Knowledge Graph Enhanced Community Detection and Characterization

Shreyansh Bhatt
Kno.e.sis, Wright State University

Swati Padhee
Kno.e.sis, Wright State University

Amit Sheth
Kno.e.sis, Wright State University

Keke Chen
Kno.e.sis, Wright State University

Valerie Shalin
Kno.e.sis, Wright State University

Derek Doran
Kno.e.sis, Wright State University

Brandon Minnery
Wright State Research Institute

ABSTRACT

Recent studies show that by combining network topology and node attributes, we can better understand community structures in complex networks. However, existing algorithms do not explore "contextually" similar node attribute values, and therefore may miss communities defined with abstract concepts. We propose a community detection and characterization algorithm that incorporates the contextual information of node attributes described by multiple domain-specific hierarchical concept graphs. The core problem is to find the context that can best summarize the nodes in communities, while also discovering communities aligned with the context summarizing communities. We formulate the two intertwined problems, optimal community-context computation, and community discovery, with a coordinate-ascent based algorithm that iteratively updates the nodes' community label assignment with a community-context and computes the best context summarizing nodes of each community. Our unique contributions include (1) a composite metric on Informativeness and Purity criteria in searching for the best context summarizing nodes of a community; (2) a node similarity measure that incorporates the context-level similarity on multiple node attributes; and (3) an integrated algorithm that drives community structure discovery by appropriately weighing edges. Experimental results on public datasets show nearly 20 percent improvement on F-measure and Jaccard for discovering underlying community structure over the current state-of-the-art of community detection methods. Community structure characterization was also accurate to find appropriate community types for four datasets. Moreover, our algorithm yields insightful community structures that explain the contextual relationships among communities, which helps us better understand two real-world applications of social networks.

ACM Reference Format:
Shreyansh Bhatt, Swati Padhee, Amit Sheth, Keke Chen, Valerie Shalin, Derek Doran, and Brandon Minnery. 2019. Knowledge Graph Enhanced Community Detection and Characterization. In *The Twelfth ACM International Conference on Web Search and Data Mining (WSDM '19), February 11–15, 2019, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3289600.3291031

1 INTRODUCTION

"Does interest in sports or music form conversational communities among participants?" Recent approaches model such problems as community detection and characterization. They report both state-of-the-art community detection accuracy and effective community characterization with node attributes driving community detection[23][33]. These approaches increase edge weights between nodes belonging to the same community if these nodes share similar node attribute values. While such techniques detect whether communities form around the *particular* sports teams or music bands explicitly referenced, they fall short on identifying whether communities are formed from participants' *general* interest in sports or music. Such problems require meaning-oriented community characterization with an assessment of accuracy that combines network nodes, edges, and node attributes. Instead of relying on apparent attribute relations, i.e., exact matching for nominal attributes and Euclidean distance for numeric attributes, we seek contextual relations between attribute values. The resulting meaningful community detection is also crucial for applications such as network visualization [21] and online-marketing[29].

Consider the friendship network of participants shown in Figure 1 with the available node attributes expressed as the city in which a participant lives. The existing approach to community detection on such a network considers"Austin", "Dallas", and "Houston" as different attribute values [23][33], missing the important subsuming relationship (i.e., they are in the same state). Considering such relationships can improve community characterization. Moreover, detecting such relationships provides a basis for updating edge weights.

We explore the use of domain-specific knowledge graphs to find such contextually meaningful attribute relationships. Domain-specific *hierarchical* knowledge graphs (HKGs) provide particularly relevant real-world clustering information. The domain-specific HKG in Figure 2 indicates that all states of United States are subsumed by "States in United States". The decomposition starting from each concept of such an HKG provides a *context.* E.g., all the concepts subsumed by "Cities in Ohio" along with "Cities in Ohio" provides a *context* "Ohio". Such knowledge graphs can be

WSDM '19, February 11–15, 2019, Melbourne, VIC, Australia
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-5940-5/19/02…$15.00
https://doi.org/10.1145/3289600.3291031

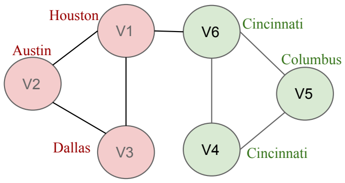


Figure 1: Friendship network with nodes representing user, edges representing friendship, and node attribute as the home-city.

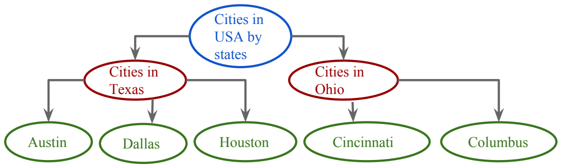


Figure 2: Hierarchical knowledge graph for USA geo-location.

generated automatically with demonstrated benefit to applications such as personalization [15]. HKGs provide complementary real-world information regarding communities or clusters that may not be explicit in the network but are nevertheless useful in finding and characterizing communities. However, incorporating domain-specific HKGs in community detection raises three key challenges. 1. There is no clear similarity measure for computing node similarity using an HKG characterization. For example, at the city level, "Austin", "Dallas", and "Houston" are different, while they are same in the context of "Cities in Texas". Additionally, we need to determine the optimal context characterizing the community structures. E.g., in Figure 1, "Cities in Texas" characterizes Community 1(V1, V2, and V3). 2. Context optimality reflects multiple factors. Moving up the hierarchy towards the root, we obtain a more generalized context subsuming more lower level attribute values. However, the generalization disguises the differences between attribute values, potentially losing details that distinguish node groups. 3. Optimizing context generalization should coordinate with the discovery of the topological structure, and the topological structure discovery should reflect computed community contexts.

