Discovering and Profiling Overlapping Communities in Location-Based Social Networks

Zhu Wang, Daqing Zhang, Xingshe Zhou,
Dingqi Yang, Zhiyong Yu, and Zhiwen Yu,

***Abstract*—With the recent surge of location-based social networks (LBSNs), such as Foursquare and Facebook Places, huge digital footprints of people's locations, profiles, and online social connections become accessible to service providers. Unlike social networks (e.g., Flickr, Facebook) that have explicit groups for users to subscribe to or join, LBSNs usually have no explicit community structure. In order to capitalize on the large number of potential users, quality community detection and profiling approaches are needed. In the meantime, the diversity of people's interests and behaviors when using LBSNs suggests that their community structures overlap. In this paper, based on the user check-in traces at venues and user/venue attributes, we come out with a novel multimode multi-attribute edge-centric coclustering framework to discover the overlapping and hierarchical communities of LBSNs users. By employing both intermode and intramode features, the proposed framework is not only able to group like-minded users from different social perspectives but also discover communities with explicit profiles indicating the interests of community members. The efficacy of our approach is validated by intensive empirical evaluations using the collected Foursquare dataset.**

***Index Terms*—Community profiling, hierarchical clustering, location-based social networks (LBSNs), overlapping community detection.**

I. Introduction

With the wide adoption of GPS-enabled smartphones, location-based social networks (LBSNs) have been experiencing increasing popularity, attracting millions of users. In LBSNs, users can explore places, write reviews, upload photos, and share locations and experiences with others. The soaring popularity of LBSNs has created opportunities for understanding collective user behaviors on a large scale, which are capable of enabling many applications, such as direct marketing, trend analysis, group search, and tracking.

One fundamental issue in social network analysis is the detection of user communities. A community is typically thought of as a group of users with more and/or better interactions amongst its members than between its members and the remainder of the network [1], [2]. However, unlike social networks (e.g., Flickr, Facebook) that provide explicit groups for users to subscribe to or join, the notion of community in LBSNs is not well defined. In order to capitalize on the huge number of potential users, quality community detection and profiling approaches are needed.

It has been well understood that people in a real social network are naturally characterized by multiple community memberships. For example, a person usually belongs to several social groups such as family, friends, and colleges; a researcher may be active in several areas. Thus, it is more reasonable to cluster users into overlapping communities rather than disjoint ones.

Most of the existing community detection approaches are based on structural features (e.g., links) [3], but the structural information of online social networks is often sparse and weak; thus, it is difficult to detect interpretable overlapping communities by considering only network structural information [4]. Fortunately, LBSNs provide rich information about the user and venue through check-ins, which makes it possible to cluster users with different preferences and interests into different communities. Specifically, the observation that a check-in on LBSNs reflects a certain aspect of the user's preferences or interests enlightens us to cluster edges instead of nodes, as the detected clusters of check-ins will naturally assign users into overlapping communities with connections to venues. Once edge clusters are obtained, overlapping communities of users can be recovered by replacing each edge with its vertices, i.e., a user is involved in a community as long as any of her check-ins falls into the community. In such a way, the obtained communities are usually highly overlapped.

We present an example of the user-venue check-in network in Fig. 1, which consists of five users and four venues. In such a network, users and venues are represented as two types of nodes, and each check-in is represented as an edge between a user node and a venue node. For this attributed bipartite network, since both users and venues have their own attributes, if we perform edge clustering to group users based solely on network structure [5], we can get two overlapping communities: Group 1 (Mary, Tom) and Group 2 (Tom, David, Bob, Eva). By implicitly using the venue mode to characterize the user mode (i.e., intermode), we can interpret Group 1 as a family community and Group 2 as a colleague community. However, if we consider not only the check-in network (i.e., intermode features) but also the attributes of users and venues (i.e., intramode features), we can get three overlapping communities: Group 1 (Mary, Tom), Group 2 (Tom, David), and Group 3 (Bob, Eva). In this case, even though Tom, David, Bob, and Eva have similar check-in patterns, they are

Manuscript received June 18, 2012; revised December 15, 2012; accepted March 6, 2013. Date of publication May 31, 2013; date of current version March 13, 2014. The work of Z. Wang was supported by the China Scholarship Council through joint Ph.D. funding. This paper was supported in part by the National Basic Research Program of China under Grant 2012CB316400, the EU FP7 Project Societies under Grant 257493, the National Natural Science Foundation of China under Grant 61222209 and Grant 61103063, the Specialized Research Fund for the Doctoral Program of Higher Education under Grant 20126102110043, the Natural Science Basic Research Plan in the Shaanxi Province of China under Grant 2012JQ8028, the Scholarship Award for Excellent Doctoral Student Granted by the Ministry of Education of China, and the Doctorate Foundation of Northwestern Polytechnical University under Grant CX201018. This paper was recommended by Associate Editor W. Pedrycz.

Z. Wang, X. Zhou, and Z. Yu are with the School of Computer Science, Northwestern Polytechnical University, Xi'an, Shaanxi 710129, China (e-mail: zhu.wang@mail.nwpu.edu.cn; zhouxs@nwpu.edu.cn; zhiwenyu@nwpu.edu.cn).

D. Zhang, D. Yang, and Z. Yu are with the Department of Telecommunication Network and Services, Institut Mines-TELECOM/TELECOM SudParis, Evry 91011, France (e-mail: daqing.zhang@it-sudparis.eu; dingqi.yang@it-sudparis.eu; zhiyong.yu@it-sudparis.eu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMC.2013.2256890

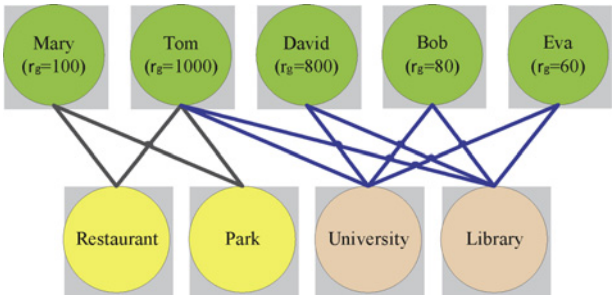

Fig. 1. User-venue check-in network example.

further grouped into two separate communities. Since Tom and David travel frequently whose radius of gyration (i.e., r_g) are 1000 km and 800 km, while Bob and Eva mainly stay locally whose r_g are 80 km and 60 km, respectively. Here, we probably can label Group 1 as a family community, Group 2 as a research staff community, and Group 3 as a teaching staff community.

Apparently, it is more reasonable to exploit both the structural information (intermode) and the node attributes (intramode) to cluster users, as we can naturally obtain communities with richer and interpretable information, even though it is a highly challenging task. While classical coclustering is one way to conduct this kind of community partitioning [6], the identified communities are disjointed, which contradicts with the actual social setting. Edge clustering has been proposed to detect communities in an overlapping manner [5], but it did not take intramode features into consideration.

