On Community Detection in Real-World Networks and the Importance of Degree Assortativity

Marek Ciglan
Institute of Informatics
Slovak Academy of Sciences
Bratislava, Slovakia
marek.ciglan@savba.sk

Michal Laclavík
Institute of Informatics
Slovak Academy of Sciences
Bratislava, Slovakia
michal.laclavik@savba.sk

Kjetil Nørvåg
Dept. of Computer and
Information Science
Norwegian University of
Science and Technology
Trondheim, Norway
kjetil.norvag@idi.ntnu.no

ABSTRACT

Graph clustering, often addressed as community detection, is a prominent task in the domain of graph data mining with dozens of algorithms proposed in recent years. In this paper, we focus on several popular community detection algorithms with low computational complexity and with decent performance on the artificial benchmarks, and we study their behaviour on real-world networks. Motivated by the observation that there is a class of networks for which the community detection methods fail to deliver good community structure, we examine the assortativity coefficient of ground-truth communities and show that assortativity of a community structure can be very different from the assortativity of the original network. We then examine the possibility of exploiting the latter by weighting edges of a network with the aim to improve the community detection outputs for networks with assortative community structure. The evaluation shows that the proposed weighting can significantly improve the results of community detection methods on networks with assortative community structure.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Data mining; E.1 [**DATA STRUCTURES**]: Graphs and networks

Keywords

community detection, network assortativity, edge weighting

1. INTRODUCTION

The goal of community detection in networks is to identify sets of nodes, *communities,* that are densely connected among themselves and have weaker connections to the other communities in the network. It is a task that can help to analyse large graphs and identify significant structures within and a classical example is to analyse social networks in order to find social groups of users. Community detection faces numerous challenges, the principal one is the lack of a consensus on the formal definition of a network community structure. The result of the ambiguity of the task definition is that a significant number of community detection algorithms having been proposed, using different quality definitions of a community structure or leaving the problem formulation in an ambiguous informal description only. In this paper, we do not propose yet another community detection algorithm, nor do we attempt to provide a new formalization of the task. Rather, we study community detection methods on *real-world networks* and the possibility of improving their precision by pre-processing the network topology.

Motivation. The motivation is that for a class of networks, the community detection techniques fail to deliver a good partitioning (in the rest of the paper, we use the term partitioning to address the result of a community detection algorithm; a partitioning is a set of detected communities). For example, when using community detection techniques to analyse the semantic network DBPedia[1], where nodes corresponds to the DBPedia concepts and edges denote a relation defined between two concepts, our expectation was that the analysis would reveal small clusters with semantically related concepts and entities. The clusters were expected to be, for example, similar to Wikipedia categories (containing groups of Wikipedia articles handpicked by human contributors and assigned to be a member of the category, class). However, the detected structure contains a few very large communities comprising the majority of the nodes.

Due to the size of the data sets, our choice of community detection methods for the network analysis was limited to a small family of fast community detection algorithms that are near-linear in the time complexity. We analysed the link graph using the label propagation [19] algorithm, a greedy modularity optimization algorithm [3] and a community detection method with parameterized community size constraint (SCCD) [4]. The community structure produced by the label propagation algorithm had the largest community, with over 2.96 million of nodes. The SCCD method with default setting yielded a structure with 78% of the nodes in the 20 largest communities, and the greedy modularity optimization method produced a partitioning with 88% of the nodes in the 20 largest clusters. The obtained results clearly did not match our expectations of the community structure of the DBPedia network.

Overview of the study. Motivated by the above observation, we analyse several large social and information networks with known ground-truth communities and compare the detected structure obtained by the three community detection algorithms to the ground-truth clusters. On four of the analysed networks, the detected clusters are a decent approximation of the ground-truth communities.

[1]Knowledge base derived from Wikipedia, http://dbpedia.org

KDD'13, August 11–14, 2013, Chicago, Illinois, USA.

For the rest of the networks, the similarity scores of the yielded partitions with the ground-truth communities are low, and at the same time, a few of the largest detected clusters contain the majority of the network nodes.

We then study the assortativity of the analysed networks and assortativity of their ground-truth communities. The interesting finding is that the assortativity coefficient of a network can be significantly different compared to the assortativity of its community structure. Based on this, we examine the possibility of modifying the network by means of edge weighting. The underlying idea is to examine whether we can increase the precision of the community detection algorithms by the weighting functions for the networks with assortative community structure. We describe the weighting heuristics and show empirically that it is a suitable approach in practice. On our test data, the similarity of the detected community structure to the ground-truth is significantly increased after applying the weightings on network with assortative community structure.

The main contributions of the paper are:

- We show that assortativity of the community structure can differ from overall assortativity of the network.
- We propose edge weighting functions designed to decrease the influence of edges connecting disassortative nodes. We show that such edge weighting on networks with assortative community structure can increase significantly the similarity of the communities identified by community detection methods to the ground-truth communities, an increase of 2 to 10 times compared to the baseline solution is reported.

The organization of the rest of the paper is as follows. Section 2 describes related work and the community detection methods with near-linear time complexity that are used in this study. Section 3 evaluates the precision of the algorithms on real-world networks with known ground-truth communities. The finding is that although on most of the studied networks the algorithms perform decently, there are networks for which the overlap of the detected clusters and the ground-truth communities is very low. The assortativity of the ground-truth community structures are examined in Section 4. In Section 5, we present the edge weighting designed to decrease importance of edges connecting disassortative nodes, and show that a significant increase in precision of community detection methods is observed on networks with assortative community structure. Finally, in Section 6 we conclude the paper.

2. PRELIMINARIES AND RELATED WORK

This section summarizes related work and preliminaries. We first discuss the community detection task and then provide more detailed overview of the three algorithms with near-linear time complexity that we use in our study.

