstruct the tensor trying to fill all its missing entries. To do so, they exploit additional information from user-user, location-feature, activity-activity, and location-activity matrices. They want to minimize the error between the real and predicted values of the reconstructed tensor.

Additionally to [58] Zheng et al. proposed a Ranking-based Personalized Collaborative Location and Activity Filtering (RPCLAF) algorithm to improve their previous work and personalize their recommendations [56]. Their algorithm models the users' pairwise preferences on activity-locations matrix by ranking loss. Positive and negative values on this matrix are distinguished. Users' pairwise preferences take positive (+1) and negative (-1) values, all other entries are assumed to be unknown and take either question mark (?) or zero (0) value.



(a) Tensor-Based 3D model cube representation



(b) Collaborate location-activity-event-time 4D model

Fig. 2: Tensor-based model 3D and 4D representation

3.2.4 Hybrid

A hybrid data representation uses two or more different data representations (i.e. a matrix and a tensor structure, a graph and a tensor structure etc. A system that recommends events has been proposed by Kayaalp et al., where content-based and collaborative method compose a hybrid model [21], [22]. In particular, the similarity between two different events is the aggregation of two metrics (only two of them are aligned). Content-based similarity metric between two events e_i and e_j and artist a_i and a_j is assumed to be $sim_{m,n} = similarity(a_n, a_m)$ from a set of $\{sim_{m,n}, sim_{n,m|n} = 1, 2, \ldots, s(A_i), m = 1, 2, \ldots, s(A_j)\}$, where largest $sim_{m,n}$ and $sim_{n,m}$ value describes the biggest similarity between two events. The CF technique correlates events based on users' ratings to them, using Equation 2. $U_{i,j}$ represents users, who have rated both events e_i and e_j, u_i is user's rating with values in the range [0-5], whereas μ_u is the mean value of the user's rating vector.

$$r_{ij} = \frac{\sum_{u \in U_{i,j}} (u_i - \mu_u)(u_j - \mu_u)}{\sqrt{\sum_{u \in U_{i,j}} (\vec{u_i} - \mu_u)}^2 \sqrt{\sum_{u \in U_{i,j}} (\vec{u_j} - \mu_u)}^2} \tag{2}$$

A hybrid model was proposed by Ye et al. in [50], where three factors (User (U), Social influence from friends (S) and Geographic influence from POIs (G)) combined linearly in a unified collaborative algorithm to recommend locations. The produced recommendations are based on a power-law probabilistic model, which captures the geographic influence among users and POIs. Specifically, using user-based CF where the user's prediction check-in probability $\widehat{C_{i,j}}$ of user u_i to location l_i is given by Equation 3, where $w_{i,k}$ is the similarity weight between users u_i and u_k. Friend-based CF is given by Equation 4, in this case, where $\widehat{C_{i,j}}$ is the predicted check-in probability of user u_i to location l_i of a set of friends F_i and a directional social weight $Sl_{k,i}$ influence u_k on u_i:

$$\widehat{C_{i,j}} = \frac{\sum_{u_k} w_{i,k} \times C_{k,j}}{\sum_{u_k} w_{i,k}} \tag{3}$$

$$\widehat{C_{i,j}} = \frac{\sum_{u_{k \in F_i}} Sl_{k,i} \times C_{k,j}}{\sum_{u_{k \in F_i}} Sl_{k,i}} \tag{4}$$

Li et al. [49] modeled users rating behavior by exploring the relation or user rating within sets of time periods. In particular, they applied a Stochastic Gradient Decent (SGD) method of user ratings. This way, their model adopts users behavior through users latent factor and based on matrix factorization technics they personalize their recommendations.

Finally, Lian et al. [27] introduced the *Collaborative Exploration and Periodically Return model (CEPR)* which forecasts whether people want to do exploration (i.e., will seek unvisited locations to visit) or not. When people are predicted to do exploration, *CEPR* model combines social knowledge with geographical proximity to recommend unvisited venues. Otherwise, it exploits periodicity and regularity of users' behaviour for figuring out the most possible locations to re-visit.

3.3 Methodologies and models

Here we present the methodologies used to model the problem (see sixth column of Table 2). It is notable that 7 different kinds of methodologies (i.e. Factorization, Random Walks, Hybrid, Semantic, Probabilistic, Classification and Clustering) have been proposed. Next we will discuss each methodology of this list in detail.

Factorization: This model tries to decompose an object (i.e. matrix, tensor etc.) into a product of other objects. By reversing this process, when these objects are multiplied, they produce the original object. When the storage size is minimized, the computation cost is reduced as well. In LBSNs, factorization models are applied in matrices and tensors to reduce their size. It is notable, that factorization is very popular as 12 frameworks use this model [2], [8]–[11], [16], [28], [42], [45], [56]–[58].

A matrix factorization approach has been proposed by Berjani et al. [2], where location recommendation is their main interest. In particular, they recommend to users locations that they have not visited in the past. First, they create a user-location matrix, where rows represent users and columns represent locations. They define a loss objective function, which tries to minimize the squared error between the true and the predicted rating for all pairs of the user-location matrix.

Symeonidis et al. [45] applied *Tensor Factorization* in GeoSocialRec social network, which recommends friends, locations and activities. Recommendation engine constructs a friend similarity matrix by implementing the FriendLink algorithm introduced in [33]. Link weighting uses check-in average geographic distances among users. To acquire these weights, they calculate average distances among all pairs of POIs and users. In particular, initial tensor A is a triplet of (user, location, and activity), which is unfolded in three new matrices. Then, tensor S is constructed by applying SVD in these three matrices, which reduces the dimensions. Also, tensor A is constructed, which is an approximation of tensor A. Finally, recommendations are made based on weights of the reconstructed A.

Random Walks: Random walk is a formalization of a path consisting of a n successive steps. In each step, the algorithm jumps to another state according to some probability distribution (often Random Walk models also jump to previous states). This way, local entities have higher probability to be chosen unlike those at a distance. Thus, location recommendations can be suggested to users based on Random walk methods. For example, a HITS-based Random Walk model was proposed by Ying et al. [52] to match users to each other. They build a *User-Keyword* (UK) directed bipartite graph consisting of users and words connected to them. They also build a *Location-Keyword* (LK) undirected bipartite graph model, consisting of locations and words connected to them. Cao et al. proposed a unified framework that creates a Markov Chain using a two-layered graph (user and location layer) [6]. They define three transition probabilities: (i) $p(U_k|L_i)$ describing the transition probability from a location L_i to a user node U_k, (ii) $p(L_i|U_k)$ describing the transition probability from a user U_k to a location node L_i, and (iii) $p(L_i, L_j|U_k)$ describing the transition probability for user U_k from location L_j to location L_i. Random Walk method is very popular as 11 more frameworks apply this model [6], [15], [18]–[20], [27], [32], [36], [37], [47], [48], [51], [52].

Probabilistic: These models predict the probability of a user's check-in based on past users' check-ins. Most of the papers of this survey implement probabilistic graphical models, which mainly consist of Markov chains and Bayesian networks.

The mathematical structure of Markov chains shows transitions from one state to another, between a finite or countable number of possible states. For example, Quercia at al. [36] proposed FriendSensing, which is a framework allowing mobile users to automatically discover their friends by using social network theories of "geographical proximity" and "link prediction" and by applying an algorithm based on Markov chains.

Moreover, a Bayesian Network (BN) is a probabilistic graphical model that represents a set of random variables and their conditional dependencies via a directed acyclic graph. Park et al. pre-processe geo-social information and train the parameters of a BN [35]. Thus, they obtain a Conditional Probability Table by performing the Expectation-Maximazation model. For every new location/activity request by a user, the highest probability parameter, learned by the BN model, is selected. In general, probabilistic methods have become very popular; for example, see [6], [10]–[12], [17]–[19], [27], [30], [39], [47], [48], [51], [53], [54].

Semantic: Semantic-based models describe context information using ontologies (i.e. OWL Web Ontology Language). These models allow effectual reasoning on data. Also, they allow service collaboration and knowledge sharing. The use of semantics can reduce the storage needed. Additionally, it can minimize the computation cost because information is stored in a nested way, which includes additional information. For example, when a specific user check-ins in a location, then additional information is stored (i.e. country name, state name, city name, street name). The semantic model is used only by 5 frameworks [14]–[16], [53], [55].

Hybrid: Hybrid models combine two or more other methods to overcome the disadvantages of each method which runs alone. Burke et al. [5] proposed many different ways, where individual models can be combined for creating a hybrid model. For example, as shown in Figure 3, Content-based filtering can be combined with CF and/or demographic, utility-based and knowledge-based filtering.

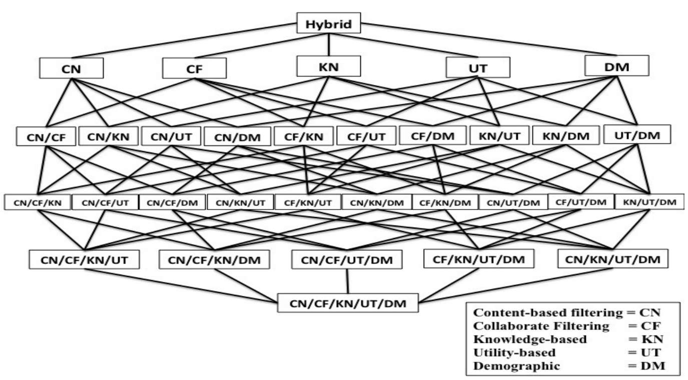


Fig. 3: Combinations in Hybrid models

Scellato et al. use a hybrid modeling technic to tackle the problem of recommendation as a binary classification [43] problem. A graph $G_t = (V_t, E_t)$ consists of many different time snapshots. $V_t = \{U_1, U_2, \ldots, U_{Nt}\}$ are nodes of N_t users and E_t edges connect users, which are in each other's friend list at each time snapshot t. For every snapshot t, they assign a positive label to every couple if they finally become connected with a link at t+1, otherwise a negative one. Classifiers are trained by building training and test sets to recognize positive or negative labeling. Features indicating potential users, which have not yet been connected, are social, place and place-social. Social set is referred to linked pairs that are friends of their friends (and the same time not place-friends) $(S_t \backslash P_t)$. Place set is referred to linked pairs that are place-friends (and the same time not friends of their friends) $(P_t \backslash S_t)$, finally Place-Social is referred to linked pairs among friends of their friends and Place-friends $(S_t \backslash P_t)$.