From the perspective of service providers, it is equally important to identify communities with similar interests and understand what each community is interested in. In contrast to existing community detection approaches that seldom address the profiling of detected communities, we intend to take community profiling into account when designing the community detection framework. We believe that it's crucial to characterize communities in a semantic manner to effectively support real-world applications. However, due to the limitation of available node information, not much work has been done on community profiling. The rich user and venue metadata available in LBSNs, especially the hierarchical structure of venue categories, provides us the possibility to semantically characterize the identified communities.

In this paper, we aim to make the following two contributions.

1) We formulate the overlapping community detection problem in LBSNs as a coclustering issue that considers both the user-venue check-in network and the attributes of users and venues. Specifically, we detect overlapping communities from an edge-centric perspective, where each edge is viewed as a link between two modes, i.e., a user mode vertex and a venue mode vertex. While existing multimode clustering methods mainly concern the intermode features, we adopt both intermode and intramode features for clustering. By introducing different attributes of users and venues as intramode features, we show that various perspectives of social communities can be revealed.
2) We consider both community detection and profiling in one unified framework and obtain communities containing user and venue information simultaneously. In such a way, each community explicitly shows who is interested in where with what attributes, which is very useful in enabling real applications. In the meantime, we analyze and compare the detected user community profiles in London, Los Angeles and New York, with interesting findings.

The rest of this paper is structured as follows. Section II presents the related work. Section III formally defines the multimode multi-attribute overlapping community detection problem. The proposed community clustering framework is presented in Section IV, followed by experimental evaluation in Section V. Afterward, Section VI analyzes the detected communities based on community profiling. We conclude our work and discuss possible future directions in Section VII.

II. Related Work

In this section, we briefly review the related work that can be classified into three categories.

The first category contains the research on understanding the collective user behaviors based on LBSNs. Scellato *et al.* [7], [8] analyzed the social, geographic and geo-social properties of four social networks (BrightKite, Foursquare, LiveJournal, and Twitter). Noulas *et al.* [9] investigated the user check-in dynamics and the presence of spatio-temporal patterns in Foursquare. Cheng *et al.* [10] studied the mobility patterns of Foursquare users and revealed the factors affecting people's mobility. Vasconcelos *et al.* [11] analyzed how Foursquare users exploited three features (i.e., tips, dones, and to-dos) to uncover different behavior profiles. Only two studies aimed at uncovering group profiles in LBSNs. Li *et al.* [12] proposed two different clustering approaches to identify user behavior patterns on BrightKite. Noulas *et al.* [13] used a spectral clustering algorithm to group Foursquare users based on the categories of venues they had checked in, aiming at identifying communities and characterizing the type of activity in each region of a city. Although the aforementioned studies offer important insights into properties of user interactions in LBSNs, none of them worked on overlapping community detection using network links and node attributes. Our work aims to fill in this gap by discovering and profiling communities in an overlapping manner.

The second category involves the work on community detection that is a classical task in complex network analysis [1], [2], [14], [15]. In order to detect communities from a network of nodes, one typically chooses an objective function based on the intuition that a cluster is a set of nodes with better internal connectivity than external connectivity, and then applies approximate or heuristic algorithms to extract node clusters by optimizing the objective function. In general, community detection can be classified into two categories: overlapping and non-overlapping approaches. Some popular methods are modularity maximization [14], [15], Girvan-

Newman algorithm [1], Louvain algorithm [16], clique percolation [17], link communities [18], etc. As users in LBSNs have rather weak and sparse relations [19], one cannot naively apply community detection based solely on the network links and expect to generate interpretable communities.

The third category focuses on community detection by considering both links and node attributes, which are the closest to our work. Several existing works on attributed graph clustering fall into this category. The main idea is to design a distance/similarity measure for vertex pairs that combines both structural and attribute information of the nodes. Based on this measure, standard clustering algorithms such as k-medoids and spectral clustering are then applied to cluster the nodes. For instance, a weighted adjacency matrix is used as the similarity measure in [20], where the weight of each edge is defined as the number of attribute values shared by the two end nodes. The authors applied graph clustering algorithms on the constructed adjacency matrix to perform clustering. The state-of-the-art distance-based approach is the SA-cluster [21] that defined a unified distance measure to combine structural and attribute similarities. Attribute nodes and edges are added to the original graph to connect nodes that share attribute values, and a neighborhood random walk model is used to measure the node closeness on the augmented graph. Afterward, a clustering algorithm SA-cluster is proposed based on the k-medoids method.

However, all these works in the last category attempted to optimize two contradictory objective functions and intended to identify disjoint communities; thus, the communities detected were not optimal and had no clear semantic meanings. In this paper, we propose to leverage both the structure links between users and venues, and their attributes to discover the overlapping community structure. Specifically, we formulate the overlapping community detection problem into a multimode multi-attribute edge clustering issue, viewing both intermode links and intramode attributes as unified features for clustering. With this novel representation, users and venues together with their attributes are grouped in a natural way, where the detected communities have explicit semantic meanings that can be interpreted as community profiles.

III. Problem Statement

In this paper, a community is defined as a cluster of edges (i.e., check-ins) with user and venue as two modes. We use $U = (u_1, u_2, \ldots, u_m)$ to represent the user set, and $V = (v_1, v_2, \ldots, v_n)$ to denote the venue category set. A community $C_i (1 \leq i \leq k)$ is a subset of users and venue categories, where k is the number of communities. On one hand, the check-in network between users and venue categories form a matrix M, where each entry $M_{ij} \in [0, \infty)$ corresponds to the number of check-ins that u_i has performed over v_j. Therefore, each user can be represented as a vector of venue categories, and each venue category can be denoted as a vector of users. On the other hand, users and venue categories might have several independent attributes, denoted as $(a_{i1}, a_{i2}, \ldots, a_{ix})$, and $(b_{j1}, b_{j2}, \ldots, b_{jy})$, respectively. Normally, every attribute reveals a certain social aspect of users or venue categories. For instance, a user has a certain number of followers and followings in Foursquare, and a venue category has a common operating time. Therefore, both the user mode and the venue mode have two types of representations: an intermode representation and an intramode representation.

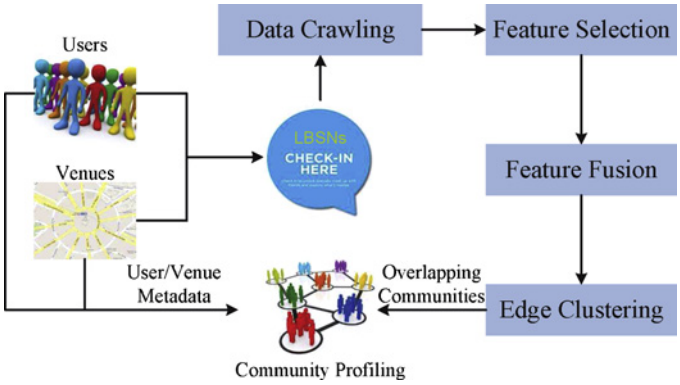

Fig. 2. Community discovering and profiling framework.