2.1 Community Detection

The task of community detection is to find the community structure of a given input network. A community structure is considered to be a collection of clusters of densely connected vertices that are less densely connected to other parts (or communities) in the network. The problem has been a very popular research topic and has been extensively studied in recent years; a good evidence of the topic's popularity is the overview paper by Fortunato [8] with more than 450 references. Since its publication in 2010 (until May 2013), it has attracted more than 1400 citations according to the Google Scholar service. Despite the significant research effort on this problem, there is no consensus on the formalization of the task and authors often use different definitions of a community or even leave the notion of the community structure in an informal description. The most widely used approach is to focus on maximizing the modularity measure (introduced by Newman and Girvan[17]) that compares how community-like the partitioning of the input network is to a random network with the same degrees of vertices. Modularity is a quality function for estimating how good the partitioning of a network is. The basic formulation of the community detection task expects as an output a partitioning of a network; that is, each node is a member of exactly one community. Numerous variants of the problem have been studied, including detection of overlapping communities where a vertex can belong to multiple communities (e.g., works by Gregory [9] and Zhang et al. [22]), clustering of bipartite graphs (e.g., Papadimitriou et al. [18]), and detection of clusters exploiting additional information in addition to network structure (e.g., attributes on nodes/edges, Yang et al. [21]).

Community-detection algorithms are usually evaluated against artificial benchmark graphs, where a community structure has been injected (e.g., [11]). The advantage is that the evaluator can tune the parameters of the generated network, the disadvantage is the artificiality itself. The real-world networks with known community structure studied in the literature are usually small ones (e.g., Zachary's karate club (36 nodes) or Dolphin social network (62 nodes)), with few exceptions; e.g., in a recent study by Yang and Leskovec [20], the authors identify the ground-truth communities for several large networks and study their properties. In this work, we reuse their data sets. Abrahao et al. in [1] study structural properties of the ground-truth communities and compare them with the properties of clusters discovered by several community detection approaches. The finding is that the communities produced by different approaches are clearly separable according to the studied properties, with the approaches based on the random walk producing communities with structural properties the most similar to the ground-truth groups. Their study focus on communities with less than 1000 nodes. The aim of our work lies in overcoming the problem of collapsing the majority of network nodes into a few huge groups.

Leskovec et al. in [14] provide an empirical comparison of community detection algorithms, studying community quality scores of clusters detected by various algorithms, with the focus on the conductance measure. One of the main finding in their study was that although community detection methods often optimize the clusters quality scores nicely, there are classes of networks where the community detection techniques perform sub-optimally. Our work confirms that observation for the near-linear community detection approaches and we show that, to a certain extent, we can overcome that behaviour. We demonstrate that a weighting of the edges, based on the assortativity of connected nodes, can increase the precision of the studied algorithms for a class of networks. The concept of edge weighting as a preprocessing for community detection has been explored before, e.g., in [2][10] where authors use different approach to the weighting; they re-weight the edges based on edge betweenness centrality and common neighbour ratio.

2.2 Algorithms with Near-Linear Complexity

A large number of the community detection methods proposed in the literature are heuristics with polynomial time complexity. In practice, the time complexity often limits the usability of a large part of community detection algorithms to small networks (e.g., see the comparative study in [12]). In our work, we intend to study the behaviour of community detection methods on large real-world networks, which limits our options to a small family of community detection techniques with near-linear time complexity. This section describes the fast greedy algorithms we have used. The

discussed algorithms are the label propagation approach by Raghavan et al. [19], a heuristic for modularity optimization by Bondel et al. [3] and a heuristic for Size-constrained Community Detection (SCCD) [4]. The base method is the label propagation (LP), the two other use the same underlying principle as LP with several modifications.

Label propagation algorithm. The label propagation approach is based on the simple idea that a node should be assigned to the community to which most of its neighbours belong to. When starting from scratch, i.e., the community structure is unknown, the label propagation algorithm assigns a unique label to all the vertices. In randomized order, the algorithm iterates over vertices and recomputes the label for each vertex in the following way: the number of neighbours with distinct labels is computed and then the vertex label is set to the label with the most members. It is repeated in iterations in which the community membership of all the nodes are updated in random order. If, during an iteration, none of the vertices change their label, the algorithm stops. However, there might be nodes changing their membership in each iteration (e.g., nodes with equally strong ties to several stable clusters), so this termination condition is not sufficient. It has been argued by the authors that in practice a good community structure is found after a few iterations and we can set the maximal number of the iterations to be performed. This results in linear time complexity (a constant number of iterations over N vertices). The problem of the method is that one (or a few) label(s) often becomes dominant and the majority of the network collapses into one single community.

Louvain method. The Louvain method proposed by Bondel et al. in [3] has two phases. The first one is based on a process very similar to the label propagation. The main difference is the label assignment, the Louvain method does not use the number of neighbours with the same labels as principal factor. Rather, it computes a possible gain in the modularity in case a vertex changes its label to other given label. In the second phase, a new community network is constructed by contracting community members into single nodes. The first phase is then used on the derived network and a hierarchy of communities is constructed.

SCCD. The Size-Constrained Community Detection method [4] is another method based on the label propagation principle. The main difference is in the scoring function that governs the label assignment of a node, where the SCCD method discriminates the scores according to the sizes of target communities. The underlying idea of the approach is to increase the ability to detect small-sized, compact clusters independently of the network size, as opposed to the modularity optimization methods that are known to have a resolution limit and tend to increase the size of generated communities with the increase of the network size.

All these three algorithms have near-linear time complexity, and the two latter have a confirmed decent precision on the artificial benchmark networks [4, 12], competitive to other well-performing methods.

3. DETECTED COMMUNITIES VS. GROUND-TRUTH CLUSTERS

In this section, we evaluate the three studied algorithms on large, real-world data sets with ground-truth clusters. The main motivation is to observe whether the partitioning produced by community detection algorithms approximate well the ground-truth communities. We consider this section particularly interesting as this exercise puts algorithms to a tough test because of the large sizes of the networks that are studied. We first describe the data sets, focusing mainly on how the ground-truth communities were identified, and introduce a ground-truth communities data set for the DBPedia knowledge graph. We then describe the evaluation functions used to assess the quality of the detected partitioning compared to ground-truth communities. We summarize the results of the analysis and discuss questions raised by the experiment; namely, the suitability of the ground-truth communities data for the community detection task and the failure of the used community detection methods to approximate well ground-truth communities for a class of networks.

3.1 Data Sets

We reuse five networks with ground-truth communities from the Stanford Large Network Dataset Collection (SLNDC) and we introduce ground-truth communities for the network based on DBPedia.

SLNDC data sets. We have used five graphs from the SLNDC collection, containing the ground-truth communities introduced in [20]. For the LiveJournal (LJ) social network containing friendship networks, user-defined groups are considered as the ground-truth communities. Similarly, user defined groups are considered as ground-truth communities in the Orkut and YouTube datasets. The DBLP dataset provides co-authorship network. Here, the ground-truth communities are created by grouping authors publishing on same venue or in the same journal. The Amazon data network contains products as nodes and the edges indicate a co-purchasing relation. The ground-truth clusters are equivalent to the product category in Amazon. In all the data sets, the identified groups were further split to connected components and each connected component is regarded as a distinct ground-truth community. In addition, all communities with less than 3 nodes were removed.

