Modularity based Community Detection in Node Attributed Graphs using Attribute Semantics

Anonymized
Affiliation

Anonymized
Affiliation

Anonymized
Affiliation

ABSTRACT

Recent research studies showed that different node attributes have a different effect on communities in node attributed graphs. Learning this effect during community detection can characterize communities as well as improve community detection. We refer this idea as *community detection and attribute inference.* The current approaches based on this uses attribute similarity kernels that do not consider attribute semantics. Can node attribute semantics further improve such community detection and characterize communities regarding topics related to node attributes? To explore this question, this paper proposes a new algorithm for community detection and attribute inference on node attributed graphs with a kernel that exploits the *semantic similarity* of node attributes utilizing a knowledge graph. The algorithm uses a novel formulation of a modularity-based optimization function which can accommodate a semantic similarity kernel in learning node attribute effect on communities. Experimental results demonstrate a marked improvement in community detection as compared to other state-of-the-art methods. Our approach also characterized communities along given topics for four datasets.

KEYWORDS

Community detection, Community characterization, Knowledge graph, Topic-specific similarity

ACM Reference Format:
Anonymized, Anonymized, and Anonymized. 2018. Modularity based Community Detection in Node Attributed Graphs using Attribute Semantics. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, Article 4, 9 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Several social networks exhibit a modularity-based community structure [27] where edges within communities are larger than those across communities. Studying such a community structure can help understand social network dynamics[32] such as the evolution of social networks. A topic correlation to such a modularity-based community structure can help understand community formation. A research study revealed that Geo-location plays a vital role in forming conversational communities in an anonymous social network Whisper [1] [27]. Such a modularity-based community detection and topic correlation can answer questions like, "What topics should I initiate to get more attention in this location?" or "What are the user perspectives on a given topic?" It also helps understand user group dynamics, such as a set of (densely connected) users are more influenced by each other and happens to be interested in a similar topic. Applications such as Wisdom of Crowd can also benefit from such an analysis. A recent study showed that diverse virtual crowd (group) of social network users could accurately predict a top performing captain in Fantasy Premier League (FPL) [2]. A crowd that can make such a prediction should consist of individuals who are less influenced by each other [21] as well as possess a diverse perspective on a prediction task [7]. A dense community structure (indicating users are possibly influencing each other) along with its characterization (indicating users' perspective on a topic) can help build such crowds.

Several approaches are proposed for community detection on node attributed graphs [4]. A few approaches also solve the community topic correlation using generative models. However, these approaches do not identify modularity based community structure and/or characterize the communities regarding a given node attributes. Recent approaches based on community detection and attribute inference showed promising results for this [24] [31]. They selectively consider/ignore attributes based on their effect on strengthening a community structure. However, these approaches do not exploit attribute semantics in learning this effect. Moreover, they do not characterize community structure regarding topics related to node attributes.

It is crucial to consider semantics (especially a context) in computing node attribute similarities for a given topic. As an example, New York City and San-Francisco are more similar in the context of housing prices than they are in the context of geographic locations. Topic/domain specific structured knowledge graph can serve topic specific semantic similarity of given concepts. Such a similarity (topic-specific similarity) between node pairs can improve community detection accuracy and characterize communities regarding a given set of topics. As an example, consider a community detection in re-tweet conversation network of Twitter users with a set of tweets available for each user. We want to detect communities based on their conversation network and two topics; defenders (defense players) & forwards (offense players) in soccer. The resulting communities should have sets of users who often re-tweet each other as well as similar to the topic of defenders or forwards indicating their interest in soccer. A knowledge graph can indicate if two users, based on their tweet content, belong to the same cluster (central defenders) for the defenders topic. As the modularity based community detection algorithm tend to keep edges within

Conference'17, July 2017, Washington, DC, USA
© 2018 Copyright held by the owner/author(s).
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

[1] www.whisper.sh

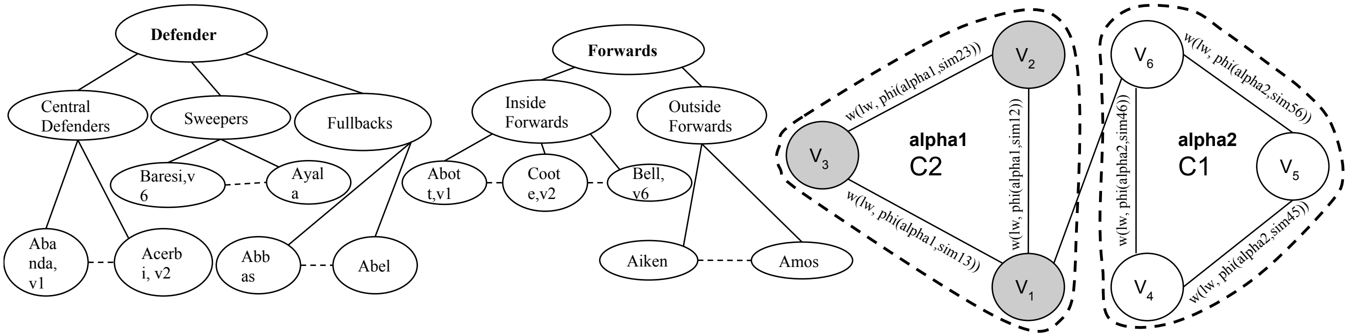

Figure 1a: Example Defenders subgraph from DBPedia

Figure 1b: Example Forwards subgraph from DBPedia

Figure 1c: Example network with V$_1$ to V$_6$ as connected nodes

Figure 1: Community detection and attribute inference Twitter user network (1c) with edges(retweet) and node attributes(user tweets). Given two topics T = (Defenders, Forwards), Identify (label nodes with community labels) and characterize communities in terms of vectors $alpha_1$, $alpha_2$. $alpha_i$ is a 2D vector representing score in terms of 2 topics in T. w is a convex function and l_w is an edge weight. phi($alpha_1$,sim_{ij}) is a knowledge graph based similarity kernel. sim_{ij} is a 2D vector each dimension representing similarity of ij computed from Defenders hierarchy(1a) and Forwards hierarchy(1b)

the same community, such a user pair is likely to be in similar community. In this paper, we propose a modularity-based community detection and attribute inference algorithm which uses the node attribute similarity values computed from knowledge graph. Our knowledge graph based similarity kernel computes a topic-specific similarity between given node attributes. The resulting topic specific similarities are represented with existing edges in a graph, indicating an edge weight when combined with a topic weight. Topic weight represents a relative importance of topics in each community. The proposed Louvain algorithm based loss function finds topic weights based on their contribution to strengthening the community structure. Topic weight vector for each community represents topic-community correlation. The resulting algorithm does not require the number of communities as input and also works for graphs with several connected components (islands). Figure 1 describes the problem specification for an example network.