These approaches combine models from all other models. Hybrid models are widely used because they can overcome drawbacks of particular methods individually. It is encountered that 12 frameworks rely on such models [6], [7], [10], [14], [21], [27], [29], [31], [39], [43], [44], [49], [50], [55].

Classification: These models try to identify the category a new observation belongs to. These models use a training dataset whose categories are already known. In the literature there exist quite a few frameworks applying this method [8], [10], [14], [24], [26]–[28], [43], [52], [55].

Clustering: The difference between classification and clustering is that clustering models try to identify in which category a new observation belongs to, by using no pre-defined or prior knowledge about the number of categories/groups and their characteristics. This model is supported by 5 algorithms [15], [27]–[29], [31].

3.4 Recommendation types

Next, we categorize algorithms based on the provided recommendation type (see sixth column of Table 2). The types of recommendations supported are: (i) friend recommendation, (ii) followee (user or community) recommendations, (iii) location/route recommendation, (iv) activity/tag recommendation, and (v) event recommendation (as introduced in Section 2). Any of these types may be provided to a user either separately or combined. Most of researchers focus on location/route recommendations [2], [4], [6]–[12], [16], [17], [20], [24]–[27], [30]–[32], [39], [41], [42], [44], [45], [47]–[50], [53]–[58]. Less researchers focus on friend recommendation [9], [32], [36], [37], [43]–[45] and followee (i.e. user/community) recommendation [13], [52] and even less on activity recommendation [19], [25], [28], [42], [44], [45], [56]–[58]. Finally, only 5 support event recommendations [14], [21], [22], [29], [32], [56].

Please notice that there are algorithms that provide two or more different types of recommendations. Algorithms that support two different types of recommendations are separated in two groups; the first group supports location and activity recommendations, whereas the second group supports friend and location recommendations. Location and activity recommendations are supported by many frameworks [25], [42], [44], [45], [56]–[58]. Less frameworks support friend and location recommendations [9], [32], [44], [45].

Moreover, there are only 4 frameworks that support three different types of recommendation (among friend, location, activity and event recommendation). Two of these frameworks recommend friends, locations and activities based on the user's request [44], [45]. Zheng et al. also provide location, activity and event recommendations [56]. Finally, Noulas et al. support friend, location and event recommendations [32].

3.5 Personalization

In this Section, we divide recommendation algorithms in LBSNs based on the fact that they support personalization or not. In the last column of Table 2) we distinguish between two cases: (i) Non-Personalised, and (ii) Personalized.

Non-Personalized recommendation is based on popular locations, events and activities. It worths mentioning that no history of the users' movements or additional information is needed for this kind of recommendation. That is, the same recommendation is provided to all users [6], [14], [42], [56], [58], based on the most popular locations.

Personalized recommendations are mainly based on User/Item-based Collaborative Filtering (CF). As shown in Table 2, most algorithms choose CF technique to recommend locations, activities, friends or products [2], [4], [7]–[13], [16], [17], [19]–[21], [24]–[32], [36], [37], [39], [41], [43]–[45], [47]–[50], [52]–[55], [57]. These recommendation algorithms take advantage of information from users' preferences and location history, and suggest personalized recommendations based on the characteristics of each individual user.

4 A NEW TAXONOMY OF ALGORITHMS

4.1 Taxonomy of the algorithms based on hybrid k-partite graph

In this section, we provide a description of the main possible entities of an LBSN, i.e. users, locations, activities and sessions and the connections among them.

In a running example, Figure 4 shows the relations among the aforementioned entities, which are represented in four layers (one layer for each entity). We have divided time of the day in four time sessions. These sessions are associated usually with one or more locations visited by one or more users during a time period. In our running example, there are five users who have visited some places (see Map Locations) and have performed activities such as photos, music text or video containing geo-tagged information.

In the running example of Figure 4, there are 15 graphs of different participating entities (i.e. unipartite, bipartite, tripartite, and quadripartite). On the right side of Figure 4, we can see the 4 generated unipartite graphs (Session, User, Location, Activity). On the left side of Figure 4, we observe 6 bipartite graphs (User-Session, Location-Session, User-Location, Location-Activity, Session-Activity and User-Activity). On the bottom of Figure 4, we show 4 constructed tripartite graphs (Session-User-Location, User-Location-Activity, Session-User-Activity and Session-Location-Activity). Finally, on the top of Figure 4, we present the quadripartite graph (Session-User-Location-Activity). In the following, we describe each one in detail.

Unipartite graphs: On the right side of Figure 4 we have 4 graphs:

1) **Session Graph** is a session-session unipartite graph consisting of relations among different time periods of a day. In particular, this graph represents the time difference among sessions.
2) **User Graph** is a user-user unipartite graph indicating the social relations among users. Each node represents a user connected with another user.
3) **Location Graph** is a location-location unipartite graph presenting relations among locations. Each location is represented as a node and is connected with another location. An edge which connects two nodes indicates a connection between these locations.
4) **Activity Graph** is an activity-activity unipartite graph presenting relations among activities. Each node represents an activity performed by the users in the past. Thus, this graph indicates the geographic distance among geolocated activities.

Bipartite graphs: On the left side of Figure 4, we have 6 graphs:

1) **User-Session Graph** is a bipartite graph that indicates the periods of a day that users perform a social activity (i.e. check-in locations, tag photos etc.).
2) **Location-Session Graph** is a bipartite graph that indicates the relation of locations and time periods.
3) **User-Location Graph** is a bipartite graph presenting locations that users have visited. There are two types of nodes. One type of node represents the user, whereas the second represents the location.
4) **Location-Activity Graph** is a bipartite graph that consists of two types of nodes, i.e. the activity that is performed in a given location.
5) **Session-Activity Graph** is a bipartite graph that presents the activities took place during a period of a day.
6) **User-Activity Graph** is the last bipartite graph presenting activities performed by users.

Tripartite graphs: At the bottom side of Figure 4, we have the following 4 tripartite graphs:

1) **Session-User-Location Graph** is a tripartite graph that presents information about locations that have been visited by users in different time periods of a day.
2) **User-Location-Activity Graph** is a tripartite graph that indicates the activities that have been performed in a specific location by the users.
3) **Session-User-Activity Graph** is a tripartite graph that represents an activity that a user performed during a session.
4) **Session-Location-Activity Graph** is the last tripartite graph which represents an activity that was performed in a location during a session.

Quadripartite graph: Finally, at the top of Figure 4 we have 1 quadripartite graph:

1) **Session-User-Location-Activity Graph:** is a quadripartite graph that incorporates all four dimensions. This way, we have knowledge about user's prefer-

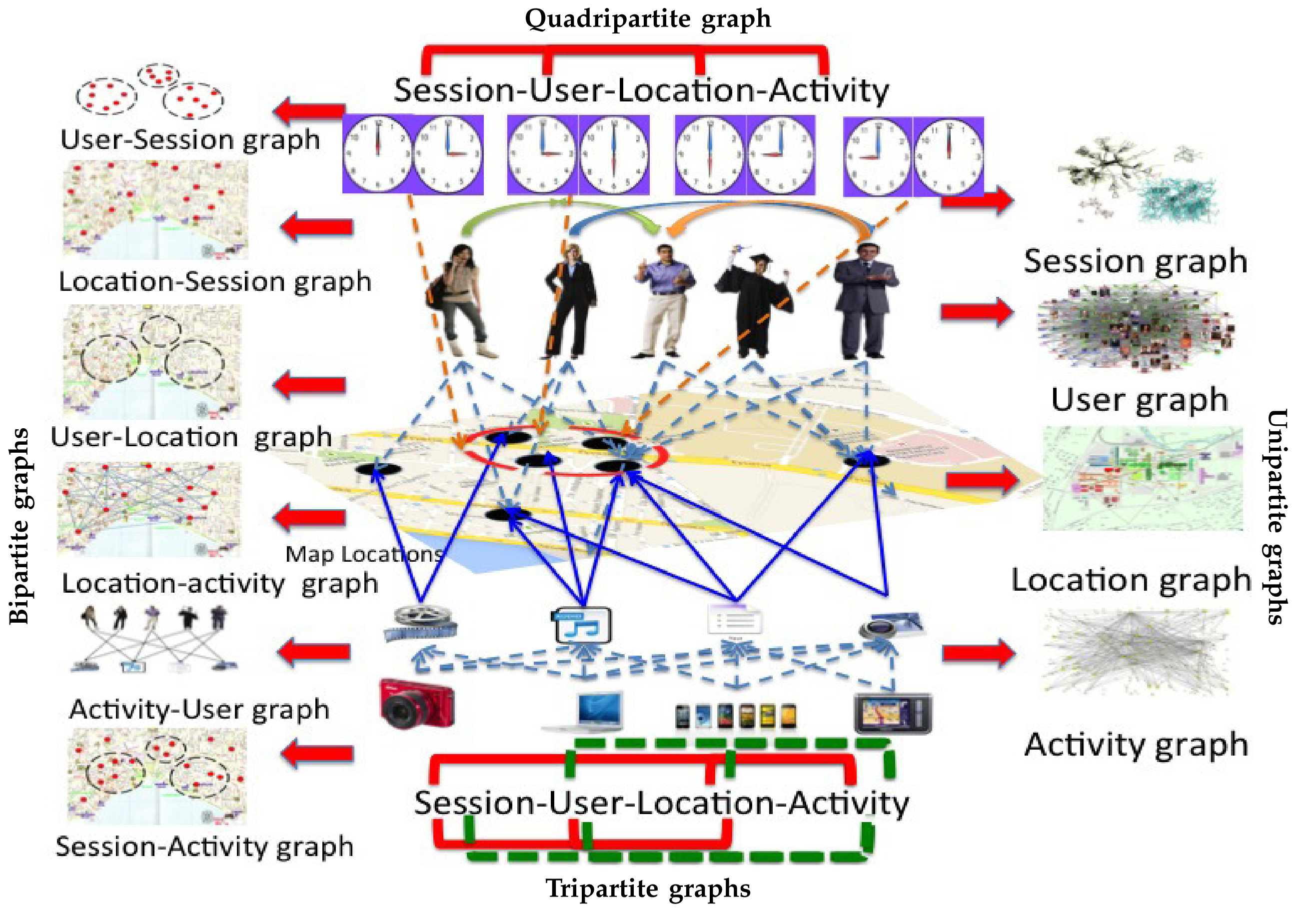


Fig. 4: Shows the relation among the Users, Location, Activities, Groups and the correlation graphs generated

ences for activities at POIs, which are divided in sessions (concerning an entire day). Obviously, this graph is the most enriched one.