Based on the above notations, the overlapping community detection in LBSNs can be formulated as a multimode multi-attribute edge-centric coclustering problem as follows.

Input:

1) A check-in matrix $M_{(|U|\times|V|)}$, where $|U|$ and $|V|$ are the numbers of users and venue categories, respectively.
2) A user attributes matrix $M_{(|U|\times|A|)}$, where $|A|$ is the number of user attributes.
3) A venue category attributes matrix $M_{(|V|\times|B|)}$, where $|B|$ is the number of venue category attributes.
4) The number of communities k, which is optional based on the clustering algorithm.

Output:

1) k overlapping communities that consist of both users and venue categories.

IV. Multimode Multi-attribute Edge Clustering Framework

The key idea of the proposed community discovering and profiling framework is shown in Fig. 2. First, features are selected based on the characteristics of the collected LBSNs dataset and then feature normalization and fusion are performed. Second, the overlapping community structure is detected by using the proposed edge clustering algorithm. Finally, by combining the detected communities together with user/venue metadata, we obtain the community profiles to interpret the social and semantic meanings of communities.

A. Multimode Multi-attribute Edge Clustering

As stated in the introduction section, we define a community in LBSNs as a group of users who are more similar with users within the group than users outside the group. Therefore, communities that aggregate similar users and venues together

should be detected by maximizing intracluster similarity. This objective function is formulated as [5]

$$\text{Obj} = \arg\max_{C} \sum_{j=1}^{k} \sum_{e_c \in C_j} \text{sim}(e_c, C_j) \tag{1}$$

where k is the number of communities, $C = \{C_1, C_2, \ldots, C_k\}$ is the detected community set, e_c denotes an edge of community C_j, and $\text{sim}(e_c, C_j)$ is the similarity between e_c and C_j.

With the above objective function, the key issue is to characterize the similarity between an edge and a community. To this end, we first introduce the definition of edge similarity. In a user-venue check-in network, each edge is associated with a user vertex and a venue vertex. By taking an edge-centric view, each edge can be treated as an instance with its two vertices as features. In other words, the similarity between a pair of edges can be defined as the similarity between the corresponding pair of user vertices and venue vertices as

$$\text{sim}_{\text{edge}}(e_i, e_j) = F(\text{sim}_u(u_i, u_j), \text{sim}_v(v_i, v_j)) \tag{2}$$

where $\text{sim}_u(u_i, u_j)$ is the similarity between two users, $\text{sim}_v(v_i, v_j)$ is the similarity between two venues, and F represents the function used to combine these two similarities. The formalism of F depends on the characteristics of the expected communities and the targeted applications. Considering the similarity trade-off between user mode and venue mode, two widely used formalisms of F are average (i.e., $(\text{sim}_u + \text{sim}_v)/2$) and multiplication (i.e., $\sqrt{\text{sim}_u \times \text{sim}_v}$). In this paper, we adopt the second notion to ensure that a pair of edges are of high similarity if and only if they are of high similarity in both user-mode and venue-mode.

Each community contains a set of edges, based on (2), the similarity between an edge e_i and a community C_j is defined as

$$\text{sim}_{e_i, C_j} = \frac{1}{|C_j|} \sum_{e_c \in C_j} \text{sim}_{\text{edge}}(e_i, e_c) \tag{3}$$

where $|C_j|$ refers to the number of edges within community C_j.

As shown in (2), the edge similarity is defined based on two mode similarities (i.e., user-mode similarity and venue-mode similarity). In the following section, we compute the mode similarity by taking into account both intermode and intramode features.

B. Feature Description

The intermode feature describes the structure similarity between a pair of edges based on the check-in relationships between users and venues. According to [5], we adopt two intermode features (i.e., user-venue similarity and venue-user similarity) in this paper, where each user is represented as a vector of venue categories and each venue category is denoted as a vector of users. The intramode feature depicts attributes similarity where each attribute corresponds to a certain social aspect of users or venues. We select three intramode features based on the characteristics of the Foursquare data, which is partially inspired by [10].

Fig. 3. Tag clouds of two Foursquare users from London. (a) Tag cloud of user A. (b) Tag cloud of user B.

1) *Intermode Feature User-Venue Similarity:* Foursquare classifies venues into 400 categories under nine parent categories. We identify 274 venue categories by merging those similar ones, and consequently based on the venue categories that a user has checked in, each user can be represented as a vector of 274 dimensions. We build a $|U| \times 274$ matrix to represent all the active users within the collected dataset. Afterward, this matrix is refined based on principal component analysis, which is able to convert a set of observation of correlated variables into a set of values of linearly uncorrelated variables under a latent space. By applying principal component analysis on the raw matrix, we obtain a $|U| \times 100$ matrix that covers 95.62% of the total variance. After the conversion, each user is represented as a vector of 100 dimensions in the latent space.

According to [5], cosine similarity is effective in characterizing intermode feature similarities. Therefore, we adopt cosine similarity to calculate the user-venue similarity for a pair of users u_m and u_n as follows:

$$\text{sim}_{uv}(u_m, u_n) = \frac{\vec{u_m} \cdot \vec{u_n}}{||\vec{u_m}|| \cdot ||\vec{u_n}||} \tag{4}$$

where $\vec{u_m}$ and $\vec{u_n}$ refer to the feature vectors of u_m and u_n in the latent space, respectively. Here, we give an example to illustrate the concept of the user-venue similarity. First, we present the tag clouds of two Foursquare users from London (marked as user A and user B) in Fig. 3 by using Wordle (www.wordle.net). We can see that the check-in patterns of these two users are very similar, since both of them have frequent check-ins at Airport and Pub.

Second, we show in Fig. 4 the 100-dimension vectors of user A and user B in the latent space. Based on (4), the cosine similarity $\text{sim}_{uv}(A, B)$ between these two vectors can be obtained. Specifically, the corresponding value is 0.696, which indicates a high similarity between user A and user B. Apparently, the result is in line with our initial estimation based on the tag clouds of these two users.

