Knowledge Graph Enhanced Community Detection and Characterization

Shreyansh Bhatt, Swati Padhee, Amit Sheth, Keke Chen, Valerie Shalin, Derek Doran, and Brandon Minnery
Kno.e.sis, Wright State University
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.

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
Austin, "Dallas", and "Houston" are different, while they are same widely used community label assignment algorithm, the Louvain 1. There is no clear similarity measure for computing node similar-
odes and edges). (ii) The algorithm does not assume a priori that a 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 \( V \), edges \( E \), node attributes \( A \), and one or more domain-specific HKGs \( H = \{H_1, \ldots, H_l\} \) 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 \( v_j \in V \) is a list of concepts indicating attribute values \( v_j = \{c_1, c_2, \ldots, c_p\} \) such that each \( c_j \in v_j \) is part of at least one \( H_i \in H \). As an example, nodes \( V_1 \sim V_6 \) in Figure 1 are nodes and each \( v_j \) has one attribute \( c_j \) (city) that is
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 is iteratively optimized 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 discuss 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 $h_i$ with $o_{ij}$ 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 $1 - \frac{h_j}{\eta_{root}}$ where $\eta_i$ 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],

$$I_c = \left(2 - \frac{\sum_{l<e} \frac{1}{S_{lj}}}{\sum_{l<root} \frac{1}{S_{lj}}} \right)$$

Here, $S_j$ refers to the number of concepts that subsume $l$. The informativeness $I_1$ 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, $I_{CitiesinOhio} = 2 - \frac{1}{2 + \frac{1}{2}} = \frac{3}{4}$. 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 $c \in v_1$ 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 Jaccard similarity [13] between $v_{ext1}$ and $v_{ext2}$.

$$J(v_{ext1}, v_{ext2}) = \frac{\sum_i \min(v_{ext1}^i, v_{ext2}^i)}{\sum_i \max(v_{ext1}^i, v_{ext2}^i)}$$

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 $h_j$ knowledge graph representing community $i$. As an example, the similarity between $v_1 = \{\text{Cincinnati}\}$ and $v_2 = \{\text{Columbus}\}$ results in $v_{ext1} = \{(\text{Cincinnati}, 1.8), \text{Ohio}(1.6), \text{USA}(1.0), \text{Columbus}(0.0)\}$ and $v_{ext2} = \{\text{Columbus}(1.8), \text{Ohio}(1.6), \text{USA}(1.0), \text{Cincinnati}(0.0)\}$. 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 optional 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 ∈ 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 subsumes 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,

\[
\max_c \left( \eta_c - \eta_c \times \frac{\sum_{1 \leq c < \bar{c}} I_c}{\sum_{c < \text{root}} I_c} \right)
\]

Here, \( \eta_i \) 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:

\[
o_{ij} = \max_c \left( (\eta_{n \in i} - \eta_{n \in i}) \times I_c \right)
\]

where \( \eta_{n \in i} \) indicates the number of concepts in i subsumed by c and \( \eta_{n \in i} \) 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 \( \bar{T} = \{ c_1, c_2, \ldots, c_f \} \) 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 ∈ 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 ≥ 0.

Input: T, \( \bar{T} \), and h_j
Output: \( c_{opt}, s_{max} \)
\( c_{opt} = \text{root}, s_{max} = \text{root_score} \)
Associate score with each concept. -1 for T and 1 for \( \bar{T} \)
level = lowest_level()
list = add_list(T, \( \bar{T} \), level)
while list not empty and level ≥ 0 do
\[
temp_list = \text{empty}
\]
for c ∈ list do
s_{cur} = score(c) × I_c
update_optimal(\( c_{opt} \), s_{max}, c, s_{cur})
for p ∈ 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 ∈ 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} ≠ 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 ∈ L if 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: \( \sum_{l=1}^{k} \frac{E_{il}}{m_i} \left( e_i - a_i^2 \right) \). 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. \( \frac{E_{il} \times X_{il}}{m_i} \) 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.
Using the new community labels, we compute the optimal context representing each community $c \in K$. 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 Input: $G=(V, E, A), H = \{h_1, h_2, \ldots, h_3\}$, max_iters, threshold Output: $L, O, S$