Next, we will present a novel taxonomy to categorize the algorithms presented in Table 2 and additionally any new algorithm, based on our hybrid k-partite graph model. It is necessary to emphasise that the hybrid k-partite graph is not a k-partite graph, since there can exist also edges among nodes of the same set (e.g. an edge between a user and another user, i.e. friendship). We denote this special case of a graph henceforth, as hybrid k-partite graph because it is a k-partite graph that consists of k-disjoint sets of nodes (i.e. time, users, locations), incorporating edges among nodes of the same set as well. To the best of our knowledge, the idea of categorising LBSNs algorithms, based on the number of participating sub-networks that they incorporate, has never been used in the past. That is, our new proposed taxonomy categorizes all algorithms in an integrated way, acting as a hyper-taxonomy of other proposed taxonomies [1].

Next, we will present the categorization of algorithms based on our proposed taxonomy, as shown in Table 3. It can be seen that there are 43 algorithms deployed from 2010 until 2015. Apparently there is none implementation of a unipartite graph. 17 of them use information from bipartite graphs, 26 of them take advantage from tripartite or higher order graphs. Please notice that all the above systems are hybrid k-partite graphs because the have at lease one unipartite network (i.e., friendship network) and one bipartite/tripartite (i.e., user, location, and/or session). Please recall that a hybrid k-partite graph consists of k-disjoint sets of nodes (i.e. time, users, locations), incorporating also edges among nodes of the same set. Moreover, in Table 3 under the column 'Method/Factor' we depict the specific main methodology or data factor used.

4.2 Unipartite graphs

Algorithms of this category use information only from unipartite graphs (i.e. either user-user, or location-location, or activity-activity etc.). Please notice that there is no research paper, which belongs only in this category, because a LBSN -by definition- should consists at least of a unipartite (i.e, user-user) and a bipartite network (i.e. user-location).

4.3 Bipartite graphs

The second category contains algorithms that use information from bipartite graphs (i.e. user-location, user-activity, user-event, location-activity, location-event etc.).

TABLE 3: Taxonomy of the algorithms based on hybrid k-partite graph, where MF denotes Matrix Factorization, TD denotes Time-Dependent algorithms, RW denotes Random Walk-based algorithms, TF denotes Tensor Factorization, Hybrid denotes Hybrid k-partite graphs, CF/CB denotes Collaborative and/or Content-based Filtering and O denotes other approaches.

Type	Number	Algorithm	Method/factor	Type	Number	Algorithm	Method/factor
Hybrid Bipartite Graphs	1	CLAF [56]	MF	Hybrid Tripartite or Higher Order Graphs	23	LBSNRank [20]	RW
	2	IFC [42]	MF		24	ST [16]	TD
	3	RMF [2]	MF		25	UTP [54]	TD
	4	ST-Unified [6]	MF		26	STG [48]	TD
	5	diffeRS [8]	MF		27	MetaFac [28]	TD
	6	UPOI-Mine [53]	MF		28	Marinho et al. [30]	TD
	7	wrwr [32]	RW		29	gSCorr [12]	TD
	8	LFBCA [47]	RW		30	SCLN [4]	O
	9	LRT [11]	TD		31	ST [17]	O
	10	CPCT [39]	TD		32	FriendSensing [36]	O
	11	LARS [26]	TD		33	GTS-FR [52]	O
	12	TCL-K [49]	TD		34	Hoodsquare [55]	O
	13	CEPR [27]	TD		35	CADC [25]	O
	14	TraMSNET [9]	O		36	CAPRF [10]	O
	15	GeoSocialDB [7]	O		37	PTR [29]	O
	16	Twittomender [13]	O		38	Eventer [21]	CF/CB
	17	MSSP [24]	O		39	SPG [43]	O
Hybrid Tripartite or Higher Order Graphs	18	ITR [45]	TF		40	USG [50]	CF/CB
	19	PCLAF [58]	TF		41	SNAIR [44]	O
	20	RPCLAF [57]	TF		42	FEOR [14]	O
	21	HMMs [31]	RW		43	Sindbad [41]	O
	22	HRW [19]	RW				

As shown in Table 3, there are 17 algorithms, which will be described briefly in the following.

4.3.1 Matrix Factorization approaches

There are multiple works which use Matrix Factorization models. In this direction, Berjani et al. created a framework named *Regularized Matrix Factorization* (RMF) to predict users' interest for locations that haven't been visited in the past [2]. User-location matrix indicates the relation between users and locations. Specifically, the values of this matrix present the interest of the user for this location. Each location and each user's preference correspond to different vectors. Inner product of a given user and a given location of the corresponding vectors are used to represent the preference of the user for the location.

Similar to Berjani et al., Del Prete et al. propose the *differs* framework to make recommendations based on proximity and similarity among users, by using the user-item matrix, which contains ratings [8]. Proximity is referred on spatial distance among users, whereas similarity is referred to communities with users which have similar behavior. By clustering users into communities, they decompose the entire user-item matrix in a community-preference vector and a smaller rating matrix. The predictions are made based on the top-k community preference of this vector.

Extending these works, Sattari et al. propose the *Improved-Feature-Combination* (IFC) algorithm, where they merge three matrices (i.e. location-activity, location-feature and activity-activity) to construct a mixed matrix T [42]. After applying SVD, they use the Cosine similarity measure to compute a similarity matrix. Finally, to provide recommendations they find the average value of top-m similar nonzero rows and top-n similar nonzero columns.

In the same direction, Zheng et al. present the *Collaborative Location Activity Filtering* (CLAF) algorithm and argue that, using locations from GPS devices and users comments on these locations, may provide new locations and activities recommendations [56]. They focus on the sparsity problem of the location-activity matrix. To solve this problem they use a CF approach under the collective matrix factorization framework. To fill location-activity matrix, they use information from auxiliary sources. The first source is location-activity matrix. To produce this matrix, they follow three steps (i.e. Grid based clustering, stay region extraction, location-based activity statistics). The second source is location-feature matrix. This matrix is produced from location feature extraction and stay regions. Finally, the third source is activity-activity matrix. This matrix mines activity correlation based on information from the World Wide Web.

In contrast to them, Cao et al. find significant semantic locations from trajectory data in a user-location network with the *ST-Unified* algorithm [6]. Their procedure is separated into three steps. At the beginning, they use OPTICS and k-means to cluster the stay points. After, they extract semantic locations by implementing the *Semantics-Enhanced Clustering* algorithm (SEM-CLS), which utilizes these results with semantics such as street addresses names and users visits patterns. Finally, they propose the *Unified Link Analysis Model*, which captures the transition probability on this bipartite network, to rank the results and make location or friend recommendations.

Finally, Ying et al. proposed an algorithm named *Urban POI Mine* (UPOI-Mine) that builds a regression-tree-based predictor to normalize check-in preference with users [53]. This algorithm uses information from: (i) social relations among users, (ii) individual preference, and (iii) POIs popularity.

4.3.2 Random Walk approaches

Wang et al. [47] proposed *Location Friendship Bookmark Coloring Algorithm* (LFBCA), which combines the *Bookmark coloring* algorithm with *Personalized PageRank* algorithm and runs over a bipartite graph network. LFBCA is based on a user-user and a user-location graph. The user-user keeps the friendship network, whereas the user-location connects two users in the graph, if they have visited a common location in the past. LFBCA combines the friendship with the geographical similarity by using a parameter β. Each time, this parameter tunes the importance of each edge depending on the type of the recommendation. At the final step, they compute the final transition probability in the graph and provide a ranked list of location recommendations.

In the same direction, Noulas et al. proposed a variation of the well-known Random Walk model, which is weighted and directed and its name is *Weighted Random Walk with Restart* (WRWR) [32]. In particular, they use a bipartite graph composed by a user-location graph consisting of users ratings for locations. Also, they use a unipartite graph containing the relations among users, derived from the user-user graph. The edges of these graphs have different weights. By giving different weights on each edge, they determine the most important network (i.e. either friends' ties or location ratings preference). After setting the weights in their Random Walk algorithm, they produce personalized friend or location recommendations.

4.3.3 Time-Depended approaches

There are many works which take time under consideration to provide recommendations, as shown in Table 3. Gao et al. [11] proposed the *Location Recommendation with Temporal effects* (LRT) algorithm. They argued that time dimension is crucial for recommendation in LBSNs and introduced a framework to provide time-aware recommendations. In their method, they split user-location matrix into multiple sub-matrices based on the time of a check-in. Then, during the 'temporal factorization' step, each sub-matrix is factorized into two matrices that keep the user and the location information, respectively. In the final step, which is called 'temporal aggregation', they aggregate each low-rank approximation sub-matrix into a new final matrix.

Similarly to Gao et al., Li et al. [49] introduced a model named *Time-aware Comparative Learning of top K ratings (TCL-K)* that builds a behavior learning model of users rating distribution. This model uses information from users rating past history.