We used several real-world datasets and two existing algorithms CPCD [31] and JCDC [20] to evaluate the proposed algorithm. We found that community assignment of individual edges is improved by 15% using knowledge graph-based similarity kernel to Jaccard similarity kernel (Section 5.2.1). The overall community detection accuracy (averaged over all datasets) is improved by 18% as compared to CPCD [20] and 14% as compared to JCDC [31] (Section 5.2.2). Our topic-specific similarity kernel found similarities of node pairs such that our algorithm assigned correct topic weights to communities for all four datasets. On the other hand, those similarity scores used in JCDC resulted in correct topic assignment to communities for two out of four datasets (Section 5.2.4).

Following summarizes the specific contributions of this paper,

- Topic-specific similarity kernel (Section 4.2)
- Modularity based community detection and attribute inference algorithm (Section 4.1)
- Detailed evaluation on three datasets for community detection task and role of knowledge graph (Section 5)

2 RELATED WORK

A survey by Bothorel et al. provides a good summary of node attributed graph clustering approaches [4]. They work by either merging the two pieces of information (edge and node attributes) before community detection or treat them separately during the community detection. Among recent approaches, an approach proposed by, Wang *et al.* works for non-text real valued node attributed graphs unlike several others. In an approach proposed by Qin *et al.*, two sources of information (link and node attributes) are combined at a different rate during community detection to consider or ignore information (node attributes or links) based on their effect on community detection accuracy. We refer these approaches in Table 1 as Clustering approaches which works for Node attributed graphs but do not characterize clusters (communities) regarding node attributes.

Method Class	NAG	SC	Non-text	Mod.	Topics
Clustering [4]	✓	✖	✓	✖	✖
LDA [5][19][11]	✓	✓	✖	✖	✖
SI [24]	✓	✓	✓	✖	✖
JCDC [31]	✓	✓	✓	✓	✖
Proposed	✓	✓	✓	✓	✓

Table 1: Comparison of methods for community detection in node-attributed graphs. Columns are labeled by the capabilities of each method. NAG: Node attributed graph clustering, SC: Community structure characterization, Non-text: Allows non-textual attributes. Mod.: Modularity based community structure, Topics: Given topic based structure characterization.

Several generative model-based approaches detect communities and provide latent topical characterization of these communities. An approach by Cho *et al.* works on a Mixed membership-based model for describing various types of interactions in social media [5]. Users can be grouped and characterized with a generative process utilizing author interactions and the keywords mentioned by users (representing node attributes) [19][11]. An approach proposed by He *et al.* models a network based on properly weighing attributes and links [10]. Then it finds communities and semantics of communities by jointly optimizing over node attributes and links using a generative model, similar to Wang *et al.* [28]. We refer these approaches in Table 1 as LDA based approaches. These approaches characterize a community structure regarding latent topics identified based on textual node attributes. They do not characterize communities in terms of a given set of topics by comparing node attributes in the context of those topics.

Our approach is inspired by the community detection and attribute inference proposed by Zhang *et al.* and Newman *et al.* [31] [24]. These approaches optimize the community structure based on links and the optimization process selectively considers/ignores node attributes based on their contribution to strengthening community structure. An approach by Newman *et al.* focuses on community detection for “diffusion based community" while we focus on “internal density based community" [6]. Also, their approach works for un-weighted single attributed graphs. It is referred as SI in Table 1. The approach by Zhang *et al.* (JCDC in Table 1) focuses on modularity based community detection and attribute inference [31]. As discussed, they do not use knowledge graph based similarity measure and use a different optimization technique to learn community-attribute correlation. We compare the proposed approach with this approach in Section 5. Yan *et al.* also showed that the attribute correlation with community structure helps improve the community discovery [29]. With the idea of community detection and attribute inference, Jia *et al.* showed that attribute association at a neighborhood level can help discover more accurate community structure [13]. They do not focus on community characterization.

Our motivation for integrating the semantics of node attributes into community detection is based on the prominent role of semantics in social network analysis. For example, Pool *et. al* argues knowledge graph based description can help community detection [26]. A survey on a semantic social network by Ereto *et al.* provides a good summary of the use of semantics and social networks [9]. One of the recent approaches by El *et al.* works on combining social data with semantics to create a semantic social network [8]. An approach by Ereto *et al.* also works on the similar modeling and focuses on finding communities and characterize them in folksonomies [9]. These approaches focus on integrating the social network links with existing ontologies for generic social network analysis. Community detection on such combined graphs can be biased with one graph (social graph or ontology) being very large when compared to the other. Also, we focus on applications in which community detection happens only on the social network links and hence we do not merge social graph with ontology. In an approach proposed by Palma *et al.*, they focus on predicting drug targeted Interaction using semantic similarity and edge partitioning [25]. They utilize semantic knowledge network however they do not use the knowledge graph hierarchies as proposed in this paper to compute the clustering specific distance. Moreover, their goal is to model a specific drug trafficking network where they compute drug similarities using background knowledge and add it to the existing links between source and target. The algorithm then works specifically to predict an edge than to accurately discover underlying community structure and characterize the community structure.

3 BACKGROUND

We first provide background information on modularity based community detection and domain-specific knowledge graph creation.

3.1 Modularity and Community detection

Modularity is a widely used metric to determine a quality of graph splits into communities based on a number of edges (edge density). It measures the fraction of the edges that fall within the given groups minus the expected fraction if edges were distributed at random[23]:

$$Q = \sum_{i=1}^{k} \left(e_{ii} - a_i^2 \right) \tag{1}$$

Here, k is a number of communities e_{ii} is a probability that an edge is in community i and a_i is a probability that a random edge would fall into community i. Modularity optimization techniques have been popular for such community detection [23]. Among several modularity optimization approaches, the Louvain algorithm has become popular because of its qualitative robustness and scalability [3].