We develop an algorithm which iteratively optimizes two tasks: (i) Optimal community label assignment while keeping the community context unchanged, (ii) Optimal community context assignment while keeping the community labels constant. For the first task, we propose a contextual similarity measure for defining node pair similarities to capture community contexts. We employ a widely used community label assignment algorithm, the Louvain community detection algorithm [3], which finds community labels for nodes using modularity maximization. For the second task, we find a concept generalization scheme that balances between two criteria: 1. *Informativeness*, which is essentially the specificity of a concept in a hierarchical knowledge graph. The lower the concept is in the hierarchy, the more specific information the generalization preserves and 2. *Purity* which is the difference between the number of nodes subsumed by a concept of a given community and neighboring communities.

Our framework has three unique features: (i) It can accept any predefined domain-specific hierarchies for any attributes (numeric or nominal), together with a topological network structure (i.e., nodes and edges). (ii) The algorithm does not assume a priori that a domain must correlate with the communities we want to discover. Instead, it will quantify the relationship between a certain domain and communities. If one exists, the algorithm will progressively find it. (iii) It allows us to analyze competing contexts on the same attributes. For example, the location attributes may have multiple different context hierarchies: one based on the geographical concepts, another on housing markets, and the third on household income levels.

As the resulting algorithm can assign more appropriate edge weights than using only attribute values, it can facilitate the discovery of an accurate community structure. We evaluated community detection accuracy on four real-world networks and five baseline community detection algorithms. The proposed algorithm improves community detection accuracy by nearly 20%. We also evaluated the accuracy of community structure characterization and found that the proposed approach was able to discover correct underlying community "types" for all four datasets while two baseline methods [23][33] failed to characterize communities for at least two datasets. We also demonstrate that contextual community detection and characterization effectively mediates the representation of the original data for two practical problems: Harassment in online social networks and diversity in crowd sampling.

We summarize the specific contributions of this paper as:

- This paper presents a possibility of complementing network data with domain-specific knowledge graph to enhance community detection.
- A contextual similarity measure and optimal community context computation approach to find more meaningful community descriptions and improve community detection accuracy.
- Detailed evaluation demonstrating enhanced community detection accuracy and meaningful community structure characterization.

This paper is organized as follows. Section 2 provides a problem formulation. Section 3 provides detail on our community detection algorithm, Section 4 evaluates the algorithm by comparing it with state-of-the-art community detection algorithms. Section 5 provides two case studies and demonstrate use of the proposed approach. Section 6 compares our work with other community detection research. Finally Section 7 provides conclusions and future work.

2 DEFINITIONS AND NOTATIONS

Our algorithm takes a graph $G = (V, E, A)$ with nodes nodes(V), edges (E), node attributes (A), and one or more domain-specific HKGs $H = \{H_1, \ldots, H_t\}$ as input. It finds community label for each node and contexts from domain-specific HKGs summarizing each community of nodes. In the following, we define definitions and notations that we are going to use in the algorithm description.

Each $H_i \in H$ is a set of concepts and relationships, $H_i = \{c_1, c_2, \ldots, c_m\}$. We generated H_i from DBpedia[1] starting with a "root node" concept and recursively extracting all the concepts connected with "skos:broader" or "subject" relationships. One can use a domain-graph generation tool such as [15] to generate such a hierarchical graph. Note that each node $v_i \in V$ is a list of concepts indicating attribute values $v_i = \{c_1, c_2, \ldots, c_p\}$ such that each $c_i \in v_i$ is part of at least one $H_i \in H$. As an example, nodes $V1 - V6$ in Figure 1 are nodes and each v_i has one attribute c_i (city) that is

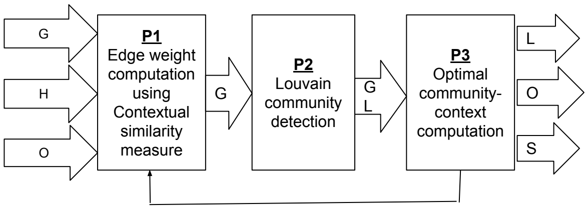

G
H
O
P1
Edge weight computation using Contextual similarity measure
G
P2
Louvain community detection
G
L
P3
Optimal community-context computation
L
O
S

Figure 3: Overview of the proposed approach. P1 computes contextual similarity between nodes and edge weights, inputs an updated graph to P2 which computes community labels (L). P3 computes community context O and concept weight vector S.

a part of the example hierarchical knowledge graph as shown in Figure 2. In this example, we are using only one domain-specific HKG but multiple domain-specific HKGs can also be the inputs.

Given G and H, the algorithm finds a community label set $L = \{l_1, l_2, \ldots, l_k\}$ such that L_i indicates community label for i^{th} vertex from a set of community labels $K = \{1, 2, \ldots, k\}$. It also computes an optimal context representing each community $i \in K$, i.e., $O = \{o_1, o_2, \ldots, o_k\}$. Here, each $o_i \in O$ is a t dimensional vector of concepts with each $O_{ij} = c_j$ s.t. $c_j \in H_j$. *O is initialized with "root concept" of each H_j.* Along with O, the algorithm also computes context scores $S = \{s_1, s_2, \ldots, s_k\}$ for each community indicating appropriateness of each context in each community. Each $s_i \in S$ is a t dimensional vector of real numbers.

3 APPROACH

The proposed algorithm to generate community labels(communities) iteratively optimizes 1) community label assignment, keeping the community context constant and 2) community context assignment, keeping the community labels constant. We then recompute edge weights with the updated community context (O). Figure 3 summarizes this approach. Next, we describe the proposed contextual similarity measure (P1), community-context computation (P3), and the proposed way of integrating new node similarity values to find final community labels L and descriptions(O).

3.1 Contextual Similarity Measure

Here, we describe the proposed similarity measure, $\phi(v_1, v_2, h_i, o_{ij})$, to compute a similarity score between nodes v_1 and v_2 in h_j with o_{ij}^{th} context. Here, i is the community to which the edge $v_1 - v_2$ belongs. Similarity is computed in the j^{th} domain-specific HKG. Note that similarity is computed in the o_{ij} context, i.e., a hierarchy starting from o_{ij}. We extend the semantic similarity measure to compute similarity between two lists of concepts represented in a HKG.