Also, Sang et al. proposed a method to solve the activity-plan recommendation problem (i.e. to make suggestions for sequential activities related to user's interests) with the use of their *Context and Personalized POI Category Transition* (CPCT) algorithm [39]. At the beginning, they use a user-activity network to extract sessions. These sessions contain information about users' check-in history. This way, they move beyond the recommendation of the next single activity according to current context. Thus, POIs recommendations are related to user's context and also are related to check-in history personalizing all recommendations.

Moreover, Levandoski et al. [26] support a novel taxonomy on a bipartite graph network over three classes of LBSNs ratings (i.e. spatial ratings for non-spatial items, non-spatial ratings for spatial items, and spatial ratings for spatial items). Thus, they introduce *Location Aware Recommender System* (LARS). Using user partitioning, they exploit users' ratings on locations. In particular, they exploit item location using the technique of travel penalty, which limits the candidate recommendations only to those in close distance.

4.3.4 Other approaches

Kodama et al. [24] proposed the *Multi Level Spatial Skyline with Preferences* (MSSP) algorithm, which incorporates geographical and user's preference information to provide location recommendation in LBSNs. Their framework allows users to keep a profile with personal information, which records their preferences in predefined categories. In this user-preference bipartite network, they apply spatial skyline queries to provide location recommendations

Similarly, Chow et al. [7] proposed the *GeoSocialDB* system to provide 2 different recommendations. A user logs-into the system and updates her profile, the geotagged messages and the ratings in objects. The first type of recommendation concerns 'news recommendations'. In this type of recommendation, information from geotagged messages and the user's profile updates are used to produce a list with news which take place in a distance close to the target user. The second type of recommendation is 'location recommendations'. In this type of recommendations, the system uses user's profile and her rating on objects to provide location recommendationss.

In contrast to the previous works, Gaete-Villegas et al. introduced new similarity measures among users.

In particular, they provided the *TraMSNET* algorithm that takes into account the notion of homophily among users [9] who travel. Finally, to provide followee recommendations, Hannon et al. [13] proposed Twittomender, which uses information taken from users profile such as check-ins behaviour, the users' main characteristics (i.e, gender, education, etc.) and their social network.

4.4 Tripartite or higher order graphs

The third category contains frameworks that use information from tripartite graphs (i.e. session-user-location, user-location-activity etc.). As shown in Table 3 there are 26 frameworks.

4.4.1 Tensor-Based approaches

A tensor-based approach is proposed by Symeonidis et al., named *Incremental-Tensor-Reduction* (ITR), supporting location and activity recommendation [45]. Tensor Factorization is applied on a 3-order tensor (i.e. user-location-activity) to provide location or activity recommendations.

Moreover, Zheng et al. proposed a different tensor-based model, named *Personalized Collaborative Location and Activity Filtering* (PCLAF), to make non-personalized location and activity recommendations [58]. This tensor is composed of auxiliary matrices. These four auxiliary matrices are: user-user, location-feature, activity-activity and user-location matrices. The decomposition of this tensor was presented in detail in Section 3.2.3. Finally, Zheng et al. [57] extended PCLAF and managed to provide personalized recommendations with the *Ranking-based Personalized Collaborative Location and Activity Filtering* (RPCLAF) algorithm.

4.4.2 Random Walk approaches

A hybrid method named *Hidden Markov Model* (HMM) was presented by Mathew et al. [31]. HMM tries to predict users' future mobility. In the beginning, they cluster users' check-ins and a sample of each cluster is used as a training set for their model. Each time a new sequence of check-ins arrives, it is compared with the most similar - to this sequence - cluster. Then, after finding the most similar cluster, HMM recommends to the user the most closer location that belongs to the most similar to the target user cluster.

Moreover, a novel algorithm was presented by Jiang et al. [19]. Specifically, they presented a *Hybrid Random Walk* (HRW) algorithm to integrate information from different domains, so that they can provide more accurate post/activity recommendations. One step further, Jin et al. proposed an algorithm to rank users' check-in histories, which are constantly changing [20]. This algorithm, named *LBSNrank*, takes into account not only check-in records but also their relationship, which can change over time. The LBSNrank algorithm returns the top-k nearby locations and at the same time returns the top-k related people to those locations.

4.4.3 Time-Depended approaches

There are multiple works which incorporate the time dimension into their model. Xiang et al. [48] presented a model, which is built on a *Session-based Temporal Graph* (STG), and exploits user, location and session information by capturing users' short-term and long-term preferences over time. Based on their STG graph, the user-location bipartite graph denotes the long term preferences of a user, whereas the location-session bipartite graph denotes the short term preferences of a user. Xiang et al. proposed also a novel recommendation algorithm named *Injected Preference Fusion* (IPF), which extends the personalized Random Walk algorithm for temporal recommendation. As far as the IPF is concerned, the preferences that are injected into the user node will be propagated to locations visited by the user at all time periods, and then tend to propagate to unknown locations, approximating to user′ s long-term preferences; while preferences injected into the session node will propagate to locations visited by the user at a session, and then tend to propagate to unknown locations, approximating to user ′s short-term preferences.

Similarly, another time-based approach was presented by Yuan et al. [54]. They exploited spatio-temporal characteristics of POIs by using a unified framework consisting of the spatial and temporal dimensions.

In particular, they used linear interpolation to compute the final recommendation score for each location l, by normalizing the two scores that correspond to the temporal and spatial information accordingly. At the end, they used a tuning parameter to compute the final probability that a user u would check-in a location l at a specific time t.

In the same direction, a time-aware and geographical weighting function was proposed by Marinho et al. to improve location recommendations [30]. In the beginning, they cluster users geographic activity based on posted photos to identify Areas of Interest (AOI). Then, they cluster activities over time by using TF-IDF (term frequency, inverse document frequency) temporal weighting to capture the distribution of user activities and predict locations to recommend to the target user.

Moreover, Lin et al. dealt with the discovery of communities over a tripartite (user, location, time) graph to provide time-aware location recommendations [28]. These recommendations take into account users' co-evolving and time-evolving actions, which constantly change. *MetaGraph Factorization* (MetaFac) is the name of their framework, which extracts community structure from various social and geographical interactions.

Finally, a probabilistic approach that exploits also time-evolving data was introduced by Gao et al. [12]. They proposed a geo-social correlation model, named *Geo-Social Correlations* (GSCorr) to solve the 'cold start' problem. Having a user-location-time tripartite graph, they argued that there are four types of geo-social circles corresponding to correlation strength (i.e. $S_{F\bar{D}}$ for Local Friends, $S_{\bar{F}\bar{D}}$ for Local Non-friends, S_{FD} for Distant

Friends, and $S_{\bar{F}D}$ for Distant Non-friends). Additionally, they study users' check-in behavior over distant and close social correlations. To predict the user's behavior for the next check-in, they use three geo-social measures (i.e. Location-Location, User-User and User-Location).

4.4.4 Collaborative and Content-based Filtering

Next, we will discuss a family of approaches which are known as collaborative filtering and/or content-based filtering. Please notice that, because of the ternary relations inherent in LBSNs (e.g., user, location, activity), many recommendation algorithms designed to operate on matrices cannot be applied directly, unless ternary relations are decomposed into three binary relations (i.e., user-location, user-activity, activity-location). This projection has been applied initially in social tagging systems [46], and made easier the application of algorithms in the user-locations or the user-activity matrices.

A well-known approach which can be directly applied on the aforementioned matrices is the User-based CF [3], [38], that forms neighborhoods based on similarities between users. In particular, for a test user, user-based method employs users' similarities to form a neighborhood of his nearest users. Then, user-based CF recommends to the test user, the most frequent items in the formed neighborhood.

Another algorithm proposed by Sarwar et al. [40], denoted as item-based CF, forms item neighborhoods based on similarities between items. In the LBSNs field, a representative work utilizing this type of algorithms for location-based recommendation can be found in [56]. However, we have to underline that the activity-location projection discards the user information and leads to non-personalized location/activity recommendations.

Moreover, *User Social Geographical influence* (USG) is a collaborative filtering recommendation algorithm presented by Ye et al. [50]. This algorithm is based on a naive Bayesian approach and incorporates social and geographic influence with users' preference. They perform user-based CF to derive user's preference and social influence. With this probabilistic approach, they estimate the check-in probability score as a candidate recommendation.

Finally, an event recommender system over a k-partite network was presented by Kayaalp et al., named *Eventer* [21]. This system combines content-based and collaborative methods. Events from different sources are fetched into the system. By mixing content-based with collaborative filtering approach, they provide a recommendation list of top-n events similar to each individual user preferences.

4.4.5 Other approaches

Hu and Ester [16] proposed a spatial model to capture the spatial (textual) perspectives of a post and to be able to predict future user's locations. Their contribution is named *Spatial-Topic* model. This way, given a user and a text document, the goal is to recommend the top-k locations that the user hasn't visited in the past. They argue that the discovery of relevant regions and topics can become more accurate, if they take into account simultaneously both users' interest and their moving behaviour. Moreover, Hu and Ester [17] extended their previous model, introducing the *Social-Topic* model. This new model captures both the social and topic aspects of user check-ins. In particular, it tries to capture the dependency between the latent topic distribution of the posts of each user and the latent POIs distribution of each location, under the assumption that users who share the same topics probably behave similarly.

In contrast to previous work, Gao et al. [10] proposed a unified framework named *Content-Aware POI Recommendation Framework (CAPRF)* that provides POI recommendations. *CAPRF* models three type of content information, which are i) user sentiment indications, ii) user-interest content, and iii) POIs characteristics content, with respect to their relation with the users check-ins.

Leung et al. employ a dynamic clustering algorithm, *Community based Agglomerative Divisive Clustering* (CADC), to distinguish trajectory data into groups with their location recommendation framework [25]. In this case also, we have a tripartite graph network, namely user-location-activity. This grouping is conducted based on similarity among users, similarity among activities and similarity among locations. At first step, they detect stay points by preprocessing GPS history data. To overcome with sparsity problem, they employ tree-based hierarchy graph to model each user's location history. Additional, to update their clusters, when new users' information arrives, they incorporate a community-based agglomerative-divisive clustering algorithm. Next, they refine their recommendations related to a user's activity in a specific location. This refinement identifies three classes of users (i.e. Travelers, Normal and Pattern users). Finally, they rank the scores using LF$\times$IUF (Location Frequency, Inverse User Frequency).