2) *Intermode Feature Venue-User Similarity:* As we have mentioned, each venue category of Foursquare can be denoted as a vector by treating users as features as well. Following the same approach as the above section, we obtain a 274×100 matrix by performing principal component analysis on the original $274 \times |U|$ matrix, which covers 95.34% of the total variance. As a result, each venue category corresponds to a vector of 100 dimensions in the latent space. For example, Fig. 5 shows the 100-dimension vectors of two venue categories (i.e., Museum and Bar) in the latent space. Similarly, the venue-user similarity is also defined using cosine similarity. Based on these two vectors, we can calculate the venue-

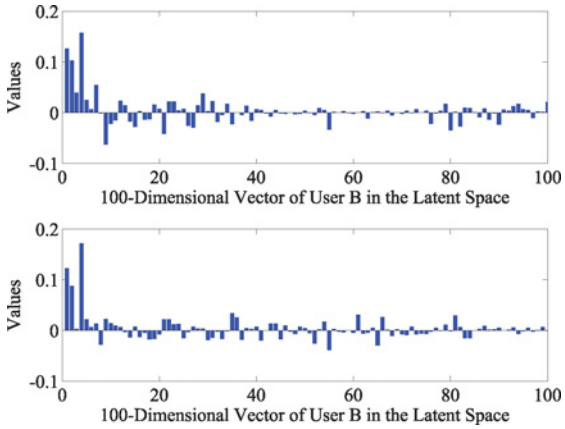

Fig. 4. 100-dimension vectors of user A and user B in the latent space.

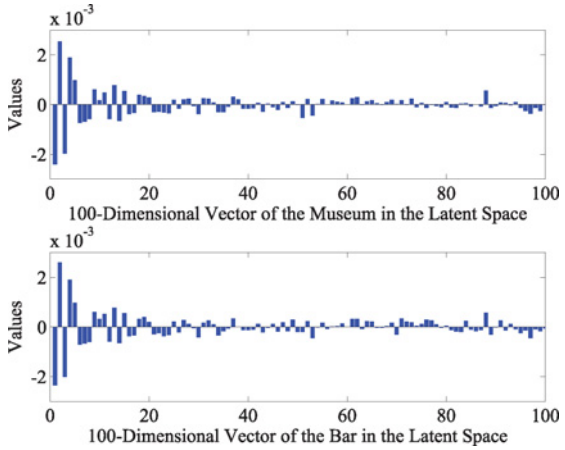

Fig. 5. 100-dimension vectors of two venue categories (i.e., *Museum* and *Bar*) in the latent space.

user similarity between Museum and Bar and the result is 0.977, which indicates that users who have ever checked in at museums are also very likely to visit bars.

3) *Intramode Feature: User Social-Influence Similarity:* There are two lists in each user's LBSN profile, a follower list and a following list. In this paper, we define a user's social influence as the ratio of her number of followers to her number of followings. Specifically, the social influence of a user u_m is formalized as

$$s_{\text{inf}}(u_m) = \frac{n_{\text{followers}}(u_m)}{n_{\text{followings}}(u_m)}. \tag{5}$$

For example, based on (5) we can calculate the social influences of user A and user B in Fig. 3, which are 0.989 and 0.700, respectively. According to the above definition, users with high social influence are those who have many followers and fewer followings. To some extent, these users act as hubs of the social network.

We introduce the first intrauser similarity feature namely user social-influence similarity based on the user social influence metric. Given a pair of users u_m and u_n, this feature is defined as

$$\text{sim}_{us}(u_m, u_n) = \frac{\min(s_{\text{inf}}(u_m), s_{\text{inf}}(u_n))}{\max(s_{\text{inf}}(u_m), s_{\text{inf}}(u_n))} \tag{6}$$

where $s_{\text{inf}}(u_m)$ and $s_{\text{inf}}(u_n)$ represent the social influence of u_m and u_n respectively. Apparently, its value falls into the interval [0, 1]. For instance, based on (6) the user social-influence similarity between user A and user B is 0.708, indicating that these two users are similar to each other from the perspective of social influences.

4) *Intramode Feature: User Geo-Span Similarity:* The user geo-span (also called radius of gyration) is another metric that can be used to distinguish the life style of different users, which is defined as the standard deviation of distances between a user's check-ins and her home location. In LBSNs, a user's home location is defined as the centroid position of her most popular check-in region [8]. The user geo-span metric is able to indicate not only how frequently but also how far a user moves. Generally, a user with low radius of gyration mainly travels locally (with few long-distance check-ins), while a user with high radius of gyration has many long-distance check-ins. The formal definition for radius of gyration is as follows:

$$r_g = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(r_i - r_h)^2} \tag{7}$$

where n is the number of check-ins made by a user, and $r_i - r_h$ is the distance between a particular check-in location r_i and the user's home location r_h. Based on (7) the geo-spans of user A and user B are computed as 2745 and 3563 km, respectively, indicating that both of them travel frequently.

By using the radius of gyration metric, we introduce the second intrauser similarity feature named user geo-span similarity. Specifically, for a pair of users u_m and u_n, the calculation of this feature is the same as (6). Again, we use the two users shown in Fig. 3 as an example, and their geo-span similarity is 0.770.

5) *Intramode Feature: Venue Temporal Similarity:* Generally, people visit and check in different kinds of venues at different time such that different venue categories can be distinguished according to their temporal check-in patterns [22]. This paper partitions a week into 168 (7×24) time slots and each time slot corresponds to 1 h in a certain day of the week, reflecting the temporal characteristic of check-ins. In such a way, we build a weekly temporal check-in band for each venue category at the hour granularity, which means each temporal band corresponds to a vector of 168 dimensions. For example, Fig. 6 plots the check-in patterns of two different venue categories: *Bar* and *Museum*. We can see that, according to the temporal check-in patterns, museums are most popular during the daytime of weekends, while bars are extremely busy on Friday and Saturday evening.

Since we have identified 274 venue categories, a 274×168 matrix is constructed and then principal component analysis is performed on this matrix, producing a new matrix of 274×20 that covers 99.92% of the total variance. Consequently, the venue temporal similarity between a pair of venues can be defined based on cosine similarity, e.g., the temporal similarity between *Bar* and *Museum* is −0.511, indicating that *Bar* and *Museum* are two quite dissimilar venue categories from the perspective of temporal check-in patterns.

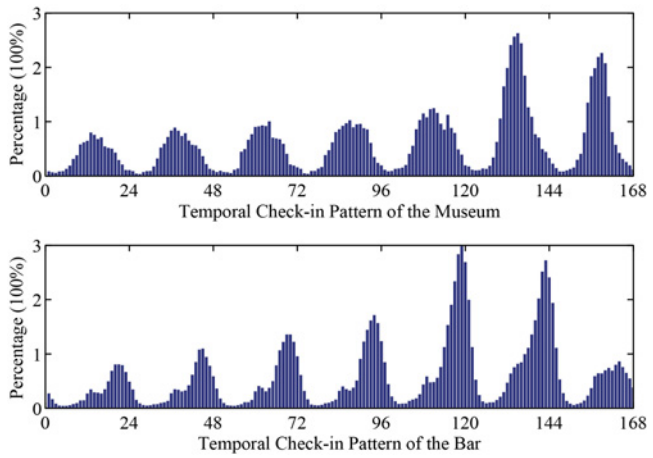

Fig. 6. Weekly check-in patterns of two venue categories: *Museum* and *Bar*.