In such a taxonomy with a given root node, the similarity between two concepts can be computed using a semantic similarity measure [26]. This measure finds the least common ancestor subsuming these concepts in the hierarchy. Similarity is the "informativeness" of that least common ancestor. More generic concepts provide less information. For example, in Figure 1, "USA" has less informativeness than "Ohio". Hence, the semantic similarity between "Cincinnati" and "Columbus" subsumed by Ohio is higher than "Columbus" and "Dallas" subsumed by USA. Informativeness, in its simplest form, is identified as $1 - \frac{\eta_i}{\eta_{root}}$ where η_i is number of concepts *subsumed* by i. Sanchez et al. proposed that inner HKG concepts should be evaluated separately from the leaves and revised informativeness formula as follows [26],

$$I_c = \left(2.0 - \frac{\sum_{l<c} \frac{1}{S_l}}{\sum_{l<root} \frac{1}{S_l}}\right) \tag{1}$$

Here, S_l refers to the number of concepts that subsumes l. The informativeness I of a concept c is summation of the subsumers over all leaves l such that $l < s$. We subtract the value from 2.0 as we want the values in (1.0, 2.0). In figure 2, $S_{Cincinnati} = 2$ and $S_{Columbus} = 2$ as they are subsumed by two concepts, "Cities in Ohio" and "Cities in USA". Hence, $I_{CitiesinOhio} = 2 - \frac{\frac{1}{2}+\frac{1}{2}}{\frac{1}{2}+\frac{1}{2}+\frac{1}{2}+\frac{1}{2}}$. The denominator has four terms corresponding to each one of the four leaves subsumed by "Cities in USA"(root).

As we have the nodes represented as a list of concepts, the existing similarity measure must find the least common ancestor of each pair of concepts from v_1 and v_2 and consider their informativeness score to compute semantic similarity. Instead, we compute the similarity between two lists. We extend each vertex list, v_1 and v_2, by recursively computing the subsuming "parents" of each concept $c \in v_i$ until o_{ij}. Along with each concept, we also compute its informativeness score. Consider an extended vertex list with concepts and informativeness score as v_{ext1} and v_{ext2}. The similarity is computed as the weighted Jacquard similarity [13] between v_{ext1} and v_{ext2}.

$$J(v_{ext1}, v_{ext2}) = \frac{\sum_l min(v_{ext1}^l, v_{ext2}^l)}{\sum_l max(v_{ext1}^l, v_{ext2}^l)} \tag{2}$$

Here, l represents vector dimensions. In our case, each one of these dimensions is a concept c and the value is its informativeness score. We chose weighted Jaccard similarity as it satisfies the following requirements. 1. v_1 and v_2 get a low similarity value if they have fewer concepts in common. 2. v_1 and v_2 get low similarity value if the concepts are repeated a different number of times. If the concept c appears three times in v_{ext1} and four times in v_{ext2} then the numerator's value for that concept will be less than the denominator leading to reduced similarity. 3. v_1 and v_2 get a low similarity value if the concepts in common have less informativeness.

This similarity computation depends on o_{ij}, i.e., a concept of h_j^{th} knowledge graph representing community i. As an example, the similarity between $v_1 = \{Cincinnati\}$ and $v_2 = \{Columbus\}$ results in $v_{ext1} = \{(Cincinnati, 1.8), Ohio(1.6), USA(1.0), Columbus(0.0)\}$ and $v_{ext2} = \{Columbus(1.8), Ohio(1.6), USA(1.0), Cincinnati(0.0)\}$. The bracketed value is the informativeness score for each concept according to HKG in Figure 2. The weighted jaccard between v_{ext1} and v_{ext2} results in similarity 0.419.

We used the Louvain algorithm to find community labels L for each node in the weighted graph. Next, we describe the process of finding an appropriate concept describing each community.

3.2 Optimal community context computation

In this subsection, we describe how we compute o_{ij}, an optimal context of $h_j \in H$ describing community $i \in C$. As described, context o_{ij} is essentially a hierarchy starting at the concept o_{ij} in

h_j. Hence, o_{ij} is represented by the concept $c \in h_j$ that is the most relevant concept for the community c. Such a concept is found based on two criteria, 1. appropriate generality (referred as purity) of a concept and 2. informativeness. Next, we describe the detailed procedure.

h_j hierarchies provide *real-world clustering knowledge*. As an example, in the context of "Cities in USA", "Austin", "Dallas", and "Houston" forms the cluster "Cities in Texas". In other words, as "Cities in Texas" subsumes three cities, it can represent and even validate these three cities being in one cluster. Each concept of h_j can potentially represent a community i based on node attribute values of nodes belonging to a community i. Our intuition for finding such a concept is as follows. *For any community, a concept can represent that community if it happens to subsume more concepts in a community than if the concepts of the community were distributed at random in a HKG.* As described in Section 3.1, use of $2 - informativeness$ can serve as a better approximation for "concepts distributed at random" than $\frac{\eta_i}{\eta_{root}}$. Hence, maximizing the following with respect to concept of a knowledge graph can indicate the optimal context representing a community,

$$max_c \left(\eta_c - \eta_c \times \frac{\sum_{l<c} \frac{1}{S_c}}{\sum_{c<root} \frac{1}{S_l}} \right) \tag{3}$$

Here, η_i is the number of concepts (belonging to a community i) subsumed by c. We also minimize the number of concepts subsumed from neighboring communities. Considering this and rearranging terms, the final maximization term is:

$$o_{ij} = max_c \left((\eta_{n\in i} - \eta_{n\in \bar{i}}) \times I_c \right) \tag{4}$$

where $\eta_{n\in i}$ indicates the number of concepts in i subsumed by c and $\eta_{n\in \bar{i}}$ indicates the number of concepts in the neighboring communities of i subsumed by c. The first term corresponds to "purity" while the second term corresponds to the informativeness of c. In addition to the concept c maximizing the score, we also retain the actual score as s_{ij} which indicates the relative context importance of context j in community i.