As far as symmetric social networks are concerned, Quercia et al. also use information from a tripartite network to make friend recommendations through their *FriendSensing* algorithm [36], [37]. Friend recommendation is a two-step procedure. In the first step, smart phones keep track of the number of times this smart phone has been co-located with another smart phone. In the second step, they process these records with two strategies, namely shortest path and Markov chain algorithms to rank all possible friends to be recommended to a target user.

In another direction, Ying et al. approximated recommendations with the *Geographic Textual Social Based Followee Recommendation* (GTS-FR) algorithm [52]. They explore the geographic, textual and social properties in their framework to make followee (asymmetric) recommendations. Geographic property represents detail features of each pair of users, textual property and social property, which represent the ratio of follower and following of each pair of users. All these three features are used in SVM classifier to define whether a user is a high ranked candidate to be followed.

Moreover, a novel neighborhood recommendation algorithm was presented by Zhang et al. [55]. *Hoodsquare* is a framework that incorporates this algorithm, which estimates the relation of an area with a user. Firstly, they measure the similarity between a user and a neighborhood, according to textual data. Textual data for neighborhoods are collected by people who had checked-in there in the past. Similarly, they argued that interests and activities, which are performed there, characterize this neighborhood. Finally, they recommend neighborhoods to a user according to his/her similarity to them.

Moreover, Brown et al. focus on exploiting crucial characteristics of communities, which determine the behavior of each user individually [4]. Their algorithm is named *Social Communities in Location based Networks* (SCLN). Their study exploits social and spatial properties of these networks. It concluded, that in each network, there can be different tendencies, which hold communities together.

A link prediction model is presented by Scellato et al. [43], named *SPG*. SPG tries to solve the link prediction problem, which is modeled as a binary classification problem. The main aim is to predict if two users without any common friends, who are visiting the same places, will become in future friends. To accomplish this, they compute two similarity measures : (i) *friends-of-friend*, and (ii) *place-friend*. The first one limits the search on 2-hop node propagation, whereas the second one limits the search into places that a user or his/her friends have visited in the past.

An OSN analyzer for intelligent recommendation system was presented by Shridhar et al. [44]. *Social Network Analyzer for Intelligent Recommendations* (SNAIR) is a system that aggregates users' information from multiple sources and creates an inclusive profile for each user. Then, analytics are performed to 'location' dimension, to 'interaction and interest' dimension and to 'job description' characteristic. Additionally, the system predicts the missing values from this user's profile based on her social relations and interactions. Then, the system can provide to the target user either article, or video or friend recommendations.

Ho et al. [14] proposed *Future Events On-line Recommendation* (FEOR) system, which extracts spatio-temporal information for future events from news articles. In addition, it performs sentimental analysis of each news article to identify the positive or negative perception of this article. Then, they combine all this information to predict and recommend suitable events for a user to attend or avoid. For instance, it could be a prediction of a traffic jam situation, which can be prevented and save a user from unpleasant delays. If the system recognizes it early, by mining web news article, then it can recommend a different route. In general, their mining model consists of two steps. The first step is the 'key words recognition', where toponyms and temporal patterns are identified. The second step is 'matching', where spatio-temporal disambiguation, de-duplication, pairing, and sentiment classification analysis are performed.

A different approach was presented by Sarwat et al. with the *Sindbad* system [41]. The characteristic of this system is that it supports three new services, which are location-aware. These services are location-dependent news feeds, news ranking and news recommendations. Thus, they managed to use information from a tripartite network (i.e, user, location, news) and combine social with spatial relevance into one system.

Moreover, Lu et al. [29] proposed a framework named *Personalized Trip Recommendation* (PTR). PTR provides personalized trip routes considering users' constraints. Their framework consists of the 'Attraction Scoring' module and the 'Parallel Multi Constraint Trip Planning' module. The first one estimates the score for each attraction, considering user and temporal properties. The second one estimates a trip planning based on constraints submitted by the user (i.e. time, budget, etc.) and matches these results with the top-n attractions resulted from the first module.

In summary, it is very hard to compare all mentioned algorithms, since there is no unified framework to test their performance for each objective (i.e. accuracy, efficiency, etc.). To draw some conclusions of our survey there is a trade-off between accuracy and efficiency.

- Algorithms that exploit more data factors (i.e. time, locations, tags, etc.) tend to be more accurate than others which exploit less data factors, since they process more information and enrich their knowledge about users preferences.
- The main drawback of algorithms that exploit more data factors is the low efficiency and the computational cost to process adequately all these data dimensions (i.e. time, location, etc.).

5 Evaluation

In this Section, we present the data sets and metrics used for testing recommendation algorithms in LBSNs. Section 5.1 describes the datasets used in each paper of our survey. Next, in Section 5.2 we present the metrics used for evaluation of the recommendation algorithms in LBSNs.

5.1 Dataset

The datasets used for testing the recommendation algorithms of our survey are shown in Table 4. As shown in the second column of Table 4, there are 43 algorithms under comparison, which were first introduced in Table 2. The most popular data set sources (i.e., Facebook, Twitter, etc.) are presented from the third to the eighth column of Table 4. The ninth column of Table 4 presents

TABLE 4: Datasets

	Algorithms	Facebook	Twitter	Foursquare	Gowalla	Self Collected	Other	Time period	Dataset information
1	CLAF [56]	-	-	-	-	✓	-	4/07-10/09	http://research.microsoft.com/en-us/projects/geolife/
2	IFC [42]	-	-	-	-	✓	-	4/07-10/09	http://research.microsoft.com/en-us/projects/geolife/
3	RMF [2]	-	-	-	✓	-	-	6/00-10/00	Austin in Texas (ATX) dataset: Users: 11896 Spots: 9525 Checkins: 249317, New York City(NYC) dataset: Users: 10132 Spots: 9290 Checkins: 114256
4	diffeRS [8]	-	-	-	-	-	✓	9/97-4/98	MovieLens Light:http://grouplens.org/datasets/movielens/ MIT Reality Mining: http://reality.media.mit.edu/.
5	LFBCA [47]	-	-	-	✓	-	✓	3/08-10/00 2/09-10/00	Gowalla:http://snap.stanford.edu/data/loc-gowalla.html Brightkite: http://snap.stanford.edu/data/loc-brightkite.html
6	LRT [11]	-	-	✓	-	-	-	1/01-3/01	http://www.public.asu.edu/~hgao16/dataset.html
7	GeoSocialDB [7]	-	-	-	-	-	-	-	-
8	wrwr [32]	-	-	✓	✓	-	-	4/00-11/00 8/00-8/00	Foursquare: cities: 11 users: 64608 locations: 61787 check-ins: 304512 AVG check-in/user 4.94 AVG check-in/location 4,78 Gowalla: cities: 11 users: 22967 locations: 98259 check-ins: 342425 AVG check-in/user 25.64 AVG check-in/location 3.18
9	CPCT [39]	-	-	-	-	-	✓	5/01-6/01	Dianping[21] : users: 2342 locations: 32891 check-ins: 152, 154 average records per user 55
10	UPOI-Mine [53]	-	-	-	✓	-	-	-	Gowalla: users: 18159 locations: 1964919 check-ins: 5341191 friend links: 392,246
11	LARS [26]	-	-	✓	-	-	✓	-	Foursquare: users: 1010192 venues: 642990, Synthetic Data: users: 2000 items: 1000 ratings: 500000
12	MSSP [24]	-	-	-	-	-	✓	-	http://www.hitachi.co.jp/Prod/comp/soft1/Entier/
13	TraMSNET [9]	-	-	-	-	✓	-	-	users: 57 locations: 209
14	ST-Unified [6]	-	-	-	-	✓	-	1/07-3/08	cars: 119 location: 352 stay points: 76139
15	CADC [25]	-	-	-	-	✓	-	4/07-10/09	http://research.microsoft.com/en-us/projects/geolife/
16	FriendSensing [36], [37]	-	-	-	-	✓	-	9/04-1/05	http://realitycommons.media.mit.edu/realitymining.html
17	GTS-FR [52]	-	-	-	-	-	✓	12/01-3/02	EveryTrail[22]: users: 35153 trips: 652121 commends: 1325577 links: 3673750
18	LBSNRank [20]	-	-	-	-	-	✓	12/01-2/02	Dianping: users: 204074 locations: 313565 check-ins: 2730072 cities: 347 links: 926720 districts: 1691
19	SCLN [4]	-	✓	-	✓	-	-	8/00-8/00 5/00-11/00	Gowalla: nodes: 65504 edges: 295380 locations: 799116 check-ins: 7388401 Twitter: nodes: 123665 edges: 544215 locations: 1024057 check-ins: 3868845
20	ITR [45]	-	-	-	-	✓	-	4/07-10/09	http://research.microsoft.com/en-us/projects/geolife/ http://delab.csd.auth.gr/geosocial
21	PCLAF [58]	-	-	-	-	✓	-	4/07-10/09	http://research.microsoft.com/en-us/projects/geolife/
22	RPCLAF [57]	-	-	-	-	✓	-	4/07-10/09	http://research.microsoft.com/en-us/projects/geolife/
23	Sindbad [41]	-	-	✓	-	-	✓	-	-
24	HRW [19]	-	-	-	-	-	✓	1/01-1/01	Tencent Weibo[23]: users: 53438 web post: 141879 user label 111
25	UTP [54]	-	-	✓	✓	-	-	8/00-7/01 2/09-10/00	Foursquare: users: 2321 locations: 5596 check-ins: 194108, Gowalla: users: 10162 locations: 24250 check-ins: 456988
26	STG [48]	-	-	-	-	-	✓		CiteULike: http://www.citeulike.org, Delicious[24]: http://www.dai-labor.de
27	MetaFac [28]	-	-	-	-	-	✓	1-27/8/08	http://www.public.asu.edu/~ylin56/kdd09sup.html
28	Hoodsquare [55]	-	✓	-	-	-	-	5/00-11/00	users: 46936 check-ins: 1500986 venues: 74728
29	gSCorr [12]	-	✓	✓	-	-	-	1/01-7/01	http://www.public.asu.edu/~hgao16/Publications.html
30	Marinho et al. [30]	-	-	-	-	-	✓	-	Panoramio[25]: users: 35474 cities: 100 photos: 349166 users/city 351 photos/user 10
31	ST [16]	-	✓	-	-	-	✓	9/00-1/01	http://infolab.tamu.edu/data/ https://www.yelp.com/datasetchallenge/
32	HMMs [31]	-	-	-	-	-	✓	4/07-10/09	http://research.microsoft.com/en-us/projects/geolife/
33	Eventer [21]	✓	-	-	-	-	-	-	-
34	SPG [43]	-	-	-	✓	-	-	5/00-8/00	Gowalla: users: 1251607 locations: 4711703 check-ins: 39932225
35	USG [50]	-	-	✓	-	-	✓	-	Foursquare: users: 153577 locations: 96,229 Whrrl: users: 5892 locations: 53432
36	SNAIR [44]	-	-	-	-	✓	-	-	users: 122 connected profiles: 76232 location: - jobs: -
37	FEOR [14]	-	-	-	-	✓	-	16-29/9/00	news articles 282 news articles 3652 online new sources 21
38	PTR [29]	-	-	-	✓	-	-	1/09-4/01	Gowalla: users: 2192 attractions: 977983 check-ins: 1422762
39	TCL-K [49]	-	-	-	-	-	✓	7/00-10/00	http://www.yelp.com.au/dataset_challenge http://sifaka.cs.uiuc.edu/wang296/Data/index.html
40	Twittomender [13]	-	✓	-	-	-	✓	-	Twitter: users: 20000 tweets: 2000000
41	CEPR [27]	-	-	-	✓	-	✓	3/11-5/13 2/09-10/10	Jiepang[26]: users: 454375 locations: 1000457 check-ins: 36143085, Gowalla: users: 107092 locations: 1280969 check-ins: 6423854
42	ST [17]	-	-	✓	-	-	✓	9/10-1/11	Foursquare: users: 29117 locations: 364259 check-ins: 785249, Yelp: users: 70817 locations: 15585 check-ins: 335022
43	CAPRF [10]	-	-	✓	-	-	-	5/08-9/13	Foursquare: users: 4287 locations: 5878 check-ins: 134556 tips: 19741 comments: 56718