\begin{algorithm}\
\textbf{while} \$\text{mod} < \text{threshold}\text{ or until max\_iters} \textbf{do}\
\text{\hspace{1cm} $w_{ij}(\omega, i, j, H, O, L) = \omega + \sum_{q=1}^{N} \phi(i, j, h_q, o_i, l)$}\
\text{\hspace{1cm} $E_{ij} = w_{ij}(\omega, i, j, H, O, L) + w_{ij}(\omega, i, j, H, O, L)$}\
\text{\hspace{1cm} $Q(E, L) = \frac{1}{2m} \sum_{k \in K} \sum_{f(i,j,l)} E_{ij} - \frac{d_i d_j}{4m^2}$}\
\text{\hspace{1cm} $L = max\_Q(E, L)$}\
\text{\hspace{1cm} for $i \in K$ do}\
\text{\hspace{2cm} $h_i \in H$ do}\
\text{\hspace{3cm} $o_{ij} = max\_Q((\eta_n e_i - \eta_n e_i) \times L_{c})$}\
\text{\hspace{1cm} end}\
\text{end}\
\end{algorithm}

\section{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. $P$ maps nodes of each community to a knowledge graph and computes an optimal context for each community resulting in the time complexity of $O(kc)$ 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 $\omega$ 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.

\section{4 EVALUATION}

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

\subsection{4.1 Datasets}

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

\subsubsection{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.

\subsubsection{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 â€” 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,

\[
\frac{1}{2\lvert C \rvert} \sum_{C_j \in C} \max_{C_i \in C} \delta(C_i, C_j) + \frac{1}{2\lvert C \rvert} \sum_{C_j \in C} \max_{C_i \in C} \delta(C_i, C_j)
\]

(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

\[
R_e = -\sum_{j=1}^{n_c} \frac{n_{cj} \log n_{cj}}{n_c}
\]

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 approaches. Like CPCD, KCDC 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.

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 \(s_i = \{(u_1, u_2)|u_1 and u_2 \in \text{same community}\}\). InterCommunitySet = \(\{d_1, d_2, \ldots, d_n\}\) where each \(d_i = \{(u_1, u_2)|u_1 and u_2 \in \text{different communities}\}\). We compared each \(s_i \in \text{IntraCommunitySet}\) to all the \(d_i \in \text{InterCommunitySet}\) resulting with \(n^2\) comparisons. Ideally, each \(s_i \in \text{IntraCommunitySet}\) should be higher than all the \(d_i \in \text{InterCommunitySet}\). The number of times \(s_i \in \text{IntraCommunitySet}\) is lower than \(d_i \in \text{InterCommunitySet}\) 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
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 of 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, for example, 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 $M_{ij}$ 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 make 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.
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.

<table>
<thead>
<tr>
<th>Crowd selection</th>
<th>Avg higher than % users</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>76%</td>
</tr>
<tr>
<td>DiversePositions</td>
<td>81%</td>
</tr>
<tr>
<td>DiverseTeams</td>
<td>86%</td>
</tr>
</tbody>
</table>

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>
<thead>
<tr>
<th>C(size)</th>
<th>Sports(Relevancy)</th>
<th>Music(Relevancy)</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1(47)</td>
<td>U.S. Women’s soccer(0.36)</td>
<td>Bob Marley(0.64)</td>
</tr>
<tr>
<td>C2(40)</td>
<td>Cleveland Browns(0.45)</td>
<td>Keke Palmer(0.55)</td>
</tr>
<tr>
<td>C3(38)</td>
<td>American Football in Boston(0.39)</td>
<td>Machine Gun Kelly(0.61)</td>
</tr>
</tbody>
</table>

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-textual 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.

A survey on a semantic 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 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.
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

[28] Chenguang Wang, Yangjou 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.