For the attribute list $T = \{c_1, c_2, \ldots, c_f\}$ and $\bar{T} = \{\bar{c}_1, \bar{c}_2, \ldots, \bar{c}_f\}$ indicating the concepts of community i and neighboring communities in h_j respectively, Algorithm 1 finds the concept maximizing Equation 4. We pre-compute the hierarchical level, e.g., the root is set to '0' and all the leaves are at level "tree height " and the informativeness of each $c \in h_j$. We create a list with concepts at the lowest level and a score associated with each concept indicating the difference between the number of concepts each subsumes from T and number of concepts it subsumes from $\bar{T}$. Then, we compute a score for each concept and update the concept with the maximum score thus far and the maximum score. Next hop "parents", i.e., concepts subsuming the current concept, are included in the list to investigate. The scores associated with the parent concepts are also attached as it indicates the number of concepts subsumed from T and $\bar{T}$. As a vertex may be represented with concepts other than leaves, there may be some concepts left in T and $\bar{T}$ that belong to higher level. They are added using *add_list* whenever the level that they belong to is processed. Because the root has no parents, the temp_list will eventually become empty. To avoid loops, we also condition on $level \geq 0$.

```
Input: $T$, $\bar{T}$, and $h_j$
Output: $c_{opt}$, $s_{max}$
$c_{opt} = root$, $s_{max} = root\_score$
Associate score with each concept. -1 for $\bar{T}$ and 1 for $T$
level = lowest_level()
list = add_list($T$, $\bar{T}$, level)
while list not empty and level ≥ 0 do
    temp_list = empty
    for $c \in list$ do
        $s_{cur}$ = score(c) ×$I_c$
        update_optimal($c_{opt}$, $s_{max}$, c, $s_{cur}$)
        for $p \in parents(c)$ do
            add_parent(temp_list, p, score(c))
        end
    end
    level = level - 1
    add_list($T$, $\bar{T}$, level)
    list = temp_list
end
```

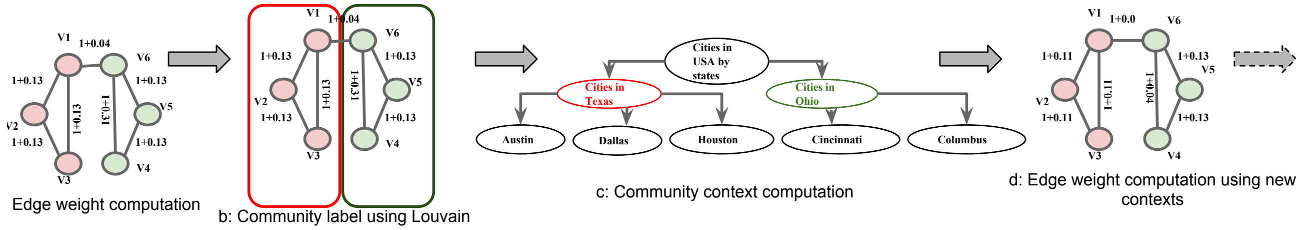
Algorithm 1: Optimal community-context computation

One of the most important steps in the algorithm is add_parent. The concept maximizing the criteria must subsume at least one of the concept of i. Thus, we explore for the solution among hierarchical "parents" of any $c \in T$. We avoid adding a parent (stop looking for a solution in the path) if its informativeness score decreases so much so that even if it were to subsume rest of the remaining concepts, it could not get a higher score than *max_score*.

3.3 Unified framework

Algorithm 2 describes the final algorithm and Figure 4 demonstrates the algorithm on the example network shown in Figure 1. We start with computing node pair similarities between all nodes for which $E_{ij} \neq 0$. We consider each edge ij as an edge from i's community to j's community. Hence, edge weight E_{ij} is computed with contexts for both communities L_i and L_j. Next, it computes community labels L by maximizing a modularity equation with respect to L. Note that $f(ij, L)$ is a function that determines whether i and j are in the same community based on their community labels. Specifically, $ij \in l$ iff $L_i = L_j = l$.

Modularity is an evaluative measure of community structure. Accordingly, *a part of graph (a group of nodes) is interesting if the number of edges within that group is higher than if the nodes were to assign into groups at random*, formally: $\sum_{i=1}^{k} \left(e_i - a_i^2 \right)$. Here, e_i is the number of edges in a community i, and a_i is the expected number of edges in community i. Note that we used a similar idea in designing our optimal community context computation. $\frac{k_i \times k_j}{4m^2}$ provides better estimation of a_i^2 as the probability of an edge belonging within a community depends on the degree of nodes connected that edge [22]. Modularity maximization is one of the most widely used community detection technique. We used Louvain algorithm based modularity maximization as it has identified qualitatively robust community structure[3]. It is a greedy algorithm that processes each vertex at random and assigns a community label based on the one that can result in the maximum modularity gain. The reader is encouraged to follow [3] for details.



Edge weight computation

b: Community label using Louvain

c: Community context computation

d: Edge weight computation using new contexts

Figure 4: Demonstration on an example network. (a) Normalized edge weights are first computed using $\omega = 1.0$ and contextual similarity kernel with root node as the context identifying each community. (b) Community labels are computed using Louvain. (c) Optimal contexts "cities in texas' and "cities in ohio" computed for c1, and c2 respectively, (d) Normalized edge weights recomputed using new contexts. Note the modularity increase from (a) to (d).