the time period that the datasets have been collected. Finally, the tenth column of Table 4 gives the URL addresses of these data sets, so that someone can easily find and download them.

In the second column of Table 4, please notice that Facebook data set is used for performing experiments in one only algorithm [21]. Moreover, there are 5 data sets crawled from Twitter and used for the experimentation of 5 algorithms. In contrast, Foursquare dataset, which contains richer information, is more popular and is used in 9 cases. Finally, Gowalla dataset [20] has been used 8 times [2], [4], [29], [32], [43], [47], [53], [54]. In any case, we have to mention that most researchers preferred to collect their own dataset [6], [9], [14], [25], [36], [37], [42], [44], [45], [56]–[58]. Moreover, many datasets come from not popular social networking sites [8], [16], [19], [20], [24], [26], [28], [30], [31], [39], [41], [47], [48], [50], [52]. Finally, as shown in the eleventh column of Table 4 thirty researchers have announced the duration that was collected their dataset [2], [4], [6], [8], [11], [12], [14], [16], [19], [20], [25], [28], [29], [31], [32], [36], [37], [39], [41]–[43], [45], [47], [49], [52], [54]–[58]. The older dataset was collected in 2005, whereas the most recent was collected in 2012. As expected, the number of users who have smart phones and check-in locations is increased every year.

Lastly, it is critical to mention, that till now there is no benchmark data set for testing the recommendation algorithms in an integrated way. That is, all papers in our survey used datasets that have different sizes, came from different sources, have attributes and different collecting periods. Under these circumstances, it is very difficult to compare algorithms in a global way and to be able to generalise the results of the performed experiments.

[20]http://www.gowalla.com

[21]http://www.dianping.com
[22]http://www.everytrail.com
[23]http://t.qq.com
[24]http://del.icio.us
[25]http://www.panoramio.com
[26]http://www.jiepang.com

5.2 Metrics

In this Section, we present the metrics used to evaluate each algorithm of our survey. These metrics are shown in the first row of Table 5 and explained in detail in the following.

TABLE 5: Metrics

	Algorithms	NDCG	Precision-Recall	RMSE	MAE	AUC	MAP	False Positive - False Negative	Other
1	CLAF [56]	✓	-	-	-	-	-	-	-
2	IFC [42]	-	-	✓	✓	-	-	-	-
3	RMF [2]	-	✓	✓	-	-	-	-	-
4	diffeRS [8]	-	-	-	✓	-	-	-	-
5	LFBCA [47]	-	✓	-	-	-	-	-	-
6	LRT [11]	-	✓	-	-	-	-	-	-
7	GeoSocialDB [7]	-	-	-	-	-	-	-	-
8	wrwr [32]	-	✓	-	-	-	-	-	-
9	CPCT [39]	-	-	-	✓	-	-	-	-
10	UPOI-Mine [53]	✓	-	-	✓	-	-	-	-
11	LARS [26]	-	-	-	-	-	-	-	-
12	MSSP [24]	-	-	-	-	-	-	-	-
13	TraMSNET [9]	-	-	-	-	-	-	-	-
14	ST-Unified [6]	✓	✓	-	-	-	✓	-	-
15	CADC [25]	✓	-	-	-	-	-	-	-
16	FriendSensing [36], [37]	-	✓	-	-	-		-	-
17	GTS-FR [52]	-	✓	-	-	-	-	-	-
18	LBSNRank [20]	-	-	-	-	-	-	-	✓
19	SCLN [4]	-	-	-	-	-	-	-	-
20	ITR [45]	-	✓	-	-	-	-	-	-
21	PCLAF [58]	✓	-	✓	-	-	-	-	-
22	RPCLAF [57]	-	-	✓	-	✓	-	-	-
23	Sindbad [41]	-	-	-	-	-	-	-	✓
24	HRW [19]	-	-	-	✓	-	-	-	✓
25	UTP [54]	-	✓	-	-	-	-	-	-
26	STG [48]	-	-	-	-	-	-	-	✓
27	MetaFac [28]	✓	✓	-	-	-	-	-	-
28	Hoodsquare [55]	-	-	-	-	-	-	-	✓
29	gSCorr [12]	-	-	-	-	-	-	-	✓
30	Marinho et al. [30]	✓	-	-	-	-	-	-	-
31	ST [16]	-	✓	-	-	-	-	-	-
32	HMMs [31]	-	-	-	-	-	-	-	-
33	Eventer [21]	-	-	-	-	-	-	✓	-
34	SPG [43]	-	✓	-	-	✓	-	-	-
35	USG [50]	-	✓	-	-	-	-	-	-
36	SNAIR [44]	-	-	-	-	-	-	-	✓
37	FEOR [14]	-	-	-	-	-	-	-	-
38	PTR [29]	✓	✓	-	-	-	-	-	-
39	TCL-K [49]	-	✓	-	-	-	✓	-	-
40	Twittomender [13]	-	✓	-	-	-	-	-	-
41	CEPR [27]	-	✓	-	-	-	-	-	-
42	ST [17]	-	✓	-	-	-	-	-	-
43	CAPRF [10]	-	✓	-	-	-	-	-	-

NDCG. The *Normalized Discounted Cumulative Gain* metric takes under consideration the relevant object' s ranking position in the recommendation list. Most relevant objects should be positioned higher in ranked recommendation list.

Precision-Recall. For a test user receiving a list of N recommended locations (top-N list), *Precision* is the ratio of the number of relevant locations in the top-N list to N. Also, *Recall* is the ratio of the number of relevant locations in the top-N list to the total number of relevant locations.

RMSE. The *Root Mean Squared Error* metric measures the square root of the Error between the real and the predict value of the user's rating for a specific location.

MAE. The *Mean Absolute Error* metric measures how close a user's rating prediction for a location is, to the real preference of the user over this location.

AUC. The *Area under Curve* metric measures the probability that a classifier ranks a positive intense higher than a negative, which in both cases are chosen randomly.

MAP. The *Mean Average Precision* metric measures the average precision scores over a set recommendations.

False Positive - False Negative. This metric measures the ratio of the recommended values that have been correctly predicted against the ones that haven't been predicted correctly.

As shown in Table 5, the most popular metric is Precision-Recall (15 times) diagram, followed by $NDCG$ (8 times), $RMSE$ and MAE (5 times) and finally MAP and FP-FN (1 time).

Finally, we have to mention again, as we did in the data sets section, that there is also lack of an integrated experimental protocol, which should be followed by all algorithms. Thus, no fair comparison can be done among algorithms' performances. The creation of a unified experimental protocol to evaluate the compared algorithms in an integrated way (using the same standards) should be considered as future work.

6 NEW TRENDS IN LBSNS

In this section, we focus on new trends in LBSNs. Some new perspectives have already been presented in our previous work [23]. In addition to these, we will present new points of view and new directions. LBSNs in our days focus on mobility and proximity. Most recommender systems take into account these two factors very seriously, but in most cases they omit other important factors. Below, we summarize some new perspectives.

Time-aware recommendations. As mentioned in subsection 3.1, time may reveal patterns in users' behavior. In this direction, as shown in Table 2, several algorithms have taken into account this feature. Since a lot of algorithms represent data using graphs, we propose the creation of a new artificial node, denoted as session node, which is associated either with one or more locations visited by the same user during a time period T_1 (see Figure 5a), or the co-location of more than two users in a place during a time period T_2 (see Figure 5b), or the multiple activities performed by one user during a time period T_3 (see Figure 5c), or finally the same activity performed by more than two users during a time period T_4 (see Figure 5d). Notice that we have introduced 4 kinds of sessions as shown in Figure 5, where cycles with blue color represent the users, red squares represent the locations, whereas green ellipses represent the sessions. The length of a session can last from one hour, to six hours, or even one day. Based on these artificial nodes a new temporal graph can be created.