Ground-truth communities for DBPedia. We describe the data set containing ground-truth communities for the DBPedia separately, as it has not been used so far in the context of community detection. DBPedia is a knowledge base derived from Wikipedia, mostly by parsing the infoboxes of Wikipedia articles. It can be viewed as a graph of interconnected entities, where the entities can have properties (e.g., a concept related to a person can have attributes such as birth date, height, occupation) and are linked by labelled relations or edges to other entities. In our previous work on ad-hoc retrieval from semantic data we were faced with the challenge to extract semantically related sets of entities from the DBPedia knowledge base [5]. As the first approximation we took the members of Wikipedia categories as such semantic sets. In DBPedia, the category membership translates into relation labelled 'subject' connecting members to the entity representing the category. This approach has two disadvantages: a) even though a large number of Wikipedia categories group semantically related entities, there are many trivial categories (e.g., Category:1970_births containing people born in 1970 - those entities can hardly be considered similar in other respects than the date of the birth); b) data set size - the categories contain only a subset of such semantically related sets (e.g., for some music bands, there could be a category grouping 'members_of ..', while there is no such category for a number of other entities of the same type).

We have approached b) by exploiting the semantic relations (labelled edges) in DBPedia to identify additional potentially semantically related sets. We have selected all the sets of vertices that fit to the following two patterns: a set of vertices connected by an outgoing edge of the same label to a common vertex v, or a set of vertices that have an incoming edge of the same label from a single vertex. We have used all the labels, except the '*wikilink*' label that denotes an existence of a hyperlink between the two Wikipedia articles, but the true semantics of the relation is not given.

The remaining problem was to distinguish good semantic groups and the trivia groups. We have used several similarity scores to measure the relatedness of the entities in the sets. We have used text-based similarity (as the DBPedia nodes contain also abstracts of related articles, we have quite a rich textual component) as well as structural similarities (using the topology of the DBPedia graph). The details are provided in [5], and to summarize, most of the used similarity measures had a high correlation, even the text-based cosine similarity with the structural similarities. We then chose the sets with similarity scores above a certain threshold.

We exploit the data set of the semantically related entities and use them to identify the ground-truth communities. From the candidate set we select groups with high score, computed as the product of internal density measure and 1-conductance. The internal density expresses how clique-like is the subgraph generated by the given set of vertices; more formally, let $a_{i,j}$ be an element of the adjacency matrix for G, then the internal density is

$$\psi(S) = \sum_{i,j \in S; i \neq j} a_{ij} / |S| \times |S - 1|$$

The conductance measure expresses how well a group of nodes is separated from the rest of the network. It is defined as the ratio of edges outgoing from the given set of nodes to the rest of the graph and total number of edges outgoing from the given set of vertices. More formally, let $a_{i,j}$ be an element of the adjacency matrix of G and $a(S) = \sum_{i \in S} \sum_{j \in V} a_{ij}$, the conductance can be defined as:

$$\varphi(S) = \frac{\sum_{i \in S, j \in \overline{S}} a_{i,j}}{min(a(S), a(\overline{S}))}$$

We have removed candidate sets with less than 3 members and having score lower than 0.1. We follow the approach taken in [20] and produce a reduced set, containing only the top 5000 ground-truth communities; the candidate sets were ranked by decreasing scores and the top 5000 were selected. The data sets are provided on the support web-page[2].

3.2 Comparing Detected and Ground Truth Clusters

For comparing the detected and ground-truth communities, we use two measures. The first one is based on set similarity and is introduced in the following text; the second one is Normalized Mutual Information which is popular in the community detection literature. We report values for both in the presented results.

Comsim - Set similarity based measure. The algorithms studied in this work output a partitioning of a network, where each node is assigned to exactly one community. The ground-truth communities in our collection contain groups of nodes that can have non-empty intersection, i.e. a vertex can be a member of multiple communities. In addition, not all the vertices of the network have to belong to a ground-truth community. To compare such structures, we need a function that measures their similarity. In addition, the ability to assess approximation of the ground-truth communities one by one would be of advantage, allowing us to analyse the produce results on a cluster level, rather than on the partitioning level. To achieve that, we compute the similarity score of each of the ground-truth communities with the most overlapping group from the network partitioning identified by community detection algorithm. The overall score would be an average of the similarity scores of distinct ground-truth communities. As we need to compare two subsets, the straightforward approach is to use Jaccard's similarity coefficient: $J(A, B) = |A \cap B| / |A \cup B|$. The remaining point to solve, is how to find the best fitting cluster from a network partitioning for a given set. We select the cluster with the largest intersection with the given ground-truth set. In case there are multiple sets with the same maximal size of intersection, we select the smallest one.

[2]http://ups.savba.sk/~marek/communities.html

Table 1: Similarity scores - comsim and NMI (in parenthesis) for the detected partitions and the ground-truth data sets (reduced - top5000 and full set of ground-truth communities).

	Louvain	Label Prop.	SCCD
DBLP-top5K	0.53 (0.24)	0.49 (0.25)	0.51 (0.26)
Amazon-top5K	0.76 (0.37)	0.85 (0.46)	0.86 (0.46)
YouTube-top5K	0.23 (0.09)	0.08 (0.02)	0.19 (0.05)
Orkut-top5K	0.04 (0.03)	0.03 (0.02)	0.24 (0.06)
LJ-top5K	0.52 (0.28)	0.57 (0.32)	0.60 (0.32)
DBPedia-top5K	0.004 (0.0008)	0.003 (0.003)	0.13 (0.06)
DBLP-all	0.34 (0.12)	0.32 (0.14)	0.32 (0.14)
Amazon-all	0.42 (0.28)	0.26 (0.24)	0.28 (0.25)
YouTube-all	0.11 (0.03)	0.03 (0.01)	0.10 (0.02)
Orkut-all	0.001 (0.05)	0.0002 (0.04)	0.01 (0.03)
LJ-all	0.03 (0.02)	0.01 (0.03)	0.04 (0.02)
DBPedia-all	0.004 (0.005)	0.003 (0.004)	0.06 (0.02)