Using the new community labels, we compute the optimal context representing each community $i \in K$. The process is repeated until maximum modularity is achieved or a max number of iterations. d_i in modularity $Q(E, L)$ is the degree of a node i, computed as the summation of all edges incident on i and m is summation of all the edge weights. ω is a hyper-parameter indicating the relative importance of edge to the node pair similarity computed using the contextual similarity. We iteratively update community label assignment and community-context vector o_i for each community i. Such an algorithm is likely to be stuck in local maxima. Thus, we repeated the process 10 times for each dataset, randomly selecting the vertex order to be processed by the Louvain algorithm. We consider the result for which we achieved the maximum modularity value.

```
Input: G=(V, E, A), $H = \{h_1, h_2, \ldots, h_t\}, max\_iters, threshold$
Output: $L, O, S$
while $mod < threshold$ or until $max\_iters$ do
    $w_{ij}(\omega, i, j, H, O, l) = \omega + \sum_{q=1}^{t} \phi(i, j, h_q, o_q, l)$
    $E_{ij} = w_{ij}(\omega, i, j, H, O, L_i) + w_{ij}(\omega, i, j, H, O, L_j)$
    $Q(E, L) = \frac{1}{2m} \sum_{l \epsilon K} \sum_{f(ij,L)} E_{ij} - \frac{d_i d_j}{4m^2}$
    $L = max_L Q(E, L)$
    for $i \in K$ do
        for $h_j \in H$ do
            $o_{ij} = max_c \left((\eta_{n \in i} - \eta_{n \in \bar{i}}) \times I_c\right)$
        end
    end
end
```

Algorithm 2: Community detection and characterization algorithm

3.4 Algorithm Complexity and Convergence

Each iteration consists of the three steps (P1, P2, and P3) described in Figure 3. P1 and P2 process each edge resulting in $O(n)$ time complexity where n indicates the number of edges. P3 maps nodes of each community to a knowledge graph and computes an optimal context for each community resulting in the time complexity of $O(ck)$ for c communities and knowledge graph of k concepts. Hence, The time complexity of the algorithm is $O(n + kc)$ since the number of iterations $i << n$.

The Louvain algorithm (P2) optimizes Modularity to find a community structure. The algorithm could diverge if the optimal community context results in edge weights that could decrease Modularity. A generic community context will result in a relatively less similarity value and indicates that the communities should be computed only using the ω and won't affect the Modularity value. A specific community context will change the edge weights to make the current community structure stronger. Hence, it is likely to increase the modularity value. Either way, the modularity value is not expected to decrease due to P3. We also found the algorithm converges to a satisfactory modularity value for all of the datasets used in experiments. We will provide detailed theoretical proof and scalable implementation of the proposed algorithm in our next paper.

4 EVALUATION

The datasets, measures, comparison baselines and results follow. We refer to the proposed approach as "KDComm".

4.1 Datasets

We used four datasets to assess community detection accuracy and community structure characterization.

4.1.1 G+ ego network. It is G+ user dataset with friends of a given user represented as nodes and friendship relationship represented as an edge[16]. Circles (communities) result from densely-connected sets of friends [22]. Each node has four features: job title, current place, university, and workplace. A user-pair(edge) is compared using knowledge graphs based on, *Category:Occupations*, *Category:Companies_by_country_and_industry*, *Category:Countries*, *Category:Universities_and_colleges_by_country.*

4.1.2 Twitter. The Twitter dataset consisted of tweets about the configuration of a team for the Fantasy Premier League (FPL). We created a re-tweet network between these users based on information about their tweets. The re-tweet network between these users represents agreement. We used DBpedia spotlight [20] to identify soccer player mentions in these tweets. The final network consisted of users as nodes, re-tweet as edges, and FPL players mentioned by a user as node attributes.

These users have different types of teams where they select players of one position more than the others. These types include 1. Forwards, 2. Defenders, 3. Mid-fielders. As they discuss their players in their FPL related tweets, a dense re-tweet network between these users with community type characterization indicates a group of users interested in similar types of teams. Hence, given a network of these users, the task divides users into three circles âĂŤ users with more "Forward" players in their team, more “Defender” players in their team, and more “Mid-fielder” players in their team. For KDComm, we generated three HKGs with following root nodes, Category:Association_football_defenders, Category:Association_football_forwards, and Category:Association_football_midfielders.

We created ground truth circles using these users’ actual team configurations available on the FPL website[2]. Users with more than the usual [1] number of players for any position is included in that circle [2].

4.1.3 DBLP. The DBLP dataset [14] is a co-author network, where each author is characterized by a set of keywords. Ground truth labels for authors are available for four categories: 1. Machine learning, 2. Data mining, 3. Databases, and 4. Information retrieval. We use a knowledge graph generated with root nodes *Category:Data_Mining*, *Category:Machine_Learning*, *Category:Databases*, and *Category:Category:Information_retrieval.*

4.1.4 Reddit. Each node in this dataset is a user, and an edge indicates users are commenting/replying to the same post, and a node attribute is a set of comments made by that user. Each post has a “sub-reddit” that indicates the type of a post. The communities in this network can be evaluated using each user’s subreddits. Users belonging to the same community are likely to discuss the same subreddits [9] We considered the first four days of April 2015 to create this network[3]. We considered subreddits related to Economics and the NFL as they were the most discussed subreddits in the dataset. The domain-specific HKGs were extracted for *Category:Economics* and *Category:National_Football_League* as root nodes.

4.2 Evaluation Measures

To evaluate community detection accuracy in G+, DBLP, and Twitter datasets, we used Yang et al. ’s community F-Measure and a Jaccard measure [31]. The evaluation function is,

$$\frac{1}{2\left|C^*\right|}\sum_{C_i^*\in C^*}\overset{max}{C_j}\in C\delta\left(C_i^*,C_j\right)+\frac{1}{2\left|C\right|}\sum_{C_j\in C}\overset{max}{C_i^*}\in C^*\delta\left(C_i^*,C_j\right) \quad (5)$$

Here, $\delta(C_i^*, C_j)$ is a similarity measure, either Jaccard or F-score similarity (F-Measure). C is the community label set found by the algorithm and C^* is the ground truth community label set. For community detection evaluation in Reddit dataset, we used Hartman et al.’s rank entropy measure for a given community $R_e = \frac{-\sum_{j=1}^{L}\frac{n_{cj}}{n_c}log_2\frac{n_{cj}}{n_c}}{log_2 n_c}$. Here, j is a subreddit in a community c. n_{cj} is the number of times users of community i commenting on subreddit j. n_c is total comments. A community c is likely to have a lower entropy value if the users of community c are commenting on a few subreddits most of the time.