4-D Explainability: Recommender systems often provide recommendations that come along with explanations giving transparency to the system's functionality. Thus, users can understand in a more comprehensive way the reason of a recommendation. An example of an explanation could be the following statement: "I

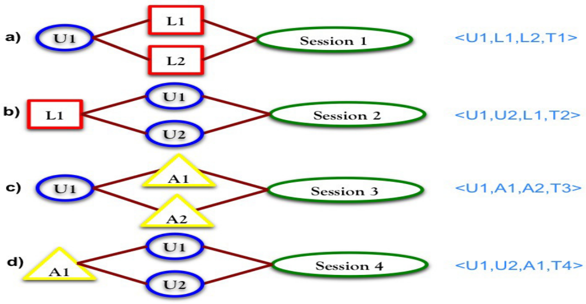


Fig. 5: Time sessions example

recommend you going to Amnesia bar for a drink, because all your friends have rated it with five stars". This explanation uses the friends of the target user to justify the recommendation. There are many different combinations of explanation styles that can be used as shown in Figure 6, initially introduced by Papadimitriou et. al [34] without any time dimension.

However, we claim that time can also play an important role for justifying a recommendation. To support this, one should check that many algorithms exploit the time dimension to provide recommendations, as shown in Table 2. To the best of our knowledge, there is no previous work that mentions the merit of providing time-based justifications. Since time-dependent recommendations seem to be an upcoming trend in recommender systems, they could use the time dimension for providing explanations. Assuming that a user asks for an activity recommendation for Friday night, an example of a time-based explanation is the following statement: "I recommend you going to Olympion bar for a drink, because all your friends go there every Friday night".

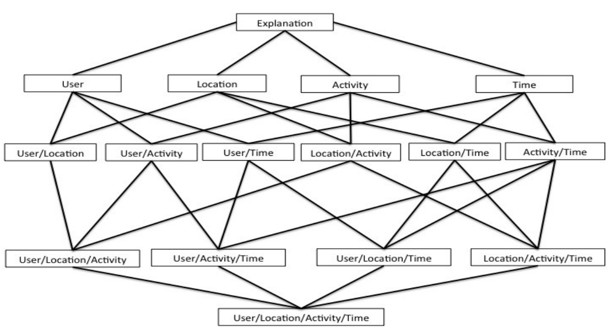


Fig. 6: Combinations in Hybrid explanations styles

As shown in Figure 6, the higher level supports four individual explanations styles (i.e. User explanation, Location explanation, Activity explanation and Time explanation). The second level supports pairs of explanation styles of first level (i.e. User-Location explanations, User-Activity explanation, User-Time explanation, Location-Activity explanation, Location-Time explanation and Activity-Time explanation). The third level supports triplets of explanation styles of the first level (i.e. User-Location-Activity explanation, User-Activity-Time explanation, User-Location-Time explanation and Location-Activity-Time explanation). Finally, the fourth layer consists of all four styles of the first level (i.e. User-Location-Activity-Time). Please notice that time in combination with other factors shown in Figure 6 can provide a hybrid model of justification, which also has no references in literature and may reveal new perspectives.

Exploitation of Geographic Hierarchy: Many researchers are missing to exploit geographic hierarchy information. That is, there is a universal system which defines a geographic hierarchy (i.e. from Continents → Countries → Regions → Cities → Districts → Roads → Addresses). As an example, 'Thessaloniki' is a city, which belongs in the Region of 'Makedonia' and to the Greek country. Consequently, all information existing in upper levels can be inherited to the city of 'Thessaloniki', creating more richer information. Thus, when a user checks-in 'Alexander the Great Avenue', we know that s/he is near the 'seaside' of 'Thessaloniki'. Researchers can benefit by incorporating a geographic ontology hierarchy into their algorithms, since interesting information can be revealed in each level of the geographic hierarchy.

7 Conclusion

Nowadays, LBSNs have flooded Internet. This trend led research to seek for recommendation algorithms in LBSNs, which are able to provide more accurate and justifiable recommendations. Moreover, during the past decade many different websites and many algorithms were introduced to provide suggestions close to users' needs. In this survey, we presented 43 recommendation algorithms in LBSNs and compared 16 real-life LBSNs. We also proposed a new taxonomy of the recommendation algorithms (i.e. hybrid k-partite graph taxonomy). Furthermore, we compared the metrics and the datasets that are used for performing the experimentation in each algorithm. Finally, we proposed new perspectives and directions for future research in the field of LBSNs.

References

[1] Jie Bao, Yu Zheng, David Wilkie, and Mohamed Mokbel. Recommendations in location-based social networks: A survey. *Geoinformatica*, 19(3):525–565, 2015.

[2] B. Berjani and T. Strufe. A recommendation system for spots in location-based online social networks. In *Proceedings 4th Workshop on Social Network Systems (SNS)*, pages 4:1–4:6, 2011.

[3] J. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings 14th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 43–52, 1998.

[4] C. Brown, V. Nicosia, S. Scellato, A. Noulas, and C. Mascolo. Where online friends meet: Social communities in location-based networks. In *Proceedings 6th International Conference on Weblogs & Social Media (ICWSM)*, 2012.

[5] R. Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, 2002.

[6] X. Cao, G. Cong, and C. Jensen. Mining significant semantic locations from gps data. *Proceedings VLDB Endowment*, 3(1-2):1009–1020, 2010.

[7] C.Y. Chow, J. Bao, and M. Mokbel. Towards location-based social networking services. In *Proceedings 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks (LBSN)*, pages 31–38, 2010.

[8] L. Del Prete and L. Capra. differs: A mobile recommender service. In *Proceedings 11th International Conference on Mobile Data Management (MDM)*, pages 21–26, 2010.

[9] J. Gaete-Villegas, M. Cha, D. Lee, and I.Y. Ko. Tramsnet: A mobile social network application for tourism. In *Proceedings ACM Conference on Ubiquitous Computing (UbiComp)*, pages 1004–1011, 2012.

[10] H. Gao, X. Hu J. Tang, and H. Liu. Context-aware point of interest recommendation on location-based social networks. In *Proceedings 29th AAAI Conference on Artificial Intelligence (AAAI)*, 2015.

[11] H. Gao, J. Tang, X. Hu, and H. Liu. Exploring temporal effects for location recommendation on location-based social networks. In *Proceedings 7th ACM Conference on Recommender Systems (RecSys)*, pages 93–100, 2013.

[12] H. Gao, J. Tang, and H. Liu. gSCorr: Modeling geo-social correlations for new check-ins on location-based social networks. In *Proceedings 21st ACM International Conference on Information & Knowledge Management (CIKM)*, pages 1582–1586, 2012.

[13] J. Hannon, M. Bennett, and B. Smyth. Recommending twitter users to follow using content and collaborative filtering approaches. In *Proceedings 4th ACM Conference on Recommender Systems (RecSys)*, pages 199–206, 2010.

[14] S.S. Ho, M. Lieberman, P. Wang, and H. Samet. Mining future spatiotemporal events and their sentiment from online news articles for location-aware recommendation system. In *Proceedings 1st ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems (MobiGIS)*, pages 25–32, 2012.

[15] A. Hotho, R. Jäschke, C. Schmitz, and G. Stumme. Information retrieval in folksonomies: Search and ranking. In *Proceedings 3rd European Conference on the Semantic Web: Research and Applications (ESWC)*, pages 411–426, 2006.

[16] B. Hu and M. Ester. Spatial topic modeling in online social media for location recommendation. In *Proceedings 7th ACM Conference on Recommender Systems (RecSys)*, pages 25–32, 2013.

[17] Bo Hu and M. Ester. Social topic modeling for point-of-interest recommendation in location-based social networks. In *Data Mining (ICDM), 2014 IEEE International Conference on*, pages 845–850, 2014.

[18] M. Jamali and M. Ester. Trustwalker: A random walk model for combining trust-based and item-based recommendation. In *Proceedings 15th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, pages 397–406, 2009.

[19] M. Jiang, P. Cui, F. Wang, Q. Yang, W. Zhu, and S. Yang. Social recommendation across multiple relational domains. In *Proceedings 21st ACM International Conference on Information & Knowledge Management (CIKM)*, pages 1422–1431, 2012.

[20] Z. Jin, D. Shi, Q. Wu, H. Yan, and H. Fan. Lbsnrank: personalized pagerank on location-based social networks. In *Proceedings ACM Conference on Ubiquitous Computing (UbiComp)*, pages 980–987, 2012.

[21] M. Kayaalp, T. Özyer, and S. Özyer. A mash-up application utilizing hybridized filtering techniques for recommending events at a social networking site. *Social Network Analysis & Mining*, 1(3):231–239, 2011.

[22] M. Kayaalp, T. Özyer, and S.T. Özyer. A collaborative and content based event recommendation system integrated with data collection scrapers and services at a social networking site. In *Proceedings International Conference on Advances in Social Network Analysis and Mining (ASONAM)*, pages 113–118, 2009.

[23] P. Kefalas, P. Symeonidis, and Y. Manolopoulos. New perspectives for recommendations in location-based social networks: Time, privacy and explainability. In *Proceedings 5th International Conference on Management of Emergent Digital EcoSystems (MEDES)*, pages 1–8, 2013.

[24] K. Kodama, Y. Iijima, X. Guo, and Y. Ishikawa. Skyline queries based on user locations and preferences for making location-based recommendations. In *Proceedings 1st ACM SIGSPATIAL International Workshop on Location Based Social Networks (LBSN)*, pages 9–16, 2009.

[25] K.W.T. Leung, D.L. Lee, and W.C. Lee. Clr: a collaborative location recommendation framework based on co-clustering. In *Proceedings 34th International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR)*, pages 305–314, 2011.