The algorithm starts with each node labeled with its id as a community label. Each node, based on its neighborhood, then determines the change in the modularity of the network ΔQ that would occur if it moved to each neighboring community. The node moves (changes its label) to the community for which it has the maximum ΔQ. This process repeats until at least one node is involved in community label exchange. This label exchange phase is followed by a graph reduction in which a new graph is created with one node for each community. An edge is added between nodes with total weights of all the inter-community edges. A self-edge is added for a node with a total edge weight of all the intra-community edges. The community label exchange happens for this new graph, and the whole process is repeated for input number of iterations or until the desired modularity is achieved.

3.2 Domain specific knowledge graph

Existing knowledge graphs such as DBPedia[1] contain a significant amount of factual information. These knowledge graphs have contextual and thematic information about real-world entities represented using hierarchical relationships. Hierarchical relationships such as *skos:broader*, *rdf:type* represent concept specifications. This specification indicates “real-word" clustering information of these entities under a given topic. As an example, San Francisco and New York City are connected to the category *Populated places in the United States by metropolitan area* by *skos:broader* relationship. It indicates that New York City and San Francisco belong to the same cluster for Populated places. Topic-specific similarity computation based on an entire knowledge graph can be computationally intensive as well as gets affected by irrelevant data[15]. As an example, DBpedia contains a lot of information regarding New York City and San Francisco which is not relevant to the populated places domain (topic).

We adapt domain specific subgraph extraction technique based on domain-specificity determined by category hierarchy [15] for this purpose. The process starts with the input "root node," i.e., topic or domain of interest and extracts children of that category connected by *skos:broader* relationship. The same procedure is repeated by considering children as new root nodes for n iterations where n is an input parameter. In the extracted hierarchy, "parent" category generalizes all the "children" categories. In our work, we also extract the *wikipedia article names* belonging to each category and treat them as "children" to that category. These extracted subgraphs have been found useful in improving quality of recommendation systems and personalizations[14][15].

As an example, the resulting subgraph for “House Price Index" topic/domain has New York City and San Francisco being equidistant leaf nodes. They share a common immediate parent category of “Populated places in the United States by metropolitan area". They belong to different categories (New York City in *States of the East Coast of the United States* and San Francisco in *States of the West Coast of the United States*) in the context of Geo-locations. This indicates that the two cities are equally relevant in the context of 'House Price Index' and belong to the same cluster. In section 4.2 we detail on how we use this knowledge graph to measure the similarity.

4 APPROACH

This section presents our methodology for integrating the semantics of attribute for community detection. We start with problem formulation.

Given a social graph $G_s = (V, \epsilon)$, a knowledge graph $G_k = (E, R)$, and a set of topics $T = \{e_1, e_2, \ldots, e_p\}$, identify a community label from $C = \{c_1, c_2, \ldots, c_m\}$ for each node and a p dimensional topic weight vector for each community $c \in C$. $T(c) = \{t(c)_1, t(c)_2, \ldots, t(c)_k\}$.

In G_s, V is a set of nodes and ϵ is a set of edges. We refer this as a social graph to differentiate it from the domain-specific knowledge graph. It can be any input graph which has nodes and edge. In G_k, $r \in R$ represents a relationship between entities $e \in E$.

Each $v \in V$ is represented by a set of entities $e \in E$ such that $v = \{e_1, e_2, \ldots, e_n\}$. Each topic $e_i \in T$ is an entity in a knowledge graph G_k. $\phi(v_1, v_2, t_i)$ is a topic-specific similarity kernel. We compute a subgraph $S_{t_i} = (E_s, R_s)$ of G_k for a given topic t_i as described in section 3.2.

4.1 Community detection

In this subsection, we provide details on Louvain algorithm based label exchange and loss function to learn topic weights.

4.1.1 Algorithm. Our algorithm for solving the task is listed in Algorithm 1. As discussed, the community detection and attribute inference refer to find an effect of node attributes on community structure. In our graph model, the node attributes can be weighed based on an individual topic associated with each attribute. As an example, in Google+, each user is represented with the place he/she lives in and university he/she attended. Hence, users can be compared based on two individual topics referring to the attributes. Attributes can also be weighed along a given set of topics. In Twitter user example we discussed Section 1, we weighed Twitter users based on their tweets along two topics. We find an effect of these topical attribute associated with each node on community structure(vector T). Our algorithm starts with finding edge based clusters(communities) considering all attributes(topics) contributing equally to the community structure(Label exchange phase) and then learns which attribute can strengthen the community structure(topic weight learning). The learned vector T represents the relative importance of each topic in each community. The edges are weighed based on the newly learned vector ($EdgeWeightUpdate()$) and we look for a community structure on the modified graph with properly weighted edges. The algorithm runs until maximum iterations or until desired modularity is achieved. *GraphCondese* refers to creating one node for each community and edges based on intra-community and inter-community edges as discussed in 3.1. The algorithm is implemented in Java and implementation can be found here. We elaborate on the mechanics of the two phases next. The implementation is available at https://goo.gl/j1Mk8Y.

```
for Max iterations OR mod ≤ threshold do
    for Max out iterations do
        while LabelExchange do
            forall $v \in V$ do
                forall $c \in \eta(v)$ do
                    gain[c, val] = $\frac{2\sum_{\forall j \in N(v,c)} w(v,j,c)}{m} - \frac{k_v a_c}{2m^2}$.
                end
                Label[$v$] = Label(max(gain))
            end
        end
        GraphCondense()
    end
    while gradient convergence do
        forall communities $(c \in C)$ do
            forall topics $(t \in T)$ do
                forall edges $((i,j) \in c)$ do
                    $grad\_edge[c][t][ij] =$
                      $argmax_t \frac{1}{2m_t} \sum_{\forall (i,j) \in c} w_t(i,j,c) - \frac{k_{it} k_{jt}}{2m_t}$
                end
                $grad[c][t] = \sum_{ij \in c} grad[c][t][ij]$
            end
            $grad[c] = \alpha * grad[c]$
        end
    end
    EdgeWeightUpdate()
end
```

Algorithm 1: Community detection and attribute inference algorithm

4.1.2 Label exchange. The label exchange is based on Louvain algorithm. We calculate edge weight between two nodes v_1 and v_2 for a given community c_1 and its weight vector $T(c)$ as following,

$$w_{v_i, v_j, c_1} = \sum_{i=1}^{p} \left(l_w - e^{-(\phi(v_1, v_2, t_i) \cdot t(c_1)_i)} \right) \tag{2}$$

Here, $\phi(v_1, v_2, t_i)$ is a similarity kernel that computes topic t_i specific similarity between nodes v_1 and v_2. $t(c_1)_i$ is a topic weight of i^{th} topic in community c_1. l_w is a hyper-parameter and represents an edge weight. Specifically, it represents a relative importance of an edge to topic-specific similarity between nodes. l_w values larger than 1.0 represents the edges are considered more important than the attribute similarities.