More formally, let $G = (V, E)$ be the graph, let $D = \{P_1, P_2, \ldots, P_k | \forall P_i, P_j : P_i \subset V, P_j \subset V, P_i \cap P_j = \emptyset \wedge \bigcup_{i=1..k} P_i = V\}$ be the partitioning produced by the community detection algorithm, let $E = \{T_1, T_2, \ldots, T_l | \forall_i T_i \subset V\}$ be the set of ground-truth communities. Let $o(T, D)$ be a set of clusters from D having maximal intersection with T :

$$o(T, D) = \{P : P \in D \wedge \forall_{P_i \in D} |P_i \cap T| \leq |P \cap T|\}$$

The best fitting clusters for a ground-truth community T is:

$$b(T, D) = \{P : P \in o(T, D) \wedge \forall_{P_i \in o(T,D)} |P_i| \geq |P|\}$$

Fitting score for a ground-truth community T is then $sim(T, D) = J(T, B) : B \in b(T, D)$. Overall score for D and E is

$$comsim(E, D) = \frac{\sum_{T_i \in E} sim(T_i, D)}{|E|}$$

Normalized mutual information. The second measure we use is Normalized Mutual Information (NMI) [6]. It is a standard measure used in the community detection literature to compare two partitionings of a network. It has been designed to evaluate non-overlapping partitions. Lancichinetti et al. in [13] have proposed generalized NMI in order to able to compare overlapping communities as well, and we also report the latter in our experiments.

Evaluation. The number and quality of the ground-truth communities can vary for different data sets. We follow the approach of [20] where they selected the top 5000 communities for each of the ground-communities data set, based on average rank of the community for six different community scores. Extracting and using the top k communities enables experimentation with high-quality clusters, and we have selected the top 5000 communities for the DBPedia data set as well. In our case, we have used the rank of the community according to conductance and internal density measures. For the evaluation purposes, we have for all the networks used both the complete and top-5000 ground-truth communities data sets. We have used the three studied community detection algorithms to cluster the six networks and we have compared the resulting partitionings with ground-truth communities with $comsim$

score and generalized *NMI* measure. Approaches based on label propagation are sensitive to the order in which the nodes are processed, so we have used the same seed for the random number generator to ensure the same order of nodes so that different algorithms process the nodes in the same sequence.

The results are summarized in Table 1, where the top half presents scores for the top 5000 community sets, the bottom half for full ground-truth communities sets. Scores for the top 5000 communities are, not surprisingly, significantly higher than the scores for the data sets with all ground-truth. Another important observation is that for some networks, the similarity score is quite high (DBLP, Amazon, LJ, YouTube), for the Orkut and DBPedia data sets, the scores are low. There are two possible explanations that we are going to discuss in more detail. First possible explanation could be that the ground-truth data sets for the three networks with low scores are flawed, and do not capture the real communities within the networks and are unsuitable for evaluation of the community detection task in general. The second explanation could be that the ground-truth communities are suitable for the evaluation, but the studied algorithms have been unable to approximate the network community structure precisely.

Suitability of the used ground-truth communities for the community detection task. The ground-truth communities were derived from network data where nodes where explicitly assigned to sets (e.g., user-defined groups). As the results in Table 1 indicate, the community detection algorithms were not very successful in approximating the ground-truth communities. A legitimate concern is whether the used data sets are suitable for testing community detection in general, and the question is whether the assignment to ground-truth communities corresponds to the community structure of a network. There is no consensus on the formal definition of a community structure, the general notion is that a good community has nodes that are densely linked among themselves and are less densely linked to other communities in the network. The latter requirement becomes questionable under the model of overlapping communities. We thus focus on the notion of a community as a densely linked group of nodes. Internal density function is a suitable measure to capture how clique-like a community is (cf. the measure as described in Section 3.1). We have computed the internal density of the ground-truth communities and have compared it to internal density of a same sized group of nodes in a random graph with identical number of network elements (nodes and edges). The results are presented in Table 2 and show that the average internal density of ground-truth communities is high, orders of magnitude higher than in the case of random graphs. The conclusion is that the ground-truth communities used in the experiments are related to the informal notion of a community as a densely connected subgraph. A possibility that cannot be dismissed from the results in Table 2 is that some of the ground-truth sets are possibly subsets of the 'real' communities. Even in that case, an important overlap with the correctly identified communities should exist.

Failure of community detection algorithms. Based on the observations above, we conclude that the low scores for community detection methods on three networks indicate their inability to detect the communities precisely. Observations of classes of networks where the community detection techniques perform sub-optimally has been also reported in [14]. Based on our preliminary observation from the introduction about the sizes of top largest detected communities on DBPedia, we have looked at the community sizes for the partitionings detected on the six networks with ground-truth communities. Table 3 presents the sizes of top 50 largest communities in the partitionings obtained by the studied methods and in the ground-truth communities data sets. As the results show, for the Orkut and DBPedia data sets, the majority of the nodes have been assigned to a few of the largest clusters. For those networks, also the similarity scores of studied methods with the ground-truth communities have been very low.

Table 2: Average internal density for the ground-truth communities and equally sized groups of nodes in random graphs. Columns prefixed by '$top5k$' report numbers related to the top 5000 best communities, columns prefixed by 'all 'concerns all the ground-truth communities in the data set.

	$top5k$	$tok5k - RND$	all	$all - RND$
DBLP	0.71	9.17×10^{-4}	0.52	2.21×10^{-3}
Amazon	0.62	4.89×10^{-4}	0.49	8.99×10^{-4}
YouTube	0.35	2.31×10^{-4}	0.37	1.64×10^{-4}
LJ	0.78	2.54×10^{-4}	0.43	2.77×10^{-4}
Orkut	0.31	1.00×10^{-2}	0.45	1.00×10^{-3}
DBPedia	0.98	8.34×10^{-5}	0.32	1.58×10^{-4}

Table 3: Fraction of nodes belonging to the top 50 communities in the partitionings identified by the methods we study and in the ground-truth communities data sets.