4.3 Results and Analysis

To evaluate KDComm, we use Liu et al.’s CPCD approach, which is superior to eight other community detection[18]. We also consider JCDC [33] which outperforms five other community detection approaches. Like CPCD, JCDC concerns edge weights based on We used UNCut [32], which outperforms three other graph clustering approaches. We used Newman’s community detection approach (referred to as SI) [23] that also uses attribute values in community structure detection and characterization. Finally, we also compared results with the Louvain algorithm, using only edge information. Evaluation results appear below for: 1. the similarity kernel. 2. community detection accuracy, 3. community structure characterization.

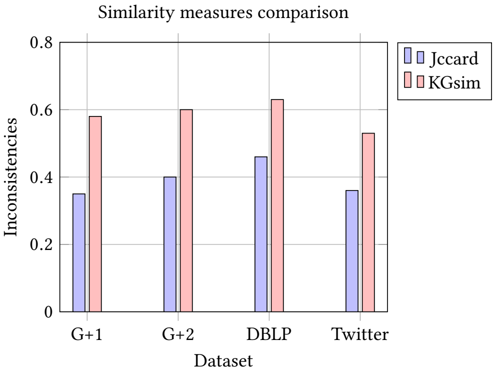


Figure 5: Similarity measures comparison. KGsim was able to assign appropriate edge weights to node pairs, resulting in lower inconsistencies corresponding to the community labels.

4.3.1 Contextual similarity measure evaluation. First we compare the proposed contextual similarity measure (referred as KGsim) with attribute value based similarity. Two sets of user pairs ($n = 1000$) are created from four datasets with ground truth community labels. IntraCommunitySet = $\{s_1, s_2, \ldots, s_n\}$ where each $s_i = \{(u_1, u_2)|u_1 and u_2 \in same community\}$. InterCommunitySet = $\{d_1, d_2, \ldots, d_n\}$ where each $d_i = \{(u_1, u_2)|u_1 and u_2 \in different communities\}$. We compared each $s_i \in$ IntraCommunitySet to all the $d_i \in$ InterCommunitySet resulting with n^2 comparisons. Ideally, each $s_i \in$ IntraCommunitySet should be higher than all the $d_i \in$ InterCommunitySet. The number of times $s_i \in$ IntraCommunitySet is lower than $d_i \in$ InterCommunitySet is computed as number of “inconsistencies”. We computed similarity using the proposed similarity measure and Jaccard similarity as Jaccard computes similarity using attribute values. Figure 5 plots the “inconsistencies” to the total comparison (n^2) ratio.

For the G+1 and G+2 datasets, we used the four features associated with each node as attribute values. For Twitter and DBLP, we used player names and author keywords respectively as attribute values. The proposed similarity measure (KGsim) had lower “inconsistencies” than Jaccard for all the four datasets. Hence, KGsim can best assist edge re-weighting. We did not compute an appropriate context relevant to each community and used “root” node as the context for each dataset.

4.3.2 Community detection accuracy. We compared community detection accuracy to other approaches. CPCD, SI and, UNcut used nominal node attribute values in the form of a 1/0 vector. We focused on the 100 most frequently used words of Reddit forums as attribute value vectors. For JCDC, we used the Jaccard similarity measure to compute similarities. Table 1 shows the results for four datasets. The F-Measure and Jaccard scores reported for G+ are averaged over all the 20 ego networks. The proposed approach achieved better

[1] http://www.soccer-training-guide.com/soccer-formations.html#.Wmk6GZM-eAI
[2] Please contact the corresponding author for the dataset.
[3] https://archive.org/details/2015_reddit_comments_corpus

Algorithm	DBLP		G+		Twitter		Reddit
	F	Jcc	F	Jcc	F	Jcc	R_e
Louvain	0.45	0.40	0.53	0.45	0.30	0.25	0.78
UNCut	0.57	0.51	0.5	0.42	0.35	0.30	0.75
CPCD	0.58	0.49	0.56	0.46	0.34	0.29	0.68
JCDC	0.54	0.5	0.58	0.48	0.33	0.28	0.62
SI	0.56	0.48	0.6	0.53	0.38	0.31	0.63
KDComm	0.66	0.59	0.71	0.60	0.47	0.39	0.48

Table 1: Community detection accuracy results. KDComm achieved the best F-score and Jaccard score for all three datasets.

Dataset	JCDC		SI		KDComm	
	M_{11}	M_{22}	M_{11}	M_{22}	M_{11}	M_{22}
Twitter	0.168	0.154	0.41	0.285	0.6	0.7
G+1	0.56	0.381	0.36	0.263	0.7	0.8
G+2	0.482	0.58	0.7	0.536	0.6	0.75
DBLP	0.32	0.232	0.56	0.377	0.56	0.64

Table 2: Users within community characterization. M is a relevancy score matrix. KDComm found appropriate topics characterizing users within a community for all four datasets while JCDC found appropriate topics for two datasets.

average scores for both measures (F-score and Jaccard) than all other approaches. For comparison on the G+ ego network dataset, we also performed a t-test between the set of F-scores received by KDComm and set of F-scores received by other approaches. A $p - value < 0.05$ also indicated superior performance of KDComm over all other baseline methods. Similarly, A $p - value < 0.05$ for Jaccard measure comparison confirms superior performance.

KDComm achieved the best F-score and Jaccard for the Twitter dataset, dividing users into three communities. As the Louvain algorithm found more than three communities, we merged communities based on community-context scores, merging users divided into two different "Defender" communities. KDComm also outperformed the other methods for the DBLP dataset as well, requiring similar community merging.

For the Reddit dataset, UNCut, CPCD, JCDC, and SI require a pre-determined number communities. We set the number of communities using KDComm. We report rank entropy averaged over all the communities. Lower entropy indicates a better community structure according to this measure [9]. KDComm achieved the lowest entropy among all methods.