C. Feature Normalization and Fusion

Due to the characteristic of various similarity features, different calculation methods might be used which lead to different value ranges. Therefore, the absolute values of different features must be normalized. To this end, we simply normalize each similarity measure sim_x into the interval [0, 1] as follows:

$$\text{sim}'_x = \frac{\text{sim}_x - \min(\text{sim}_x)}{\max(\text{sim}_x) - \min(\text{sim}_x)} \tag{8}$$

where sim'_x is the normalized format of sim_x. For example, before normalization the temporal similarity between *Bar* and *Museum* is -0.511, and the minimum and maximum venue temporal similarities are -0.878 and 1.000, respectively. Then, based on (8) the normalized temporal similarity between *Bar* and *Museum* will be 0.196. Similarly, the normalized user-venue similarity between user A and user B becomes 0.813, and the normalized venue-user similarity between *Bar* and *Museum* changes to 0.985. It should be pointed out that both the social-influence similarity and the geo-span similarity will not change after normalization, since the original values of these two features already fall into the interval [0, 1] according to their definitions.

Afterward, another issue is to fuse different features. Considering that each edge consists of two nodes, we first define user similarity and venue similarity as

$$\text{sim}_u = \frac{1}{|f_u|} \sum \text{sim}'_{u*} \tag{9}$$

$$\text{sim}_v = \frac{1}{|f_v|} \sum \text{sim}'_{v*} \tag{10}$$

where $|f_u|$ and $|f_v|$ represent the number of selected features for user-mode and venue-mode, respectively; sim'_{u*} and sim'_{v*} refer to the normalized similarity. For example, the similarity between user A and user B in Fig. 3 is 0.764, and the similarity between *Museum* and *Bar* is 0.591. Then, based on (2), the edge similarity is calculated as follows:

$$\text{sim}_{\text{edge}} = \sqrt{\text{sim}_u \times \text{sim}_v}. \tag{11}$$

Suppose that there is an edge $e_{A,Museum}$ between user A and *Museum* and another edge $e_{B,Bar}$ between user B and *Bar*, the similarity between these two edges can be obtained by using (11), and its value is 0.672.

Based on the above normalization and fusion mechanisms, different forms of edge similarities can be obtained by using different feature combinations, which are able to reveal the community structures in LBSNs from different perspectives.

D. Clustering Algorithm

Based on the above formulation, the multimode multi-attribute edge clustering problem is converted into an ordinary clustering issue, which can be handled by using multiple clustering algorithms. In this paper, we propose a two-step hierarchical clustering algorithm to detect overlapping communities of LBSNs users, where a variant of k-means is used as the baseline method.

To discover overlapping communities of LBSNs users, we adjust the classical k-means algorithm as follows.

1) While k-means selects the geometric center of all the instances (i.e., edges) in a cluster as its centroid, we represent each centroid by using the whole set of instances within the cluster. According to the definition of the similarity between an edge and a cluster in (2), if a set of multimode multi-attribute edges are denoted by a single vector, the obtained similarity will be significantly different.
2) While representing each centroid as a set of instances ensures the precision of the obtained similarity, the computation complexity increases from $O(k \times N)$ to $O(N^2)$. To improve the time efficiency, each centroid C_j is denoted as a structure that consists of four components: a list of current instances within the centroid (E_{C_j}), a list of instances that are assigned to the centroid during last iteration (E_{A,C_j}), a list of instances that are removed from the centroid during last iteration (E_{R,C_j}), and the similarity array between the previous centroid and the whole set of instances ($\text{sim}(E_{P,C_j}, E)$). Based on such a structure for the centroid, the computation complexity can be decreased to $O((|E_{A,C_j}| + |E_{R,C_j}|) \times N)$.

Based on the above adjustments, the proposed k-means based multimode multi-attribute edge clustering ($M^2Clustering$) method is presented in Algorithm 1.

At the beginning, k edges are randomly selected (line 1) based on which a set of initial centroids are constructed (lines 2–7). Afterward, during the iteration, given a centroid C_j we compute the similarity that each edge e_i has obtained (line 14) and the similarity it has lost (line 15) during the last reassignment, based on which the current similarity between e_j and C_j is calculated (line 16). An edge e_i will be assigned to the centroid that is most similar to itself, and the corresponding similarity is marked as maxsim_i (lines 17–20). Centroid updating is performed based on the reassignment of edges (line 23). At the end of each iteration, the current value of the objective function Obj_{cur} is calculated (line 24) to compare with the previous value Obj_{pre} (line 25). The iteration terminates if and only if the absolute difference between these two values is smaller than the predefined threshold ϵ (line 26). Experiments based on our dataset show that, in most cases, the algorithm converges within 100 iterations.

Algorithm 1 *M^2Clustering*—Edge clustering based on k-means

Input:

- E, an edge list $\{e_i | 1 \le i \le n\}$
- k, the number of communities
- M_u, the user-user similarity matrix
- M_v, the venue-venue similarity matrix

Output:

- C, a set of detected communities

k edges are randomly selected $\{e_j | 1 \le j \le k\}$
for each e_j **do**
  $E_{C_j} \leftarrow \{e_j\}$
  $E_{A,C_j} \leftarrow E_{C_j}$
  $E_{R,C_j} \leftarrow \emptyset$
  $\text{sim}(E_{P,C_j}, E) \leftarrow \text{zeros}(|E|)$
end for
$\{\text{maxsim}_i | 1 \le i \le n\} \leftarrow 0$
repeat
  $\text{Obj}_{\text{pre}} \leftarrow \sum \text{maxsim}_i$
  reset $\{\text{maxsim}_i\}$
  for each C_j **do**
    for each e_i in E **do**
      calculate $\text{sim}(E_{A,C_j}, e_i)$
      calculate $\text{sim}(E_{R,C_j}, e_i)$
      $\text{sim}(E_{C_j}, e_i) \leftarrow \text{sim}(E_{P,C_j}, e_i) + \text{sim}(E_{A,C_j}, e_i) - \text{sim}(E_{R,C_j}, e_i)$
      if $\text{sim}(E_{C_j}, e_i) > \text{maxsim}_i$ **then**
        $\text{maxsim}_i \leftarrow \text{sim}(E_{C_j}, e_i)$
        assign e_i to C_j
      end if
    end for
  end for
  update the centroids
  $\text{Obj}_{\text{cur}} \leftarrow \sum \text{maxsim}_i$
  $\Delta \leftarrow \text{abs}(\text{Obj}_{\text{cur}} - \text{Obj}_{\text{pre}})$
until $\Delta < \epsilon$

However, similar to most of the k-means based algorithms, *M^2Clustering* has several drawbacks. For example, it is sensitive to the initial centroids, since the clustering results vary according to different runs; the number of clusters (i.e., k) must be prespecified, which is not easy to deal with in case of large datasets. To overcome these drawbacks, we adopt a two-step hierarchical multimode multi-attribute edge clustering (*HM^2Clustering*) approach as shown in Algorithm 2.