	Lou fract. of N	LP fract. of N	SCCD fract. of N	Ground-truth fract. of N
DBLP	0.11	0.07	0.05	0.32
Amazon	0.03	0.03	0.04	0.83
YouTube	0.50	0.66	0.17	0.01
LJ	0.67	0.71	0.07	0.09
Orkut	0.97	0.99	0.93	0.09
DBPedia	0.98	0.96	0.89	0.001

4. ASSORTATIVITY OF THE COMMUNITY STRUCTURE

As shown in previous section, the studied community detection algorithms failed to deliver a decent approximation of the ground-truth community structure for the DBPedia and Orkut networks. The detected partitionings for both networks also contained the majority of the nodes in a few huge communities. A possible explanation of the tendency of all examined label-propagation-based algorithms on DBPedia and Orkut to collapse the majority of the nodes to a small number of groups is that it is caused by a large number of edges connecting the majority of nodes to the high degree nodes in the dense core of the network. In the iterative process of the label propagation, the nodes are gradually reassigned to the communities centred around the dense network core. We explain the intuition behind this hypothesis in the following subsection.

4.1 Microscopic Look at the Data Set

We first look at an example of a DBPedia node to gain an intuition on what is the 'meaning' of the edges adjacent to a node with respect to the degrees of nodes connected by those edges (edges in our DBPedia dataset correspond to hyperlinks between Wikipedia articles). We have selected a node at random and we look at the related Wikipedia article. Here are the few first sentences from the chosen article, links to other articles are enclosed by [and] symbols:

Golden Child (play)

Golden Child is an [Obie Award]-winning play by [American] [playwright] [David Henry Hwang]. The play was developed [Off-Broadway] and premiered there on November 19, 1996 at the

[Joseph Papp Public Theater]. It was directed by [James Lapine], with [Tsai Chin] and [Jodi Long] in the cast.

Let us assume a human evaluator is assigned with the task of ranking the linked entities by their importance/relation to the topic of the article (*Golden Child (play)*). He would probably mark the author of the play, the director and actresses as more strongly related than the theatre and the production company, which in turn would be probably ranked as more strongly related than the concepts 'Playwright' and 'United States'. We can look on the degrees of the related nodes in our DBPedia data set: *[Golden Child (play); 37], [Obie Award, 731], [United States, 650 509], [Playwright, 5332], [David Henry Hwang, 389], [Off-Broadway, 703], [Joseph Papp Public Theater, 520], [James Lapine, 299], [Tsai Chin (actress), 103], [Jodi Long, 61]*. The most unrelated concepts to the entity '*Golden Child (play)*' are probably those with very high degree. One might even formulate a hypothesis that the node degree in DBPedia is correlated with the topic generality of the related Wikipedia article.

In our example case, if we try to sort the list of entities according to the difference in degrees between the target and the inspected node, we would probably receive a decent approximation of a human judgement of their relatedness to the examined node (Wikipedia topic). We do not want to draw any conclusions from this simple example, it only gives us a hint that a degree of connectivity of the low-degree nodes to high-degree nodes might affect the outcome of label propagation-based algorithms.

4.2 Assortativity of a Community

There is a well established measure to study how the nodes of different degrees are connected in the network, the *degree assortativity coefficient*. The degree assortativity coefficient (AC) denotes a tendency of nodes to be connected with other nodes of similar degree. It is defined as the Pearson correlation coefficient of degrees of pairs of nodes connected by an edge in the network ([16]). Let M be the number of edges, j_i and k_i be the degrees of the i-th edge, the assortativity coefficient can be computed as follows:

$$r = \frac{M^{-1}\sum_i j_i k_i - [M^{-1}\sum_i (j_i + k_i)/2]^2}{M^{-1}\sum_i (j_i^2 + k_i^2) - [M^{-1}\sum_i (j_i + k_i)/2]^2}$$

We examine the assortativity of the networks with ground-truth communities more closely. Namely, we compute the assortativity of the whole networks and assortativity of the community structure; i.e. for the latter, we use only edges belonging to ground-truth communities, while keeping the original node degree for the computation of the assortativity coefficient. The results are reported in Table 4, showing assortativity coefficient of the whole network, assortativity computed using only top 5000 clusters, and assortativity computed using all of ground-truth clusters.

The results show that the assortativity of the community structure can be very different compared to the assortativity of the original network. For example, YouTube and DBPedia are disassortative, while having assortative community structure. Five of the analysed networks have positive assortativity coefficient of the ground-truth communities, four of which are strongly assortative. The interpretation is that the networks with assortative community structure have important parts of communities composed of edges connecting nodes with similar degrees.

5. EDGE WEIGHTING

In Section 3 we studied the similarity of the partitionings yielded by the community detection algorithms and the ground-truth communities. The results revealed that for the DBpedia and Orkut networks, the similarity scores are very low. In the previous section, we showed that those two networks have an assortative community structure, which means that the communities they comprise are composed of an important portion of edges connecting nodes with similar degrees. Based on these observations, we examine the possibility of modifying the network structure by weighting its edges, in a way that lower the weight of the edges connecting disassortative nodes. The hypothesis is that, for networks with assortative community structure, such a modification could affect positively the similarity of the detected clusters with ground-truth clusters. We first propose two edge-weighting functions designed to lower the influence of the edges connecting low and high degree nodes. We then study the effects of such weighting on the network. We study the clusters detected on the weighted versions of the networks by the community detection algorithms; we compute their similarity to the ground-truth communities and compare it to the results obtained from original, unweighted networks.

Table 4: Assortativity of the networks with ground-truth communities and the assortativity of their community structures.

	net AC	Top5k Comm. AC	All Comm. AC
DBLP	0.267	0.436	0.446
Amazon	-0.059	-0.077	-0.026
YouTube	-0.037	0.067	0.068
LJ	0.045	0.464	0.365
Orkut	0.016	0.233	0.326
DBPedia	-0.018	0.958	0.973

5.1 Weighting Heuristic

So far, we have analysed the networks without weights on edges, which is equivalent to the situation where weights are equal on all the edges. In the following, we propose heuristic weighting functions, designed to decrease the importance of edges connecting nodes with very different degrees, which could lead to an increase of the network's assortativity. Instead of the vertex degree, we will compute the assortativity of the sum of weights of edges adjacent to a vertex; we will refer to it as weighted degree assortativity. It is a necessary change to capture the effect of the weight on edges. For most of the networks with ground-truth communities, we have no information other than the network structure (that is, we do not have any attributes on nodes or edges that could be used to derive nodes similarity). We thus have to base our weighting functions purely on the network structure.