[26] J. Levandoski, M. Sarwat, A. Eldawy, and M. Mokbel. Lars: A location-aware recommender system. In *Proceedings 28th IEEE International Conference on Data Engineering (ICDE)*, pages 450–461, 2012.

[27] Defu Lian, Xing Xie, Vincent W. Zheng, Nicholas Jing Yuan, Fuzheng Zhang, and Enhong Chen. Cepr: A collaborative exploration and periodically returning model for location prediction. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 6(1):8:1–8:27, 2015.

[28] Y.R. Lin, J. Sun, P. Castro, R. Konuru, H. Sundaram, and A. Kelliher. Metafac: Community discovery via relational hypergraph factorization. In *Proceedings 15th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, pages 527–536, 2009.

[29] E.H.C. Lu, C.Y. Chen, and V. Tseng. Personalized trip recommendation with multiple constraints by mining user check-in behaviors. In *Proceedings 20th International Conference on Advances in Geographic Information Systems (SIGSPATIAL)*, pages 209–218, 2012.

[30] L.B. Marinho, I. Nunes, T. Sandholm, C. Nóbrega, J. Araújo, and C.E.S. Pires. Improving location recommendations with temporal pattern extraction. In *Proceedings 18th Brazilian Symposium on Multimedia & the Web (WebMedia)*, pages 293–296, 2012.

[31] W. Mathew, R. Raposo, and B. Martins. Predicting future locations with hidden markov models. In *Proceedings ACM Conference on Ubiquitous Computing (UbiComp)*, pages 911–918, 2012.

[32] A. Noulas, S. Scellato, N. Lathia, and C. Mascolo. A random walk around the city: New venue recommendation in location-based social networks. In *Proceedings International Conference on Privacy, Security, Risk & Trust (PASSAT), and International Conference on Social Computing (SocialCom)*, pages 144–153, 2012.

[33] A. Papadimitriou, P. Symeonidis, and Y. Manolopoulos. Friendlink: Link prediction in social networks via bounded local path traversal. In *Proceedings International Conference on Computational Aspects of Social Networks (CASoN)*, pages 66–71, 2011.

[34] A. Papadimitriou, P. Symeonidis, and Y. Manolopoulos. A generalized taxonomy of explanations styles for traditional and social recommender systems. *Data Mining & Knowledge Discovery*, 24(3):555–583, 2012.

[35] 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 4th International Conference in Ubiquitous Intelligence and Computing (UIC)*, pages 1130–1139, 2007.

[36] D. Quercia and L. Capra. Friendsensing: recommending friends using mobile phones. In *Proceedings 3rd ACM Conference on Recommender systems (RecSys)*, pages 273–276, 2009.

[37] D. Quercia, J. Ellis, and L. Capra. Using mobile phones to nurture social networks. *IEEE Pervasive Computing*, 9(3):12–20, 2010.

[38] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens: An open architecture for collaborative filtering on netnews. In *Proceedings ACM Conference Computer Supported Collaborative Work (CSCW)*, pages 175–186, 1994.

[39] J. Sang, T. Mei, J.T. Sun, C. Xu, and S. Li. Probabilistic sequential pois recommendation via check-in data. In *Proceedings 20th International Conference on Advances in Geographic Information Systems (SIGSPATIAL)*, pages 402–405, 2012.

[40] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings 10th International Conference on World Wide Web (WWW)*, pages 285–295, 2001.

[41] M. Sarwat, J. Bao, A. Eldawy, J. Levandoski, A. Magdy, and M. Mokbel. Sindbad: a location-based social networking system. In *Proceedings ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pages 649–652, 2012.

[42] M. Sattari, M. Manguoglu, I. Toroslu, P. Symeonidis, P. Senkul, and Y. Manolopoulos. Geo-activity recommendations by using improved feature combination. In *Proceedings ACM Conference on Ubiquitous Computing (UbiComp)*, pages 996–1003, 2012.

[43] S. Scellato, A. Noulas, and C. Mascolo. Exploiting place features in link prediction on location-based social networks. In *Proceedings 17th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, pages 1046–1054, 2011.

[44] S. Shridhar, M. Lakhanpuria, A. Charak, A. Gupta, and S. Shridhar. Snair: a framework for personalised recommendations based on social network analysis. In *Proceedings 5th ACM SIGSPATIAL*

International Workshop on Location Based Social Networks (LBSN), pages 55–61, 2012.
[45] P. Symeonidis, A. Papadimitriou, Y. Manolopoulos, P. Senkul, and I. Toroslu. Geo-social recommendations based on incremental tensor reduction and local path traversal. In *Proceedings 3rd ACM SIGSPATIAL International Workshop on Location-Based Social Networks (LBSN)*, pages 89–96, 2011.
[46] K. Tso-Sutter, L.B. Marinho, and L. Schmidt-Thieme. Tag-aware recommender systems by fusion of collaborative filtering algorithms. In *Proceedings ACM Symposium on Applied computing (SAC)*, pages 1995–1999, 2008.
[47] H. Wang, M. Terrovitis, and N. Mamoulis. Location recommendation in location-based social networks using user check-in data. In *Proceedings ACM Conference on International Conference on Advances in Geographic Information Systems (SIGSPATIAL)*, pages 364–373, 2013.
[48] L. Xiang, Q. Yuan, S. Zhao, L. Chen, X. Zhang, Q. Yang, and Jimeng Sun. Temporal recommendation on graphs via long- and short-term preference fusion. In *Proceedings 16th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, pages 723–732, 2010.
[49] L. Xin, X. Guandong, C. Enhong, and Z. Yu. Learning recency based comparative choice towards point-of-interest recommendation. *Expert Systems with Applications*, 42(9):4274–4283, 2015.
[50] M. Ye, P. Yin, W.C. Lee, and D.L. Lee. Exploiting geographical influence for collaborative point-of-interest recommendation. In *Proceedings 34th International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR)*, pages 325–334, 2011.
[51] Z. Yin, M. Gupta, T. Weninger, and Han J. A unified framework for link recommendation using random walks. In *Proceedings International Conference on Advances in Social Network Analysis & Mining (ASONAM)*, pages 152–159, 2010.
[52] J.C. Ying, H.C. Lu, and V. Tseng. Followee recommendation in asymmetrical location-based social networks. In *Proceedings ACM Conference on Ubiquitous Computing (UbiComp)*, pages 988–995, 2012.
[53] J.J.C. Ying, E.H.C. Lu, W.N. Kuo, and V. Tseng. Urban point-of-interest recommendation by mining user check-in behaviors. In *Proceedings ACM SIGKDD International Workshop on Urban Computing (UrbComp)*, pages 63–70, 2012.
[54] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. Magnenat Thalmann. Time-aware point-of-interest recommendation. In *Proceedings 36th International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR)*, pages 363–372, 2013.
[55] A. Zhang, A. Noulas, S. Scellato, and C. Mascolo. Hoodsquare: Modeling and recommending neighborhoods in location-based social networks. *CoRR*, abs/1308.3657, abs/1308.3657, 2013.
[56] V. Zheng, Y. Zheng, X. Xie, and Q. Yang. Collaborative location and activity recommendations with gps history data. In *Proceedings 19th International Conference on World Wide Web (WWW)*, pages 1029–1038, 2010.
[57] V. Zheng, Y. Zheng, X. Xie, and Q. Yang. Towards mobile intelligence: Learning from gps history data for collaborative recommendation. *Artificial Intelligence*, 184-185:17–37, 2012.
[58] V.W. Zheng, B. Cao, Y. Zheng, X. Xie, and Q. Yang. Collaborative filtering meets mobile recommendation: A user-centered approach. In *Proceedings 24th AAAI Conference on Artificial Intelligence (AAAI)*, 2010.

Pavlos Kefalas received his B.Sc. degree from the Department of Computer Science of the University of Create. Also, he received his M.Sc. degree from the Department of Informatics of the Aristotle University of Thessaloniki (AUTH). Currently, he is Ph.D. candidate at the same department, member of the Data Engineering Lab. His research interests include recommender systems, reasoning, data mining, algorithms, distributed databases, usage mining, content mining and graph mining.

Panagiotis Symeonidis received his B.Sc. degree in Applied Informatics (1996) and his M.Sc. degree in Information Systems (2004) from the Macedonia University, Greece. He received the Ph.D. degree (2008) in web mining from the Department of Informatics of AUTH. Currently, he is working as a postdoctoral researcher at the Data Engineering Lab of the latter department. He is co-author of 2 books and $>$35 articles in international journals and conference proceedings, whereas his articles have received $>$1000 citations. He teaches courses on databases, data mining and data warehousing in a postgraduate program in AUTH. He serves as director of the 1st SEK (School Laboratory Center) of Stavroupolis, Thessaloniki. His research interests include web mining (usage mining, content mining and graph mining), information retrieval and filtering, recommender systems, social media in Web 2.0 and time-evolving online social networks.

Yannis Manolopoulos is Professor with the Department of Informatics of AUTH. In the past, he has been with the University of Toronto, the University of Maryland at College Park and the University of Cyprus. He has also served as Rector of the University of Western Macedonia, Greece, Head of his own department and Vice-Chair of the Greek Computer Society. He has co-authored 5 monographs published by Kluwer and Springer, 8 textbooks in Greek, as well as $>$300 journal and conference papers related to Data Management. He received $>$9000 citations from $>$1300 distinct academic institutions and 3 best paper awards from SIGMOD, ECML/PKDD and MEDES conferences. He has also served as main co-organizer of several major fora, among others: ADBIS'2002, SSTD'2003, SSDBM'2004, ICEIS'2006, EANN'2007, ICANN'2010, AIAI'2012, WISE'2013, CAISE'2014, MEDI'2015 conferences. He has acted as evaluator for funding agencies in Austria, Canada, Cyprus, Czech Republic, Estonia, EU, Hong-Kong, Georgia, Greece, Israel, Italy and Russia. Currently, he serves in the Editorial Board of The VLDB Journal, The World Wide Web Journal, The Computer Journal, among others.