Following Louvain algorithm, each node $v \in V$ updates its community label from a set of community labels $\eta(v) \bigcup c_v$. Here, $\eta(v)$ is a set of neighboring communities of the node v and c_v is the current community label. A node updates its community label to the one for which the node finds the maximum modularity gain. Modularity gain is calculated as following,

$$\Delta Q_{i,c} = \frac{2\sum_{\forall j \in N(i,c)} w(i,j,c)}{m} - \frac{k_i a_c}{2m^2} \tag{3}$$

$\Delta Q_{i,c}$ is a modularity gain of moving a node i to community c. $N(i,c)$ is a set of neighboring nodes of i that belong to a community c. We consider each edge as an interaction between two nodes. If the community of source node, c_1 differs from the community of a target node, c_2 (edge between v_1 and v_6 in Figure 1) then we compute edge weight by adding $w(i,j,c_1)$ and $w(i,j,c_2)$. If the source and target nodes share the same community (edge between v_1 and v_2 in Figure 1), then edge weight is computed as $2w(i,j,c_v)$. It is multiplied by 2 since we are considering all the edges within a community. Based on this, the node degree k_i is defined as following,

$$k_i = \sum_{\forall c \in (\eta(i) \bigcup cur)} \sum_{\forall j \in N(i,c)} w(i,j,c) + w(i,j,cur) \tag{4}$$

Here cur is the current community of node i. Total edge weight m is calculated as following,

$$m = \sum_{\forall i \in V} k_i \tag{5}$$

Here, k_i represents a degree of node i.

4.1.3 Topic weight update. In this step, we update each community's ($c \in C$) Topic weight vector $T(c)$. This modularity optimization is based on Louvain algorithm. Each topic weight is updated independently of the other topic weights using gradient ascent. In other words, a modularity optimization is carried out for a graph with edges and edge weight computed with topic-specific similarities of the node pair. All the topic weights are initialized with the same value, and the topic with the highest value at the end of Topic weight update represents the most relevant topic in that community. Following formalizes the topic weight update,

$$t(c)_l = argmax_{t(c)_l} \frac{1}{2m_l} \sum_{\forall (i,j) \in c} w_l(i,j,c) - \frac{k_{il} k_{jl}}{2m_l} \tag{6}$$

Here, $w_l(i,j,c)$ represents a topic specific edge weight which is calculated as following,

$$w_l(i,j,c) = \left(l_w - e^{-(\phi(v_1,v_2,t_l) \times t(c)_l)}\right) \tag{7}$$

k_{il} and k_{jl} represents the node degrees computed with the l^{th} topic weight and m_l is a summation of node degree with l^{th} topic weight.

4.2 Topic-specific similarity kernel

Here we describe the topic-specific similarity kernel $\phi(v_1, v_2, t_i)$ where $v_1 = \{e_1, e_2, \ldots, e_n\}$ and $v_2 = \{e_2, \ldots, e_l\}$. The similarity kernel uses a $S_{t_i} = (E_s, R_s)$ topic-specific subgraph to compute the similarity where each $e_i \in E_s$. A subgraph S_{t_i} represents topic relevancy of each node as number of hops from root node to each $e \in v$. If all the $e \in v_1$ and $e \in v_2$ belong to the same hierarchy in S_{t_i} then they have a higher similarity than they not belonging to the same hierarchy. The real valued similarity score should reflect both (topic relevancy and similarity) for a given edge (v_1, v_2).

As discussed in section 3.2 the knowledge graph hierarchy represents concept specifications. For each concept $e \in v_1, v_2$, we compute n-hop *parent nodes* i.e. concepts generalizing current concept. For the experiments in this paper we chose $n = min(4, concept\_to\_root)$ where $(conept\_to\_root)$ represents the number of hops from current concept to root node. Each node set is extended with these parent concepts resulting in, $v_{1ext} = \{e_1, ep_1, \ldots, ep_n\}$ and $v_{2ext} = \{e_2, ep_2, \ldots, ep_l\}$ where ep_i represents 1-hop, 2-hop or 3-hop parents.

The common concepts between v_{1ext} and v_{2ext} represents the similarity between v_1 and v_2 i.e. if they have more common concepts then they are more similar. However, it may happen that v_1 and v_2 have more common concepts but less relevancy in the topic i.e. v_1 and v_2 are very far from the root topic. Hence, next we compute the topic relevancy for each $e \in v_{1ext}, v_{2ext}$ using following equation.

$$CS = \frac{CW}{pl * d} \tag{8}$$

CW represents the weight of each concept. As an example, for a Twitter user, CW represents a number of times a concept was mentioned in user's tweets. Immediate parents are more relevant to a concept as compared to two-hop parents hence the CS (concept score) is weighted by both pl(parent level) and d(depth). d represents the distance (number of hops) of e from the topic root node and pl is the distance (number of hops) of a concept being scored from e.

Next, we compute an overlap between v_1 and v_2 using following equations,

$$WJ(c_i, c_j) = min(c_i, c_j) * \left(1.0 - abs\left(\frac{c_i - c_j}{c_i + c_j}\right)\right) \tag{9}$$

$$Score = \frac{\sum_{i,j \in m} WJ(c_i, c_j)}{\sum_{i,j \in all} WJ(c_i, c_j)} \tag{10}$$

Here, m represents common concepts of v_{1ext} and v_{2ext} while all represents all concept pairs. WJ(weighted Jaccard) measures the similarity between two concepts where each concept is represented by the CS computed in the previous step. If two concepts being compared have a similar score, then they belong close to each other in the hierarchy and should get a higher similarity value. If they are far from the root node or the matching concepts are n-hop parents of a concept, then they have a less CS and should get a less similarity value. $min(c_i, c_j)$ is multiplied to capture this.

The measure (kernel) is a variant of Jaccard similarity where instead of comparing only given user attributes, we compare "parents" of these attributes as well. In Section 5.2.1 we compare the proposed similarity measure to Jaccard similarity measure for edge weight computation.