The assortativity measures the tendency to link similar nodes, and the similarity in our case is the sum of edge weights. We thus try to use weighting functions that would penalize the links between highly disassortative vertices, i.e., vertices with very different degrees. The underlying idea is to decrease the importance of edges connecting disassortative nodes. We propose two weighting functions that differ in the degree of penalization of the edges linking disassortative nodes. Our first experimental weighting function would be 10^{1-x}, where x is the number of digits in a decimal notation of the division of nodes degrees.

More formally, let $d(v)$ denote the degree of a vertex v. Let

$$f_1(z, y) = \mathit{floor}(log_{10}(max(z, y)/min(z, y))$$

The weighting function $w(e_{i,j})$ is:

$$w1(e_{i,j}) = 10^{-f_1(d(i),d(j))}$$

In practice, this weighting would assign weight of 1 to an edge connecting nodes with the degrees of a same magnitude, weight

Table 5: Similarity scores of ground truth communities with the detected communities on: original unweighted network (*Orig*), on network re-weighted with *w1*, on network re-weighted with *w2*. Reported values are the similarity measures *Comsim* and *NMI* (in parenthesis).

Network	Alg.	Orig	w1	w2
DBLP	Lou	0.54 (0.24)	0.54 (0.26)	**0.56** (0.26)
	LP	0.49 (0.25)	0.49 (0.24)	0.53 (0.25)
	SCCD	0.52 (0.26)	0.51 (0.25)	0.54 (0.25)
Amazon	Lou	0.76 (0.37)	0.84 (0.43)	0.83 (0.42)
	LP	0.85 (0.46)	0.85 (0.46)	0.83 (0.44)
	SCCD	**0.86** (0.46)	0.86 (0.46)	0.83 (0.44)
YouTube	Lou	0.23 (0.10)	0.23 (0.11)	**0.31** (0.12)
	LP	0.08 (0.02)	0.14 (0.05)	0.24 (0.09)
	SCCD	0.19 (0.06)	0.21 (0.08)	0.26 (0.10)
LJ	Lou	0.52 (0.28)	0.49(0.27)	0.52 (0.28)
	LP	0.57 (0.32)	0.59 (0.32)	0.62 (0.32)
	SCCD	0.6 (0.32)	0.61 (0.32)	**0.62** (0.33)
Orkut	Lou	0.04 (0.03)	0.04 (0.04)	0.17 (0.05)
	LP	0.03 (0.02)	0.03 (0.03)	0.11 (0.04)
	SCCD	0.24 (0.06)	0.20 (0.06)	**0.25** (0.06)
DBPedia	Lou	0.004 (0.0008)	0.016 (0.03)	0.053 (0.15)
	LP	0.003 (0.003)	0.054 (0.14)	0.31(0.17)
	SCCD	0.13 (0.06)	0.26 (0.15)	**0.34** (0.18)

of 10^{-1} to an edge connecting a node with degree one order of magnitude lower that the other's node degree, and so on.

The second weighting function assigns a weight of 10^{1-x}, where x is the number of digits in a decimal notation of the difference of nodes degrees. Let

$$f_2(z, y) = \mathit{floor}(log_{10}(|z - y|)$$

The weighting function $w(e_{i,j})$ is:

$$w2(e_{i,j}) = 10^{-f_2(d(i),d(j))}$$

This weighting assigns weights as follows: edge linking nodes with $|d(i) - d(j)| < 10$ would be assigned with weight 1, edge $e_{i,j}$ with $|d(i) - d(j)| < 100$ would be assigned with weight of 0.1, and so on. The difference between $w1$ and $w2$ can be significant, e.g., if we compute weights for an edge connecting nodes with the degrees of 50 and 600; $w1$ would give the weight of 0.1 while the weighting $w2$ would give 0.01.

5.2 Detected Community Structure of Weighted Networks

The next step is to analyse the effects of the weighting on the results of the community detection method compared to the ground-truth communities. We compare the similarity scores ($comsim$ and *NMI*) achieved by the very same algorithms on the original unweighted networks with the scores achieved on the weighted versions of the same networks.

The results are summarized in Table 5. For each network, we list the similarity scores of the ground-truth communities with the partitionings detected by community detection methods on: original unweighted network (column *Orig*), on network re-weighted with function $w1$ and on network re-weighted with function $w2$. We provide results of the comparison for the sets of top 5000 ground-truth communities. We highlight with a bold font the best similarity scores for all the networks and we underline the important improvements in similarity score achieved by weighting. The first observation is that there are networks (DBLP, Amazon and LJ and YouTube data sets) for which the community detection methods achieve high similarity scores with the ground-truth communities even without weighting. For those networks, the weighting caused small increases in similarity scores (in several cases the score has been marginally decreased). The best performing algorithm for this class has been the Louvain method. In this context, it is interesting to mention the work by Dunbar [7], that states that the degree in social networks can be frequently limited by the inability of the actors to maintain a large number of connections. This is reflected by the positive assortativity coefficient of the network and might be the cause of low effect of the proposed weighting schemes. The second class, containing the Orkut and DBPedia data sets, was quite challenging for the community detection techniques and the similarity score were very low on unweighted networks. After the re-weighting, the detected communities were significantly better approximations of the ground-truth communities. For the SCCD and Louvain methods on top 5000 datasets, the similarity score increased 2 to 14 times. For the label propagation method, the increase is even more dramatic (e.g., factor of 116 for DBPedia data set). The reason for such a dramatic increase in similarity is that the algorithm often collapses the majority of the node's network into a single community. The weighing of the edges was a successful strategy to prevent this behaviour. We can conclude that for the networks on which the algorithms were achieving marginal similarity scores, the weighting had a very positive effect on the similarity of the detected communities with ground-truth.

Figure 1 depicts similarity scores for distinct communities in the ground-truth data sets with the best fitting clusters in the detected network partitionings for the Orkut, DBPedia and Amazon data sets. The x-axis is the rank of a ground-truth community, the y-axis depicts the Jaccard similarity of the compared sets. The figure visualizes how the weighting improved similarity of the detected clusters with the ground-truth.