4.3.3 Characterization of community structure. Next, we evaluated whether KDComm characterized users belonging to different communities with an appropriate community type. For each dataset, we considered users from two communities and evaluated whether KDComm, SI, and JCDC can find underlying two communities and compute an appropriate type of community-based on node attributes. We considered attributes such that attribute type can identify community type. All the three methods(KDComm, SI, and JCDC) compute a “relevancy score” of each attribute type to each community, E.g., S for KDcomm. These “relevancy scores” for two attribute types and two communities can be represented as a 2x2 matrix, M. Each cell of this matrix indicates the relevancy score of attribute type to a community.

The relatively larger score for an attribute type indicates greater importance for that attribute type. All four datasets had community type and labels for nodes. 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 to as G+1 and G+2) for which we distinguished two ground truth communities based on “University” and “Workplace” attributes/contexts.

As the inputs were provided with two contexts/attribute types, a correct attribute type assignment is reflected by a higher score assigned to that attribute type relative to the other attribute type. As we used a normalized attribute/context score for each method, a score > 0.5 indicates a particular attribute type as the community type. We had attribute type 1, “forwards”, “University”, “Data Mining” a more relevant to Twitter, G+, and DBLP datasets’ community one according to ground truth. Hence, we expect a context1 (T1) score higher than 0.5 for community 1 and a context2 (T2) score higher than 0.5 for community 2. Hence, we expect relevancy score matrix M_{11} and M_{22} to be higher than 0.5. KDComm found the expected community-context scores for all the four datasets (see Table 2). Both JCDC and SI failed to find the expected community-context scores for at least two datasets.

5 NETWORK EXPLORATION

An application illustrates the quality of community identification. In this section, we discuss two real-world datasets that we explored using the proposed method.

5.1 Twitter and Wisdom of Crowd

A “wisdom of the crowd” application demonstrates superior community identification. Accordingly, a group of *independent* and *diverse* individuals can makes a superior collective decision [27].

Fantasy soccer captain prediction is such a task where we can witness the “wisdom of crowd”[8][2]. As a diverse set of individuals in captain prediction task bring diverse perspectives, their aggregated judgment is likely to be more accurate than randomly sampled participants. However, the characterization of diversity in such a task is still an open research area. As KDComm can effectively find densely connected users (indicating potential mutual influence) along with their contextual characterization, it can support the formation of a diverse crowd. One member from each community forms a new crowd of potentially diverse and independent users. Here, we used soccer positions as a context for community detection, i.e., HKG of Defenders, Forward, and Mid-fielders as described in Section 4.1.2. Additionally, we also used three English Premier League teams (Manchester United, Liverpool, and Arsenal) as contexts in KDComm, based on their explicit mentions in user tweets, and ran our community detection. From the resulting communities, we formed 100 diverse crowds of size six by randomly picking two users from each type of community. Here, type of community refers to the type (specific soccer position or a team) for which the community had a maximum s_{ij} score. We explored both sets of community semantics: DiversePositions and DiverseTeams.

We used the evaluation dataset as described in [2] for the FPL captain prediction task. We consider a crowd’s captain choice as the captain selected by the greatest number of individuals in a crowd. The actual Fantasy points received by that captain demonstrates how well the two sets of community semantics perform compare to individual users.

Crowd selection	Avg higher than % users
Random	76%
DiversePositions	81%
DiverseTeams	86%

Table 3: Diversity based crowd selection and wisdom of crowd. DiverseTeams set of crowds outperform individual users 81% and 86% of the time depending upon community semantics.

C(size)	Sports(Relevancy)	Music(Relevancy)
C1(47)	U.S. Women's soccer(0.36)	Bob Marley(0.64)
C2(40)	Cleveland Browns(0.45)	Keke Palmer(0.55)
C3(38)	American Football in Boston(0.39)	Machine Gun Kelly(0.61)

Table 4: Top 3 communities identified using Sports and Music contexts. Community description in Sports and Music contexts provided along with the normalized relevancy scores. Music context was found to be more relevant in creating the community structure.

Table 3 shows results for the number of individual users that a crowd outperformed on an average. Specifically, an average captain score of DiversePositions and DiverseTeams was compared with the captain score achieved by individual crowd members. We also created one more set of Random crowds, by selecting 100 crowds of six individuals at random. We found that a random crowd, on an average, performed better than 75% of the individual users. However, the DiversePositions crowd set outperformed 81% of the individual crowd members, and the DiverseTeams outperformed 86% of the individual crowd members.

As the proposed approach can identify network division along contexts, it can help analyze raw network data and inform relatively more sophisticated tasks such as "Crowd Wisdom".

5.2 School student communication network

The network that results from high school students' Twitter conversation network contains topics or contexts that create dense conversation groups. We demonstrate the use of the proposed approach to explore whether certain topics/contexts form a dense conversation community structure and contribute to the identification and characterization of insider-outsider [19], phenomena that contribute to harassment potential. We crawled for 388 high school students' tweets and had each student as a node, a mention or reply as an edge, and relevant domain-words from tweets as node attributes.

We explored two contexts, American Sports and American Music, to find whether they form modular conversational communities. First, we analyzed the conversation network without considering node attributes. The final modularity value of 0.32 does indicate a community structure based on edges alone. However, using domain-specific knowledge graph created with "Category:Sports_in_the United_States", we also generated node (student) attributes as domain relevant concepts characterizing each node and performed the proposed community detection. We discovered community structure with improved modularity of 0.35. Similar processing with "Category:American_music" resulted in community structure with a higher modularity score of 0.38. Next, we used both the contexts in community detection. We found a slightly better modularity score of 0.4. All of the modularity scores improve with node attributes, supporting the claim that the proposed algorithm favored American music (more informative context) and downplayed Sports (less informative context). As described in Table 4, community-context relevancy scores also indicated that Music was more informative in finding the community structure than sports. It also provides the most relevant contexts associated with four of the largest communities. To analyze the divergence from the edge-based community structure, we computed F-measure defined in the evaluation. F-score of 0.38 between edge-based community structure and community structure with both contexts suggested a divergence in assigning community labels to nodes in the presence of the contexts. Hence, contextual analysis has the potential to improve insider-outsider identification and characterization (with contexts identified for communities). Isolated nodes (student) suggest harassment potential [10]. Moreover, by characterizing the context, the approach can also provide the foundation for predicting the harassment potential for a new node not considered in the original community detection.