Each edge (node pair) in G_s is characterized along a given set of topics using this similarity measure. For p topics, each edge is represented with p real values. The number of concepts in topic-specific subgraphs, extracted from DBpedia, varies with the type of topic. This variation can affect the similarity scores. As an example, various concepts extracted from user tweets compared using a large topic-specific subgraph reveals a different score than being compared using a small subgraph. This difference may not be because of the topic relevancy but because of the domain representation in DBpedia. Hence, we may not be able to compare these scores. We address this by normalizing each topic score of all the node pairs (edges) based on their Z values. In other words, each edge is now represented by the Z score for p topics. Here, each topic score of an edge is computed using mean and standard deviation of all the edge scores for that topic.

5 EVALUATION

In this section, we discuss the datasets, evaluation measure, existing works used for comparison, results, and analysis.

5.1 Datasets and Evaluation measures

The first two datasets are used for social circle discovery application where the ground truth circles are provided [18]. Circle discovery refers to identifying social circles for a user (ego) given his/her friends/followers (alters).

We use this application for evaluation as it has been identified that circles are formed by densely-connected sets of alters [18] [23]. Also, a user expects to form a circle of friends sharing an attribute [18]. Community detection and attribute inference algorithm iteratively find a set of circles refined based on the attributes such as city, university, etc.

5.1.1 Facebook ego network. This dataset (FB) has ego networks of 10 Facebook users. As we focus on the non-overlapping community detection, we used 6 users' ego networks which do not have overlapping community assignments. Each user is described by a set of binary features. As we did not have information regarding specific feature type, we used community detection on this dataset for evaluating the community detection and attribute correlation (Section 4.1) without the topic-specific similarity kernel.

5.1.2 G+ ego network. It has 20 ego networks without overlapping communities. Each user is characterized by four features: job title, current place, university, and workplace. We use each feature as a topic and characterize the communities/circles based on these four features. We extracted four domain-specific subgraphs from DBpedia with "root node" *Category:Occupations*, *Category:Companies_by_country_and_industry*, *Category:Countries*, *Category:Universities_and_colleges_by_country* for the respective topics. As we have a specific attribute categorized, we compare these attributes only in their respective domain-specific sub-graph.

5.1.3 Twitter. We used a Fantasy Premier League (FPL) related tweets and users' fantasy team configuration dataset [2]. These users play FPL where they build their own "Fantasy" soccer team and earn points based on it. We found all the re-tweet edges between these users, which represent information agreement between users. We used DBpedia spotlight [22] to find soccer player mentions from these users' tweets. We consider user as a node, retweet as an edge, player mentions as node attributes.

Users tend to have different kind of teams (more defensive or more offensive) based on their perception of FPL and strategy to get more points. Creating a virtual group of users with diverse perspectives on a game can help prediction in FPL tasks. A dense re-tweet network of these users indicates "similar" users in FPL context. If these users are also mentioning players of one type (Defenders, Forwards, Mid-fielders) more often than the other, they are likely to have that type of team, i.e., a team with more Defenders, Forwards or Mid-fielders. We consider three topics Forwards, Defenders, Mid-fielders for community detection and attribute inference. The edge weight for each topic is found using topic-specific Knowledge graphs (Defenders, Forwards, Mid-fielders).

We categorize these users into same three categories as topics. We created ground truth circles using these users' actual team configuration. Users with more than usual [2] number of players for any position are included in that circle. The dataset is available to download with code.

5.1.4 Lawyernet. We use the Lawyernet dataset [16], friendship network for 71 lawyers. Each lawyer is annotated with six attributes: status, gender, years with the firm, practice, practice, and law school attended. As each non-numeric attribute has either two or three option, it did not require knowledge graph based comparison. Hence, we use this dataset for evaluating the community detection and attribute inference algorithm (Section 4.1). We use the friendship network and ground truth values as described in JCDC [31].

5.1.5 DBLP. We use the DBLP four area dataset [12] for evaluating the community characterization. It is a co-author network, and each author is characterized by a set of keywords. Authors are categorized in Machine learning, Data mining, Databases, and Information retrieval. We use the co-authorship network between authors of two communities, Data mining, and Machine learning. We use the knowledge graph generated with root nodes *Category:Data_Mining* and *Category:Machine_Learning* for comparing authors based on their keywords. Author keywords are translated to the DBpedia concepts using DBpedia lookup API[3].

5.1.6 Evaluation Measures. We use community F-Measure and Jaccard measure used by Yang *et al.* to evaluate similar datasets for community detection task [30]. It essentially finds agreement between the community structure discovered and ground truth community structure. The evaluation function is,

$$\frac{1}{2\,|C^*|}\sum_{C_i^* \in C^*} \overset{max}{C_j} \in C\delta\left(C_i^*, C_j\right) + \frac{1}{2\,|C|}\sum_{C_j \in C} \overset{max}{C_i^*} \in C^*\delta\left(C_i^*, C_j\right) \quad (11)$$

Here, $\delta(C_i^*, C_j)$ is a similarity measure. We considered Jaccard and F-score similarity (F-Measure). We chose not to compare NMI as few of the nodes in FB and G+ datasets did not have a ground truth community label. Also, the Twitter dataset had several connected components in its re-tweet network and three circles to divide the network. Hence, we did not use NMI comparison for that dataset.

5.2 Results and Analysis

We compared our approach with Liu *et al.* as it is one of the latest community detection approaches in node attributed graphs which does not focus on community characterization [20]. We also compare with Zhang *et al.* as it solves the similar modularity based community detection and attributed inference problem but without using topic-specific similarity kernel and with a different optimization procedure for attribute correlation [31].

[2] http://www.soccer-training-guide.com/soccer-formations.html#.Wmk6GZM-eAI
[3] http://wiki.dbpedia.org/projects/dbpedia-lookup

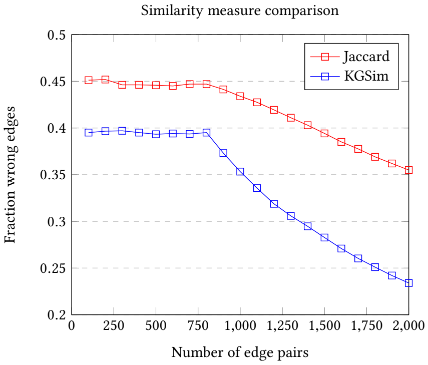

Figure 2: Similarity measure comparison. Topic-specific similarity kernel works better than Jaccard similarity for categorizing individual user edge pairs

5.2.1 Similarity measure comparison. Several community detection approaches use similarity measures such as Jaccard similarity for comparing node attributes. We compare topic-specific similarity kernel with Jaccard similarity for computing edge similarity scores. We use the Twitter dataset in this experiment. Two sets of user pairs are created. set1 = $\{s_1, s_2, \ldots, s_n\}$ where each $s_i = \{(u_1, u_2)|u_1andu_2 \in samecircle\}$. set2 = $\{d_1, d_2, \ldots, d_n\}$ where each $d_i = \{(u_1, u_2)|u_1andu_2 \in differentcircles\}$. We compare each $s_i \in$ set1 to all the $d_i \in$ set2 resulting with n_2 comparisons. Ideally, each $s_i \in$ set1 should be higher than all the $d_i \in$ set2. The number of times $s_i \in$ set1 is lower than $d_i \in$ set2 is computed as number of "inconsistencies". Figure 2 plots the "inconsistencies" to the total comparison (n_2) ratio.