At the first step, edges are clustered into a large number (K) of groups based on Algorithm 1. Without loss of generality, for a given dataset the value of K is defined as $\sqrt{|E| \times |U|}$ in this paper. These groups are then agglomerated into larger clusters using average-linkage hierarchical clustering at the second step. The reason why we use average-linkage rather than single-linkage or complete-linkage is mainly due to accuracy consideration, according to experimental results. At the end of Algorithm 2, all the edges belong to a single cluster, and the history of the clustering process is stored in a dendrogram. Community structures of multiple granularities are contained

Algorithm 2 *HM^2Clustering*—Two-step hierarchical edge clustering

Input:

- E, an edge list $\{e_i | 1 \le i \le n\}$
- K, a large number which is $\gg k$
- M_u, the user–user similarity matrix
- M_v, the venue-venue similarity matrix

Output:

- D, an edge dendrogram

invoke Algorithm 1 to generate K edge groups $\{G_j\}$
calculate pairwise similarity w_{ab} for connected edge groups G_a and G_b
repeat
  find the largest w_{ab}
  merge G_a and G_b, update related weights
until $|G| <= 1$

in this dendrogram, which can be recovered by introducing different cut thresholds. With such an method, the clustering results are less sensitive to initialization, and k is not necessary to prespecify since the hierarchical method provides results at multiple resolutions.

We adopt the above two-step hierarchical clustering approach rather than the classical hierarchical clustering algorithm because hierarchical clustering can be quite time consuming when processing large dataset.

V. Performance Evaluation

This section presents the evaluation results of the proposed overlapping community detection framework by performing experiments based on multiple feature sets. We begin with the description of data collection, followed by experiment setup and benchmark, and then present the obtained results.

A. Data Collection

Foursquare API provides limited access for retrieving check-in information; therefore, we resort to Twitter streaming API[1] to crawl the publicly shared check-ins in this paper, since approximately 20% to 25% Foursquare users choose to publish their check-ins through Twitter plug-in [9]. Our data collection started from October 24th, 2011 and lasted for eight weeks, which results in a raw dataset of more than 12 million check-ins performed by more than 700 000 users over 3 million venues. In the meantime, we also crawled metadata related to users and venues, including every user's profile and the detailed information of each venue.

B. Experiment Setup

To evaluate the performance of the proposed framework, we chose three large cities (i.e., London, Los Angeles, and New York) as the target societies. To this end, we preprocess the collected raw dataset as follows. First, we excluded check-ins that are performed over invalid venues, where invalid venues refer to those that cannot be resolved by Foursquare API. Second, we only keep users who have performed at least one

[1]Available at https://dev.twitter.com/docs.

TABLE I
DIFFERENT FEATURE SETS EVALUATED IN THE EXPERIMENTS

Feature Set	Used Features
I	UV (User-Venue Similarity) and VU (Venue-User Similarity), which is the same as Edge Clustering.
II	UV, VU and VT (Venue Temporal Similarity).
III	UV, VU, VT and US (User Social-Influence Similarity).
IV	UV, VU, VT and UG (User Geo-Span Similarity).
V	UV, VU, VT, US, and UG.

check-in per week on the average (referred to as active users), which means inactive users together with their check-ins are excluded. Finally, users who used agent software conducting remote and large scale automatic check-ins (with a check-in speed faster than than 1200 km/h, which is the common airplane speed) are defined as sudden move users [10], and check-ins from these users are eliminated as well.

After the above data cleansing, we retrieve the dataset for the three targeted cities as follows. We first calculate the home location of all the active users, and then a set of users for each city are selected based on the distance between their home locations and the geometric center of the corresponding city. Specifically, we set the distance threshold as 10 km, yielding 2408, 2596, and 3503 users for London, Los Angeles, and New York, respectively. Afterward, all the check-ins produced by these users during the data collection period are extracted, resulting 70 777, 93 010, and 108 451 check-ins, respectively. In the meantime, all the intermode and intramode features used in the experiments are calculated.

Based on the dataset of these three cities, we mainly conducted experiments to evaluate the quality of the detected communities when using different algorithms and different feature sets.

C. Benchmark

In this paper, we conducted a series of experiments to evaluate the performance of two different algorithms (i.e., $M^2Clustering$ and $HM^2Clustering$) under five different feature sets, as shown in Table I, where each feature set corresponds to a specific social perspective. Specifically, $M^2Clustering$ under Feature Set I is the same as Edge Clustering [5], which is a state-of-the-art overlapping community detection method and is used as the baseline.

D. Quality of the Detected Communities

Since we do not have the ground truth [23] about the real community structure of Foursquare data, we resort to indirectly evaluating the proposed framework as follows. First, the purpose of our framework is to cluster users who have similar habits and preferences into the same community, and we mainly leverage the check-in information. Second, the heterogeneous behaviors of users have strong intercorrelations. Intuitively, users visiting similar venues tend to share similar interests, which are reflected through the topics they discuss (i.e., tips). Therefore, we attempt to estimate the proposed community detection framework by testing whether the tips that are posted by users from the same community are also of high similarity, just as the community member's check-in patterns. In this paper, we define the average similarity among tips within a community as community tip similarity. Intuitively, a quality community detection method should achieve high community tip similarity, even though the tip information has not been leveraged when clustering communities. Particularly, a tip t_k, which is left by user u_m at venue category v_n, falls into community C_j if and only if there is an edge e_{u_m,v_n} that belongs to C_j.

Fig. 7. Tag cloud of one topic in the constructed topic model.

To compute the similarity between a pair of tips, we first project each tip to a latent topic space by using latent Dirichlet allocation (LDA), which is able to mine higher level representations (i.e., topics) from a collection of documents [24]. Specifically, LDA helps explain the similarity of tips by grouping tips into topics. A mixture of these topics then constitutes the observed tips.

We use the MALLET topic modeling toolkit to obtain the topic representation of each tip [25]. Suppose that tips are grouped into N_T topics; then a tip t_k can be formally represented as a topic vector $< tv_1, tv_2, \ldots, tv_i, \ldots, tv_{N_T} >$, where tv_i is equal to the number of words in t_k that are projected to the ith topic. Consequently, the community tip similarity can be defined by using cosine similarity.

In order to conduct the experiments, we first retrieve the tips that are left at the 2 477 122 venues in our dataset, and get a collection of more than six million tips. Afterward, non-English tips are filtered out which leads to 369 083 tips in English contributed by 66 843 users over 228 514 venues. Specifically, there are 5628, 7303, and 9911 English tips that were posted by Foursquare users from London, Los Angeles and New York, respectively.

Without loss of generality, we set the number of topics as 100 in the experiment. Fig. 7 illustrates the tag cloud of one topic.

Consequently, each tip can be denoted as a topic vector of 100 dimensions. We perform community detection using the proposed framework on the London dataset. Specifically, both the $M^2Clustering$ and the $HM^2Clustering$ algorithms were repeated 10 times for all the five feature sets listed in Table I. Then for each of the detected communities, we calculate its community tip similarity. The average community tip similarity of different feature sets is shown in Fig. 8.