6 RELATED WORK

Bothorel et al. provides a good summary of community detection methods that incorporate graph attributes [4]. Among the recent approaches, Wang *et al.* works for non-text real-valued node attributed graphs unlike several others [4]. In an approach proposed by Qin *et al.*, link and node attributes are combined at different rates during community detection for improved community detection accuracy. Contrary to the proposed approach, these works do not focus on characterizing community structures. CPCD [18] and UNCut[32] used in the comparison also focus on identifying communities than characterizing these communities.

Several generative models also detect communities and provide information as to the labels that nodes in a community have in common[5][17][12]. Among recent approaches, He *et al.* finds communities by jointly optimizing over node attributes and links using a generative model [11], similar to Wang *et al.*[30]. These approaches characterize a community structure by revealing latent topics within the textual node attributes of a community. They do not work for non-textual node attributes nor do they find communities along given set of topics. The latent community description is less informative compared to the community descriptions identified by proposed approach.

The community detection in node attributed graphs from Zhang *et al.* [33] and Newman *et al.* [23] inspires our own method. Such methods find communities based on edges and then refine these communities, i.e., by changing edge weights, based on node attribute values. However, Zhang *et al.* and Newman *et al.* do not make use of attribute semantics as we suggest here. Hence, these approaches can not identify communities for different domains as required by the application discussed in section 5.

Our belief in external knowledge enhancing community detection in a network is rooted in past work that demonstrated the prominent role of semantics in social network analysis. For example, El *et al.* combines social data with data semantics to create a semantic social network [6]. Pool *et al.* argues that a knowledge graph-based description should inform community structures based on user interests and beliefs [25]. A survey on a semantic social network by Ereto *et al.* summarizes the use of semantics in social network analysis[7]. Palma *et al.* focuses on predicting drug targeted Interaction using semantic similarity and edge partitioning [24]. These approaches integrate the social network links with existing ontologies for generic social network analysis. However, community detection on such combined graphs can be biased with one graph (social graph or ontology) being larger than the other. Wang et al. reported that real-world knowledge represented in knowledge graphs could improve document clustering [28]. Nevertheless, they did not focus on community detection with links connecting nodes and attributes identifying nodes.

7 CONCLUSIONS AND FUTURE WORK

In this paper, we presented an algorithm to incorporate hierarchical concepts about node attributes into community detection. Our core contributions include (1) a combined metric that describes concept informativeness in the hierarchy and concept purity in summarizing communities, which are used to guide the search for optimal concept generalization; (2) a node similarity measure that synthesizes multiple generalized concepts for community detection; and (3) a community detection algorithm that alternatively optimizes concept generalization and community structures. Our evaluation results showed that concept generalization can not only improve the quality of community detection, but also provides a meaning-oriented characterization of community structure. The results vary depending on the choice of domains and knowledge sources. We demonstrated that readily available and automatically extracted knowledge source can also have vital improvements. An exciting direction to explore is to extend this approach to identify "path-based" and "diffusion-based" communities. Also, exploring the role of knowledge in other network analysis tasks such as influence detection.

8 ACKNOWLEDGMENTS

This work was supported by Army Research Office Grant No. W911NF-16-1-0300 and NSF Grant No. CNS 1513721. We thank Dr. Mohammad Akbari for helpful discussions.

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.

[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] 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.

[7] 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.

[8] Daniel G Goldstein, Randolph Preston McAfee, and Siddharth Suri. 2014. The wisdom of smaller, smarter crowds. In *Proceedings of the fifteenth ACM conference on Economics and computation*. ACM, 471–488.

[9] Ryan Hartman, Josemar Faustino, Diego Pinheiro, and Ronaldo Menezes. 2017. Assessing the suitability of network community detection to available metadata using rank stability. In *Proceedings of the International Conference on Web Intelligence*. ACM, 162–169.

[10] Richard J Hazler and Sharon A Denham. 2002. Social isolation of youth at risk: Conceptualizations and practical implications. *Journal of Counseling & Development* 80, 4 (2002), 403–409.

[11] 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

[12] 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.

[13] Sergey Ioffe. 2010. Improved consistent sampling, weighted minhash and l1 sketching. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*. IEEE, 246–255.

[14] 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).

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

[17] 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.

[18] 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.

[19] Shelley McKeown, Reeshma Haji, and Neil Ferguson. 2016. Understanding Peace and Conflict Through Social Identity Theory. *Contemporary Global Perspectives. Switzerland: Springer* (2016).

[20] 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.

[21] Galileo Mark Namata, Brian Staats, Lise Getoor, and Ben Shneiderman. 2007. A dual-view approach to interactive network visualization. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*. ACM, 939–942.

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

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

[24] 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.

[25] 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.

[26] David Sánchez and Montserrat Batet. 2012. A new model to compute the information content of concepts from taxonomic knowledge. *International Journal on Semantic Web and Information Systems (IJSWIS)* 8, 2 (2012), 34–50.

[27] James Surowiecki. 2005. *The wisdom of crowds*. Anchor.

[28] Chenguang Wang, Yangqiu Song, Ahmed El-Kishky, Dan Roth, Ming Zhang, and Jiawei Han. 2015. Incorporating world knowledge to document clustering via heterogeneous information networks. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1215–1224.

[29] Pengfei Wang, Jiafeng Guo, and Yanyan Lan. 2014. Modeling retail transaction data for personalized shopping recommendation. In *Proceedings of the 23rd ACM international conference on conference on information and knowledge management*. ACM, 1979–1982.

[30] 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

[31] 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.

[32] Wei Ye, Linfei Zhou, Xin Sun, Claudia Plant, and Christian Böhm. 2017. Attributed Graph Clustering with Unimodal Normalized Cut. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 601–616.

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