5.3 Comparison with other Weighting Schemes

The results reported in Table 5 shows that the simple weighting functions, using only nodes degrees, can significantly improve the clustering results. In the next study, we look at how the proposed weighting compares to random weighting. In addition, a comparison with other relevant weighting mechanism would be helpful and illustrative. The natural candidates were the weightings proposed in [2, 10], however we did not used those weighting schemes due to their computational complexity, which makes them unsuitable for the size of analysed networks. Instead, we compare with a measure used by Milne and Witten in [15] to assess the similarity of the concepts in Wikipedia (we will refer to this measure as *MW* measure). Although, as to our best knowledge, it has not been used in the context of community detection so far, it exploits network topology to estimate the similarity of the nodes, has a proven good performance for node similarity estimation and it is computationally inexpensive. Let a and b be the nodes and sets A and B be the sets of nodes that link to a and b. The *MW* measure is:

$$\mathit{MW}(e_{a,b}) = \frac{log(max(|A|, |B|)) - log(A \cap B)}{log(|V|) - log(min(|A|, |B|))}$$

The results are reported in Table 6, we summarize the the scores achieved on weighted versions of networks, using the following weighting schemes: $w2$, random weighting (rnd), *MW* weighting and a linear combination of $w2$ and *MW*. An interesting observation from this experiment is that the methods are quite resilient to the random weighting and the results are very near the original

Table 6: Similarity scores of ground truth communities with the detected communities on networks weighted by following schemes: $w2$, random weighting rnd, MW weighting and $w2 \times MW$ Reported values are similarity measures *Comsim* and *NMI* (in parenthesis).

Net	Alg.	$w2$	rnd	*MW*	*MW* $\times$ $w2$
DBLP	Lou	0.56	0.47(0.23)	0.54 (0.26)	0.56(0.26)
	LP	0.53	0.48(0.23)	0.56 (0.27)	0.56(0.26)
	SCCD	0.54	0.52(0.24)	0.57 (0.27)	**0.57**(0.27)
Amazon	Lou	0.83	0.83(0.43)	0.85 (0.43)	0.83(0.42)
	LP	0.83	0.82(0.43)	0.85 (0.44)	0.84(0.43)
	SCCD	0.83	0.82(0.42)	**0.85** (0.44)	0.83(0.42)
YouTube	Lou	0.31	0.15(0.05)	0.20 (0.09)	0.30(0.12)
	LP	0.24	0.11(0.04)	0.17 (0.07)	0.29(0.11)
	SCCD	0.26	0.21(0.06)	0.32 (0.13)	**0.32**(0.12)
LJ	Lou	0.52	0.50(0.27)	0.53 (0.28)	0.53(0.29)
	LP	0.62	0.56(0.31)	0.61 (0.33)	0.63(0.33)
	SCCD	0.62	0.62(0.33)	0.63 (0.34)	**0.63**(0.33)
Orkut	Lou	0.17	0.06(0.03)	0.08(0.03)	0.13(0.04)
	LP	0.11	0.03(0.02)	0.06(0.03)	0.18(0.04)
	SCCD	**0.25**	0.22(0.05)	0.16 (0.04)	0.19(0.05)
DBPedia	Lou	0.05	0.03(0.01)	0.05(0.03)	0.29(0.15)
	LP	0.31	0.001	0.33	0.40
			(0.001)	(0.16)	(0.20)
	SCCD	0.34	0.15(0.06)	0.36 (0.18)	**0.41**(0.20)

ones. Another finding is that the *mw* measure, when used for edge weighting, can also increase the precision of the studied community detection methods. The highest score for DBPedia data set has been achieved by the linear combination of our $w2$ and *MW* measure, indicating that $w2$ brought additional discriminative information to the *MW* measure, which uses the commonalities in the neighbourhood vectors to assess weights.

5.4 Discussion

The goal of the work has been to verify whether weighting of the edges can help in better approximating the clusters in networks with assortative community structure. An important observation is that both proposed weighting functions caused significant increase in similarity scores for the partitionings detected by the studied algorithms for the DBPedia and Orkut networks. For those two networks, the proposed topology-preprocessing allowed increasing the similarity of the detected communities from marginal values. Interesting improvements can be observed also for YouTube. The DBLP, Amazon and LJ networks received high similarity scores for the partitioning detected from original unweighted networks. After the weighting, the scores for those networks stayed at approximately the same values. For the Amazon network, which has a slightly disassortative community structure, the results after the weighting even marginally decreased. The weighting function $w2$ is more aggressive in penalizing the links between disassortative nodes and it has shown to cause higher scores than $w1$.

We would also like to discuss how the proposed weighting functions can be interpreted. A valid question on the presented work might be: A graph on real collaboration networks such as DBLP will have edges between nodes that have large differences in node degrees. Consider for example the case of a young researcher who collaborates with a well-known senior researcher, what will be the effect of weighting discriminating such relation? The important point is that the approach does not remove the relationship from the computation, it only lowers its weight. That is, if the young researcher does not have other connections within the network, he will be assigned to the same community as the senior researcher. If the young researcher has already published work with his young colleagues, he would probably be assigned to a common community with them, and the senior researcher would probably be assigned to other community, e.g. with his collaborators that are also senior researchers. The two communities might even be merged on a higher level of hierarchy in a hierarchical clustering approach.

6. CONCLUSION

In this work, we have studied the behaviour of community detection algorithms with near-linear time complexity on real-world networks. We have observed that for several networks the studied algorithms fail to approximate the ground-truth communities well. We have studied the assortativity of the networks and their ground-truth communities. We have shown that the assortativity of the community structure is independent of the overall network assortativity. Two networks for which the community detection algorithms failed to deliver good partitioning have assortative community structure. We have proposed weighting functions designed to decrease the disassortativity of the connected nodes and we have studied the effect of such weighting on the networks. The empirical observation is that for the class of networks with assortative community structure, the weighting of the edges can result in significant improvements in the similarity of detected cluster. In several cases, improvements in the similarity score of an order of magnitude have been observed.

Acknowledgements

We would like to thank the anonymous reviewers for their very helpful comments that have significantly improved this paper. This work is supported by projects:
CLAN APVV-0809-11, VEGA 2/0185/13, ITMS: 26240220072 and VENIS FP7-284984.

7. REFERENCES

[1] B. Abrahao, S. Soundarajan, J. Hopcroft, and R. Kleinberg. On the separability of structural classes of communities. In *Proc. of KDD'2012*, pages 624–632, 2012.

[2] J. W. Berry, B. Hendrickson, R. A. LaViolette, and C. A. Phillips. Tolerating the community detection resolution limit with edge weighting. *Phys. Rev. E*, 83:056119, May 2011.

[3] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10), 2008.