According to Fig. 8, for both the $M^2Clustering$ method and the $HM^2Clustering$ method Feature Set IV is the most competitive feature set, while Feature Set V is the next most competitive one, where the user geo-span feature has been leveraged. This indicates that users who have similar geo-spans are most likely to discuss similar topics. In the meantime, while most of the introduced intramode features are able to increase the community tip similarity, the user social-influence

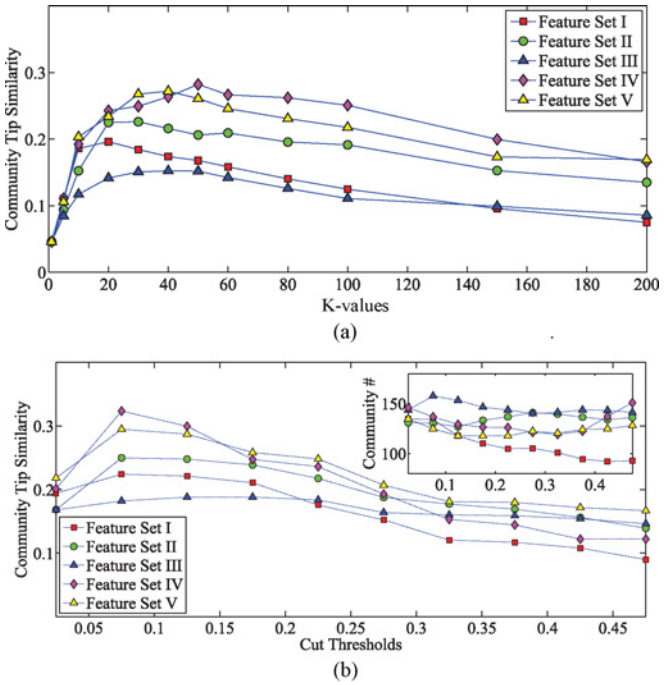

Fig. 8. Community tip similarity of London dataset. (a) Community tip similarity of $M^2Clustering$. (b) Community tip similarity of $HM^2Clustering$. The inset presents the corresponding number of detected communities under different cut thresholds.

feature tends to lower the performance, i.e., the performance of Feature Set III is worse than the baseline feature set Feature Set I. The reason might be that there is no correlation between users' social influences and their tip topics.

In the meantime, by comparing Fig. 8(a) and (b), we can find that the $HM^2Clustering$ method consistently achieves better performance than the $M^2Clustering$ method under all the feature sets, e.g., the highest community tip similarity of $HM^2Clustering$ under Feature Set IV is 0.323, while the corresponding value of $M^2Clustering$ is 0.288. The reason should be that $HM^2Clustering$ is able to reveal almost all the possible community structures by introducing different cut thresholds, and as a consequence it is more likely achieve a higher community tip similarity.

Similar results were obtained for both Los Angeles and New York datasets.

VI. Community Profiling and Analysis

The objective of community profiling is to show how the detected communities look like. In this section, we first give a community profiling mechanism based on the metadata of community members (both users and venues). Afterward, we show the usefulness of community profiling through its ability to quantitatively characterize different cities.

A. Community Profiling Based on Metadata

Community profiles are characterized by the metadata of users and venues that fall into the community, e.g., user geo-spans, user social influences and venue categories. To characterize a community, we first calculate the importance of each user and venue category based on their involvement

Fig. 9. Tag cloud of one community of London Foursquare users.

degree, and then depict the community by constructing a feature vector that summarizes the characters of a community.

Specifically, the importance of a user u_m in community C_j is quantified by the percentage of u_m's check-ins that fall into C_j, and the importance of a venue category v_n in community C_j is defined as the percentage of C_j's check-ins that is of category v_n. A user or venue category is a significant entity of a community if and only if its importance exceeds a predefined threshold θ. Without loss of generality, in this paper we use 0.1 as the importance threshold for both user and venue category. In such a way, we can obtain a list of important users and a list of important venue categories for each community, based on which a community can be represented as follows:

$$P_{C_j} = \{ \quad < f_{u_1}, e_{u_1} >, \ldots, < f_{u_m}, e_{u_m} >, \ldots, < f_{v_1}, e_{v_1} >, \ldots, < f_{v_n}, e_{v_n} >, \ldots \}. \tag{12}$$

In (12), each tuple $< f_{u_m}, e_{u_m} >$ or $< f_{v_n}, e_{v_n} >$ denotes either a user mode feature or a venue mode feature and the corresponding value. Given a community C_j, on one hand, the value of a user mode feature is calculated based on the metadata of its important users. For example, the geo-span of C_j is defined as the average of its important members' radius of gyration. On the other hand, the venue mode feature of C_j mainly refers to the significant venue categories, where each feature corresponds to a venue category v_n and its value is the importance of v_n. We use one community of London Foursquare users as an example (obtained using $M^2Clustering$ under Feature Set II where k is set as 30), which includes 195 users and 66 venue categories (its tag cloud is shown in Fig. 9). Based on the above definition, we can find that among its 195 user members there are 146 important ones, and among its 66 venue category members four of them are important venue categories (i.e., *Museum*, *Art Gallery*, *Stadium*, and *Religious Center*) and their importance values are 0.21, 0.17, 0.12, and 0.11, respectively. Thereby, this community can be profiled as {<*Social-Influence*, 1.26>,<*Geo-Span*, 523.1>,<*Museum*, 0.21>,<*Art Gallery*, 0.17>,<*Stadium*, 0.12>,<*Religious Center*, 0.11>}.

B. Revealing City Characteristics

This section presents how the proposed community profiling mechanism can be leveraged to reveal the common and different phenomenon among multiple cities. Without loss of generality, we analyzed London, Los Angeles, and New York, where communities were detected based on the $M^2Clustering$ method under Feature Set II and k is set as 30. To make the comparison manageable, we first cluster the

TABLE II
DISTRIBUTION OF GROUPS IN THREE CITIES

City / Group	London			Los Angeles			New York		
Food	453	19%	2	1700	65%	7	1913	55%	8
Transport	1236	51%	6	871	34%	4	876	25%	4
Work	1052	44%	5	791	30%	4	1011	29%	4
Art	438	18%	2	438	17%	3	509	15%	3
Store	1341	56%	4	972	37%	3	459	13%	2
Nightlife	1601	66%	6	436	17%	1	797	23%	4
Home	234	10%	1	446	17%	3	225	6%	1
Cafe	268	11%	1	581	22%	2	399	11%	1
Outdoor	290	12%	2	280	11%	2	190	5 %	1
Sport	160	7%	1	172	7%	1	447	13%	2

detected communities of each city into ten groups and then compare the characteristics of their profiles (mainly venue mode features). The results are shown in Table II, where ten groups are generated for the mentioned three cities, and each group is labeled according to the major user check-in categories. In each entry of Table II, the three numbers denote the size of a group, the ratio of group size and the total number of Foursquare users of the corresponding city, and the number of communities within a group, respectively.