Topic-specific similarity results in less "inconsistencies" compared to Jaccard similarity. The sudden dip at edge pairs=800 is due to edges being added from Forwards, and Midfielders circles. Both similarity measures performed better for Forwards and Midfielders compared to Defenders circle. However, the difference between the two similarity measures is quite evident with an increase in the number of edges. The similarity measure affects a large portion of input graph to the community detection.

5.2.2 Community detection accuracy. Next, we evaluate community detection accuracy with and without using the topic-specific similarity Table 2. We found the proposed algorithm performing slightly better regarding F-Measure and Jaccard than CPCD and JCDC for Lawyernet dataset. Unlike the FB and G+ datasets, all the community labels were provided for the Lawyernet dataset. Hence, we computed NMI scores for this dataset. Proposed algorithm achieved an improved NMI over both JCDC and CPCD. The F-Measure and Jaccard scores reported for FB (and G+) are averaged over all the 6 (and 20) ego networks. The proposed algorithm outperformed JCDC and CPCD for FB dataset as well.

As we had the node attribute values for the G+ dataset, we computed attribute similarities using knowledge graph and using Jaccard for JCDC and the proposed algorithm. KGCD-K refers similarities computed without using knowledge graph, and JCDC+K refers node attribute similarities computed using knowledge graph. We found that topic-specific similarity kernel improves F-Measure and Jaccard scores for JCDC over using Jaccard similarity. It did not result in a large improvement (as compared to Twitter) for the proposed approach as several users did not have the attribute values.

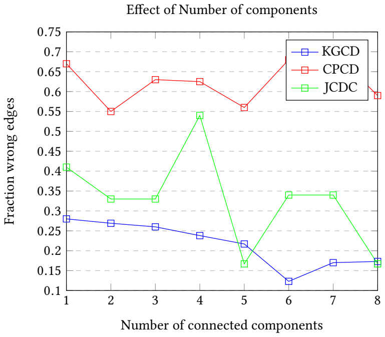

Figure 3: Effect of connected components. As the proposed algorithm treats each connected component separately, the initial community structure is less affected by the presence of connected components as compared to the other two algorithms.

For Twitter dataset, CPCD was provided a representation of each Twitter user with a vector which indicates the number of times a player was mentioned by that user. The Twitter dataset had several connected components, and we had three circles (Forwards, Defenders, Midfielders) to divide the users. Hence, we merged the communities for our approach based on the topic affinities. All the communities which had the highest score for defenders were merged (same for mid-fielders and forwards). We saw a relatively large improvement in F-Measure and Jaccard scores for using the topic-specific similarity. Also, the proposed algorithm was able to correctly label the communities with the most number of defenders, forwards, and mid-fielders by computing the maximum value for the respective topic in that community. We found JCDC labeled two communities (identified the highest score for) "Defense" topic and did not output any community with the highest score for "Forwards" topic. One of the reasons for relatively less F-Measure, Jaccard score and community characterization accuracy for this dataset is the presence of connected components. As both, CPCD and JCDC requires the number of communities as input, they often end up processing nodes belonging to different connected components for community label assignment and (for JCDC) attribute inference which affects the F1 and Jaccard scores. Next, we investigate the effect of connected components in a graph for community detection.

5.2.3 Effect of connected components. , For next set of experiments, we generated a random graph (nodes=30, edges=88) using SNAP [17]. The nodes were labeled with dummy attributes, all having nearly equivalent values such that the node attributes do not weigh in community detection. We did not choose to have equal values for node attributes since CPCD had an issue processing such graph for community detection. We found 2 communities with all three algorithms for this initial graph referred to as "initial community structure." We started adding connected components to this initial graph (nodes=10, edges=25) such that each component had an apparent community structure of two communities. For CPCD and JCDC we increased the number of communities to be discovered by 2 with the addition of each component. We expect the initial community structure to not change in the presence of other connected components. We measure this *Change_Rate* with each added component as a ratio of, number of edges that changed

Algorithm	LawyerNet			FB		G+		Twitter	
	F-Measure	Jaccard	NMI	F-Measure	Jaccard	F-Measure	Jaccard	F-Measure	Jaccard
CPCD	0.86	0.79	0.5	0.45	0.35	0.56	0.46	0.34	0.29
JCDC	0.88	0.8	0.54	0.46	0.38	0.58	0.48	0.33	0.28
JCDC+K	N/A	N/A	N/A	N/A	N/A	0.6	0.52	0.38	0.32
KGCD-K	0.89	0.81	0.58	0.51	0.42	0.68	0.56	0.39	0.32
KGCD	N/A	N/A	N/A	N/A	N/A	0.71	0.6	0.41	0.34

Table 2: Community discovery results. The F-measure and Jaccard scores were improved by 9% than the best performing baseline for datasets that do not require a use of topic-specific similarity kernel and 18% for datasets that benefits from topic-specific similarity kernel

their community from the initial community structure to the total number of edges in the initial graph.

Figure 3 plots *Change_Rate* for three algorithms. As we discussed, both CPCD and JCDC process nodes in a graph for community label exchange regardless of the components they belong to. This affects the community assignment of the initial graph. The number of edges that change their communities in the presence of islands is quite high as compared to the proposed approach. For our algorithm, the number of communities depends on the total edge weights in the graph ($\frac{1}{2m}$ term). As the number of edges increases, some nodes tend to change their community label assignment based on the change in a number of communities. Hence, we get a non-zero *Change_Rate* for our algorithm.

5.2.4 Topic correlation. Next, we evaluate the community topic correlation. We considered users from two communities based on their ground truth community labels. We want to categorize these users into communities and characterize each community regarding two topics where a topic refers to a community type. We selected Twitter users from "Forwards" & "Defenders" communities, G+ users from "University" & "Workplace" communities and DBLP authors from "Data Mining" & "Machine Learning" communities. We considered two G+ ego networks (referred as G+1 and G+2) for which we distinguished two ground truth communities based on "University" and "Workplace" attributes. In Table 3, columns Community1 and Community2 shows normalized scores for topic 1 and topic 2 between all the user/author pairs belonging to that community (based on ground truth community label) and have an edge between them. Similarly, "Inter-community" shows these scores for authors belonging to different communities and have an edge between them. The author pair scores are computed using topic-specific similarity. It was able to assign similarity values such that each community gets a higher similarity to one topic than the other.

Columns for JCDC and KGCD shows the community topic correlation, F-Measure score, and Jaccard similarity results. JCDC was able to compute topic affinity scores such that it labels Community1 and Community2 by different topics for G+2 and DBLP datasets. However, it failed to do so for Twitter and G+1 dataset. A possible reason for which is JCDC attribute correlation. It depends on maximizing intra-community edge weights (according to loss function in [31]). Hence, it ends up computing the highest score of the topic for which it has the maximum intra-community edges. On the other hand, the proposed algorithm correlates the topic scores based on loss function that optimizes the whole community structure and hence it was able to identify distinguishing topic labels for each community. We also found that such an optimization process for topic weights computation affects the F-Measure and Jaccard scores.

5.2.5 Hyper-parameters effect. We treat l_w (a relative edge weight) and $Maxiterations$(a number of times label exchange and topic weight computation is repeated) as hyper-parameters. In this subsection, we describe the effect of each hyper-parameter on F_1 community score. Figure 4a-4d depicts results for maximum main iterations and Figure 5a-5d for l_w. F-Measure score reported for G+ and FB datasets is an average over all the user ego networks. Several user ego networks, Twitter user network, and Lawyernet were not large because of which we achieved a good F1 score within first few iterations. $l_w < 1.0$ indicates that attribute similarities are considered more important than the edge similarity. For FB dataset, we get the best F1 score for the $l_w = 2.0$. For G+ and Twitter dataset, we got the best results for $l_w < 1.0$. It refers that topic-specific similarity kernel can assign accurate values hence they were found more important in discovering underlying community structure. Lawyernet friendship network resembled their status based on which the ground truth values were generated [31]. Hence, edge weight was found more important in community detection than attribute similarity scores.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we developed a topic-specific similarity kernel and a community detection algorithm. Our results showed that topic-specific similarity computed using knowledge graph could accurately compute edge weights as well as improved overall community detection accuracy. Our modularity based community detection and attribute inference algorithm further improved the community detection accuracy using topic-specific similarities. The resulting algorithm was able to label communities regarding given topics. An exciting direction to explore is to use such a similarity measure to identify "path-based" and "diffusion-based" communities.

REFERENCES

[1] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. *The semantic web* (2007), 722–735.

[2] Shreyansh Bhatt, Brandon Minnery, Srikanth Nadella, Beth Bullemer, Valerie Shalin, and Amit Sheth. 2017. Enhancing crowd wisdom using measures of diversity computed from social media data. In *Proceedings of the International Conference on Web Intelligence*. ACM, 907–913.

[3] Vincent D Blondel, Jean-Lou Guillaume, Renaud Lambiotte, and Étienne Lefebvre. 2011. The Louvain method for community detection in large networks. *J of Statistical Mechanics: Theory and Experiment* 10 (2011), P10008.

[4] Cecile Bothorel, Juan David Cruz, Matteo Magnani, and Barbora Micenkova. 2015. Clustering attributed graphs: models, measures and methods. *Network Science* 3, 3 (2015), 408–444.

Dataset	Community1			Community2			Inter-community		JCDC			KGCD		
	T_1	T_2	U_n	T_1	T_2	U_n	T_1	T_2	(C_1T_1,C_1T_2)	(C_2T_1,C_2T_2)	(F_1, Jcc)	(C_1T_1,C_1T_2)	(C_2T_1,C_2T_2)	(F_1, Jcc)
Twitter	6.15	6.48	450	4.93	4.94	150	1.83	2.2	(.168,.172)	(.154,.162)	(.51,.41)	(.285,.226)	(.156,.135)	(.58,.53)
G+1	2.2	2.18	80	1.9	0.5	30	0.8	0.6	(.362,.316)	(.381,0.125)	(.6,.53)	(.263,.285)	(.316,.262)	(.7,.6)
G+2	3.2	1.3	40	0.6	3.8	31	0.03	0.02	(.482,.304)	(.12,.53)	(.7,.65)	(.536,.139)	(.21,.54)	(.72,.65)
DBLP	2.69	3.25	338	0.81	0.29	230	1.2	1.18	(.32,.5)	(0.232,0.162)	(.56,.48)	(.377,.691)	(.319,.143)	(.64,.55)

Table 3: Community topic correlation. Topic-specific similarity kernel combined with Louvain algorithm based loss function improves the community topic labeling accuracy as compared to JCDC.

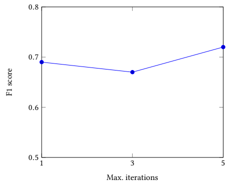
Figure 4a: G+ dataset effect of Max. iterations

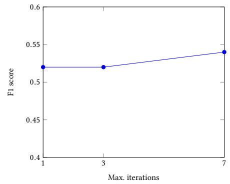
Figure 4b: FB dataset effect of Max. iterations

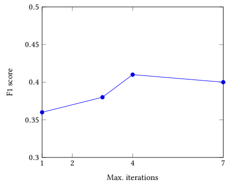
Figure 4c: Twitter dataset effect of Max. iterations

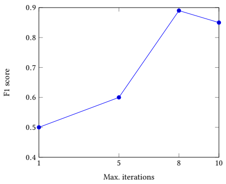
Figure 4d: Lawyernet dataset effect of Max. iterations

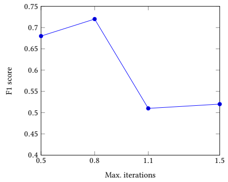
Figure 5a: G+ dataset effect of l_w

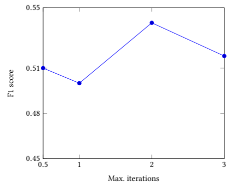
Figure 5b: FB dataset effect of l_w

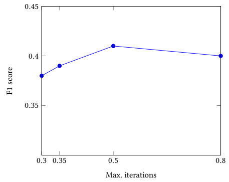
Figure 5c: Twitter dataset effect of l_w

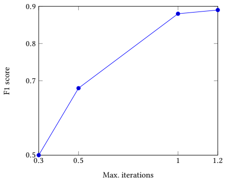
Figure 5d: Lawyernet dataset effect of l_w

[5] Yoon-Sik Cho, Greg Ver Steeg, Emilio Ferrara, and Aram Galstyan. 2016. Latent space model for multi-modal social data. In *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 447–458.

[6] Michele Coscia, Fosca Giannotti, and Dino Pedreschi. 2011. A classification for community discovery methods in complex networks. *Statistical Analysis and Data Mining: The ASA Data Science Journal* 4, 5 (2011), 512–546.

[7] Clintin P Davis-Stober, David V Budescu, Jason Dana, and Stephen B Broomell. 2014. When is a crowd wise? *Decision* 1, 2 (2014), 79.

[8] Asmae El Kassiri and Fatima-Zahra Belouadha. 2015. Towards a unified semantic model for online social networks analysis and interoperability. In *Intelligent Systems: Theories and Applications (SITA), 2015 10th International Conference on*. IEEE, 1–6.

[9] Guillaume Erétéo, Michel Buffa, Fabien Gandon, and Olivier Corby. 2009. Analysis of a real online social network using semantic web frameworks. *The Semantic Web-ISWC 2009* (2009), 180–195.

[10] Dongxiao He, Zhiyong Feng, Di Jin, Xiaobao Wang, and Weixiong Zhang. 2017. Joint Identification of Network Communities and Semantics via Integrative Modeling of Network Topologies and Node Contents. (2017). https://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14612

[11] Thanh Ho and Phuc Do. 2015. Discovering Communities of Users on Social Networks Based on Topic Model Combined with Kohonen Network. In *Knowledge and Systems Engineering (KSE), 2015 Seventh International Conference on*. IEEE, 268–273.

[12] Ming Ji, Yizhou Sun, Marina Danilevsky, Jiawei Han, and Jing Gao. 2010. Graph regularized transductive classification on heterogeneous information networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 570–586.

[13] Caiyan Jia, Yafang Li, Matthew B Carson, Xiaoyang Wang, and Jian Yu. 2017. Node Attribute-enhanced Community Detection in Complex Networks. *Scientific Reports* 7 (2017).

[14] Pavan Kapanipathi, Prateek Jain, Chitra Venkataramani, and Amit Sheth. 2014. User interests identification on twitter using a hierarchical knowledge base. In *European Semantic Web Conference*. Springer, 99–113.

[15] Sarasi Lalithsena, Pavan Kapanipathi, and Amit Sheth. 2016. Harnessing relationships for domain-specific subgraph extraction: A recommendation use case. In *Big Data (Big Data), 2016 IEEE International Conference on*. IEEE, 706–715.

[16] Emmanuel Lazega. 2001. *The collegial phenomenon: The social mechanisms of cooperation among peers in a corporate law partnership*. Oxford University Press on Demand.

[17] Jure Leskovec. 2012. Stanford network analysis package (snap). *URL http://snap. stanford. edu* (2012).

[18] Jure Leskovec and Julian J Mcauley. 2012. Learning to discover social circles in ego networks. In *Advances in neural information processing systems*. 539–547.

[19] Chunshan Li, William K Cheung, Yunming Ye, Xiaofeng Zhang, Dianhui Chu, and Xin Li. 2015. The author-topic-community model for author interest profiling and community discovery. *Knowledge and Information Systems* 44, 2 (2015), 359–383.

[20] Liyuan Liu, Linli Xu, Zhen Wangy, and Enhong Chen. 2015. Community detection based on structure and content: A content propagation perspective. In *Data Mining (ICDM), 2015 IEEE International Conference on*. IEEE, 271–280.

[21] Jan Lorenz, Heiko Rauhut, Frank Schweitzer, and Dirk Helbing. 2011. How social influence can undermine the wisdom of crowd effect. *Proceedings of the National Academy of Sciences* 108, 22 (2011), 9020–9025.

[22] Pablo N Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. 2011. DBpedia spotlight: shedding light on the web of documents. In *Proceedings of the 7th international conference on semantic systems*. ACM, 1–8.

[23] Mark EJ Newman. 2006. Modularity and community structure in networks. *Proceedings of the national academy of sciences* 103, 23 (2006), 8577–8582.

[24] Mark EJ Newman and Aaron Clauset. 2016. Structure and inference in annotated networks. *Nature communications* 7 (2016).

[25] Guillermo Palma, Maria-Esther Vidal, and Louiqa Raschid. 2014. Drug-target interaction prediction using semantic similarity and edge partitioning. In *International Semantic Web Conference*. Springer, 131–146.

[26] Simon Pool, Francesco Bonchi, and Matthijs van Leeuwen. 2014. Description-driven community detection. *ACM Transactions on Intelligent Systems and Technology (TIST)* 5, 2 (2014), 28.

[27] Gang Wang, Bolun Wang, Tianyi Wang, Ana Nika, Haitao Zheng, and Ben Y Zhao. 2014. Whispers in the dark: analysis of an anonymous social network. In *Proceedings of the 2014 Conference on Internet Measurement Conference*. ACM, 137–150.

[28] Xiao Wang, Di Jin, Xiaochun Cao, Liang Yang, and Weixiong Zhang. 2016. Semantic Community Identification in Large Attribute Networks. (2016). https://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/11964

[29] Bowei Yan and Purnamrita Sarkar. 2016. Convex Relaxation for Community Detection with Covariates. *arXiv preprint arXiv:1607.02675* (2016).

[30] Jaewon Yang, Julian McAuley, and Jure Leskovec. 2013. Community detection in networks with node attributes. In *Data Mining (ICDM), 2013 IEEE 13th international conference on*. IEEE, 1151–1156.

[31] Yuan Zhang, Elizaveta Levina, Ji Zhu, et al. 2016. Community detection in networks with node features. *Electronic Journal of Statistics* 10, 2 (2016), 3153–3178.

[32] Xiaohan Zhao, Alessandra Sala, Christo Wilson, Xiao Wang, Sabrina Gaito, Haitao Zheng, and Ben Y Zhao. 2012. Multi-scale dynamics in a massive online social network. In *Proceedings of the 2012 Internet Measurement Conference*. ACM, 171–184.