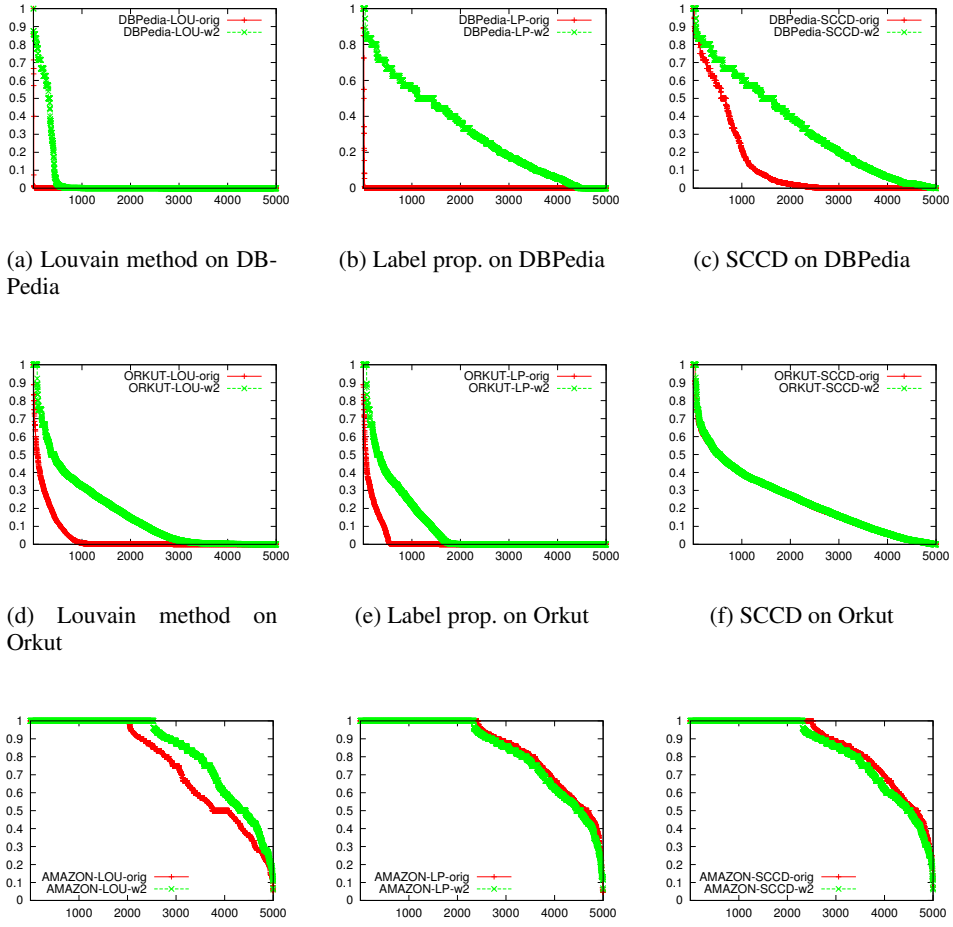
[4] M. Ciglan and K. Nørvåg. Fast detection of size-constrained communities in large networks. In *Proc. of WISE'2010*, pages 91–104, 2010.

[5] M. Ciglan, K. Nørvåg, and L. Hluchý. The SemSets model for ad-hoc semantic list search. In *Proc. of WWW'2012*, pages 131–140, 2012.

[6] L. Danon, J. Duch, A. Diaz-Guilera, and A. Arenas. Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(9), 2005.

[7] R. I. M. Dunbar. Coevolution of neocortical size, group size and language in humans. *Behavioral and Brain Sciences*, 16:681–694, December 1993.

[8] S. Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75 – 174, 2010.



(a) Louvain method on DBPedia

(b) Label prop. on DBPedia

(c) SCCD on DBPedia

(d) Louvain method on Orkut

(e) Label prop. on Orkut

(f) SCCD on Orkut

(g) Louvain method on Amazon

(h) Label prop. on Amazon

(i) SCCD on Amazon

Figure 1: Similarity of the community detection algorithms results with Top 5000 ground-truth communities. The x-axis is the rank of the ground-truth community, y-axis depicts the Jaccard similarity coefficient with the best fitting cluster.

[9] S. Gregory. A fast algorithm to find overlapping communities in networks. In *Proc. of PKDD'2008*, pages 408–423, 2008.

[10] A. Khadivi, A. Ajdari Rad, and M. Hasler. Network community-detection enhancement by proper weighting. *Phys. Rev. E*, 83:046104, Apr 2011.

[11] A. Lancichinetti and S. Fortunato. Benchmarks for testing comm. detection algorithms on directed and weighted graphs with overlapping communities. *Phys. Rev. E*, 80(1), 2009.

[12] A. Lancichinetti and S. Fortunato. Community detection algorithms: A comparative analysis. *Phys. Rev. E*, 80(5):056117, Nov 2009.

[13] A. Lancichinetti, S. Fortunato, and J. Kertész. Detecting the overlapping and hierarchical comm. structure in complex networks. *New Journal of Physics*, 11(3):033015, 2009.

[14] J. Leskovec, K. J. Lang, and M. Mahoney. Empirical comparison of algorithms for network community detection. In *Proc. of WWW'2010*, pages 631–640, 2010.

[15] D. Milne and I. H. Witten. Learning to link with Wikipedia. In *Proc. of CIKM'08*, pages 509–518, 2008.

[16] M. E. J. Newman. Assortative mixing in networks. *Physical Review Letters*, 89(20):208701, 2002.

[17] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E*, 69(2):026113, Feb 2004.

[18] S. Papadimitriou, J. Sun, C. Faloutsos, and P. S. Yu. Hierarchical, parameter-free community discovery. In *Proc. of PKDD'2008*, pages 170–187, 2008.

[19] U. N. Raghavan, R. Albert, and S. Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Phys. Rev. E*, 76(3):036106, Sep 2007.

[20] J. Yang and J. Leskovec. Defining and evaluating network communities based on ground-truth. In *Proc. of ICDM'2012*, pages 745–754, 2012.

[21] T. Yang, R. Jin, Y. Chi, and S. Zhu. Combining link and content for community detection: a discriminative approach. In *Proc. of KDD'2009*, pages 927–936, 2009.

[22] Y. Zhang, J. Wang, Y. Wang, and L. Zhou. Parallel community detection on large networks with propinquity dynamics. In *Proc. of KDD'2009*, pages 997–1006, 2009.