By comparing the results in Table II, the characteristics of these three cities can be revealed. Generally, all the ten groups are observed even through the ratios are different. For example, while work and transport related communities cover a large portion of users, less users fall into communities of home, outdoor, and sport. This is easy to understand because people usually have more time and are more likely to check in and socialize when they eat or travel than they conduct other activities.

Interestingly, the three cities vary widely in several aspects even though all of them are English-speaking cities. For instance, 66% of the Foursquare users in London fall into the nightlife group, which is almost four times and three times as that of Los Angeles (17%) and New York (23%), respectively. Compared to London, more Foursquare users in Los Angeles (65%) and New York (55%) belong to the food group, while the percentage in London is only 19%. This might be due to the fact that London residents are more fond of staying in bars or clubs to drink and relax, while residents in the other two cities tend to spend more time in restaurants. In the meantime, more Foursquare users in London fall into the transport group, which might be explained as more people take public transport in London, while more Los Angeles and New York residents use private cars to go to work. Another finding is that more London residents use Foursquare during work, which might indicate that people are more relaxed at work in London. For the two American cities, a larger proportion of Los Angeles Foursquare users falls into the outdoor group, which should be due to the fact that the weather in New York is so cold in winter (as our data collection started from October 24, 2011 and lasted for eight weeks) that more people are willing to take part in indoor exercise rather than going outside for practise (i.e., a larger proportion of New York Foursquare users falls into the sport group).

Based on the characteristics of different communities, the company is able to provide better services and/or achieve better benefits. For example, if a new wine company plans to promote its products with limited budget, it would be more reasonable to target the relevant communities in London rather than Los Angeles or New York. In case a cafe chain plans to open a new shop in the U.S., Los Angeles should be a better choice compared to New York.

VII. CONCLUSION

In this paper, by leveraging the user-venue check-in network and user/venue attributes, we proposed a multimode multi-attribute edge-centric coclustering framework to detect overlapping communities for LBSNs users. Experimental results showed that the proposed framework was able to discover high quality overlapping communities from different perspectives and at multiple granularity, which can be used to facilitate different applications, such as group advertising and marketing. In the meantime, we reported several interesting findings obtained through community profiling and analysis.

The preliminary study suggested several interesting problems that were worth further exploring. Providing a framework to guide the selection and fusion of different features is one direction to work on. The proposed community detection framework can also help the study of friend and place recommendation mechanisms.

ACKNOWLEDGMENT

The authors would like to thank all of their colleagues for their discussions and suggestions.

REFERENCES

[1] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E*, vol. 69, no. 2, pp. 26 113–26 127, 2004.
[2] S. Fortunato, "Community detection in graphs," *Phys. Rep.*, vol. 486, nos. 3–5, pp. 75–174, 2010.
[3] Y.-Y. Ahn, J. P. Bagrow, and S. Lehmann, "Link communities reveal multiscale complexity in networks," *Nature*, vol. 466, no. 7307, pp. 761–764, 2010.
[4] J. D. Cruz, C. Bothorel, and F. Poulet, "Entropy based community detection in augmented social networks," in *Proc. IEEE CASoN*, 2011, pp. 163–168.
[5] X. Wang, L. Tang, H. Gao, and H. Liu, "Discovering overlapping groups in social media," in *Proc. ICDM*, 2010, pp. 569–578.
[6] I. S. Dhillon, "Coclustering documents and words using bipartite spectral graph partitioning," in *Proc. KDD*, 2001, pp. 269–274.
[7] S. Scellato, C. Mascolo, M. Musolesi, and V. Latora, "Distance matters: Geo-social metrics for online social networks," in *Proc. WOSN*, 2010, p. 8.
[8] S. Scellato, A. Noulas, R. Lambiotte, and C. Mascolo, "Socio-spatial properties of online location-based social networks," in *Proc. ICWSM*, 2011, pp. 329–336.
[9] A. Noulas, S. Scellato, C. Mascolo, and M. Pontil, "An empirical study of geographic user activity patterns in Foursquare," in *Proc. ICWSM*, 2011, pp. 570–573.
[10] Z. Cheng, J. Caverlee, K. Lee, and D. Z. Sui, "Exploring millions of footprints in location sharing services," in *Proc. ICWSM*, 2011, pp. 81–88.
[11] M. A. Vasconcelos, S. Ricci, J. Almeida, F. Benevenuto, and V. Almeida, "Tips, dones and todos: Uncovering user profiles in Foursquare," in *Proc. WSDM*, 2012, pp. 653–662.
[12] N. Li and G. Chen, "Analysis of a location-based social network," in *Proc. CSE*, 2009, pp. 263–270.

[13] A. Noulas, S. Scellato, C. Mascolo, and M. Pontil, "Exploiting semantic annotations for clustering geographic areas and users in location-based social networks," in *Proc. ICWSM*, 2011, pp. 32–35.

[14] A. Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very large networks," *Phys. Rev. E*, vol. 70, no. 6, pp. 66 111–66 116, 2004.

[15] K. Wakita and T. Tsurumi, "Finding community structure in mega-scale social networks," in *Proc. WWW*, 2007, pp. 1275–1276.

[16] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Statist. Mech. Theory Exp.*, vol. 2008, no. 10, p. P10008, 2008.

[17] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, no. 7043, pp. 814–818, 2005.

[18] Y.-Y. Ahn, J. P. Bagrow, and S. Lehmann, "Link communities reveal multiscale complexity in networks," *Nature*, vol. 466, no. 7307, pp. 761–764, 2010.

[19] L. Tang and H. Liu, "Community detection and mining in social media," *Synthesis Lectures Data Mining Knowledge Discovery*, vol. 2, no. 1, pp. 1–137, 2010.

[20] K. Steinhaeuser and N. V. Chawla, "Community detection in a large real-world social network," in *Social Computing, Behavioral Modeling, and Prediction*, H. Liu, J. J. Salerno, and M. J. Young, Eds. New York, NY, USA: Springer US, 2008, pp. 168–175.

[21] Y. Zhou, H. Cheng, and J. X. Yu, "Graph clustering based on structural/attribute similarities," in *Proc. VLDB Endow.*, vol. 2, no. 1. 2009, pp. 718–729.

[22] M. Ye, K. Janowicz, C. Mülligann, and W.-C. Lee, "What you are is when you are: The temporal dimension of feature types in location-based social networks," in *Proc. GIS*, 2011, pp. 102–111.

[23] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *PNAS*, vol. 99, no. 12, pp. 7821–7826, 2002.

[24] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.

[25] A. K. McCallum. (2002). *MALLET: A Machine Learning for Language Toolkit* [Online]. Available: