# Addressing the Resolution Limit and the Field of View Limit in Community Mining

Shiva Zamani Gharaghooshi[1], Osmar R. Zaïane[1], Christine Largeron[2], Mohammadmahdi Zafarmand[1], and Chang Liu[1]

[1] Alberta Machine Intelligence Institute, University of Alberta, Edmonton, Alberta, Canada {zamanigh,zaiane,zafarman,chang6}@ualberta.ca
[2] Laboratoire Hubert Curien, Université de Lyon, Saint- Etienne, France
Christine.Largeron@univ-st-etienne.fr

**Abstract.** We introduce a novel efficient approach for community detection based on a formal definition of the notion of community. We name the links that run between communities weak links and links being inside communities strong links. We put forward a new objective function, called SIWO (Strong Inside, Weak Outside) which encourages adding strong links to the communities while avoiding weak links. This process allows us to effectively discover communities in social networks without the resolution and field of view limit problems some popular approaches suffer from. The time complexity of this new method is linear in the number of edges. We demonstrate the effectiveness of our approach on various real and artificial datasets with large and small communities.

**Keywords:** Community Detection, Social Network Analysis

## 1  Introduction

Community detection is an important task in social network analysis and can be used in different domains where entities and their relations are presented as graphs. It allows us to find linked nodes that we call communities inside graphs. There are community detection methods that partition the graph into subgroups of nodes such as the spectral bisection method [4] or the Kernighan-Lin algorithm [27]. There are also hierarchical methods such as the divisive algorithms based on edge betweenness of Girwan et al. [18] or agglomerative algorithms based on dynamical process such as Walktrap [20], Infomap [24] or Label propagation [22]. We do not detail them and refer the interested reader to [7, 10, 12], but we come back on another class of hierarchical algorithms that aim at maximizing Q-modularity introduced by Newman et al. [18]. After the greedy agglomerative algorithm initially introduced by Newman [19], Blondel et al. [5] proposed Louvain, one of the fastest algorithms to optimize Q-modularity and to solve the community detection task. However, Fortunato et al. [11] showed that Q-Modularity suffers from the resolution limit which means by optimizing Q-modularity, communities that are smaller than a scale cannot be resolved.

The field of view limit [25] is in contrast to the resolution limit leads to overpartitioning the communities with a large diameter.

To overcome the resolution limit of Q-modularity, several proposals have been made, notably by [2, 17, 23], who introduced variants of this criterion allowing the detection of community structures at different levels of granularity. However, these revised criteria make the method time-consuming since they require to tune a parameter. Therefore, we retain the greedy approach of Louvain for its efficiency and ability to handle very large networks, but we introduce SIWO because it relies on the notions of strong and weak links defined in Section 2.

We consider that a community corresponds to a subgraph sparsely connected to the rest of the graph. Contrary to the majority of methods which do not formally define what is a community and simply consider that it corresponds to a subset of nodes densely connected internally, we define the conditions a subgraph should meet to be considered as a community in Section 2. In Section 3, we present the generic community detection algorithm. We can apply this general process regardless of the objective function to improve other community detection methods as our experiments show.

Finally, the extensive experiments described in Sections 4 and 5, confirm that our objective function is less sensitive to the resolution and the field of view limit compared to the objective functions mentioned earlier. Also, our algorithm has consistently good performance regardless of the size of communities in a network and is efficient on large size networks having up to a million edges.

## 2   Notations and definitions

### 2.1   Strong and weak links

A community is oftentimes defined as a subgraph in which nodes are densely connected while sparsely connected to the rest of the graph. One way to find such subgraphs is to divide the network into parts so that the number of links lying inside that part is maximized. However, if there is no prior information about the number of communities or their sizes, one can maximize the number of links within communities by putting all the nodes in one community, but the final result will not be the true communities. To avoid this approach, we penalize the missing links within the communities and we introduce the notions of strong and weak links.
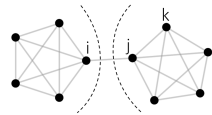


**Fig. 1.** A network with two communities; each consists of a clique of size 5.



**Fig. 2.** A network with 2 communities and 4 dangling nodes (1, 2, 3, and 4).

Weak links lie between communities, while strong links are inside them. We develop our criterion so that it encourages adding strong links to the communities while avoiding weak ones instead of penalizing the missing links. As these different types of links play different roles in graph connectivity; removing a weak link may divide the graph into disconnected subgraphs, whereas removing a random link would not. Let us focus on the link between nodes $i$ and $j$ in Fig. 1 and also the link between nodes $j$ and $k$ in this graph. Node $j$ is connected to all the neighbors of node $k$, whereas node $i$ and $j$ have no common neighbors. As generally, nodes in the same community are more likely to have common neighbors, (i,j) can be considered as a weak link whereas (j,k) as a strong link and it is exactly what we want to capture through weights assigned to the links.

## 2.2  Edge Strength

Given a graph $G = (V, E)$ where $V$ is the set of nodes and $E$ the set of edges, we propose to assign a weight in the range of $(-1, 1)$ to each edge; such that strong links have larger weights. As nodes in the same community tend to have more common neighbors compared to nodes in different communities, if $S_{xy} > S_{xy'}$ then $e_{xy}$ is more likely to be a strong link compared to $e_{xy'}$ with $S_{xy}$ defined by:

$$S_{xy} = |\{k \in V : (x, k) \in E, (y, k) \in E\}| \tag{1}$$

We can compare two links according to $S$ only if they share a node. Thus, if we consider nodes $x$ and $y$ that have 5 and 20 links incident to them, then $S$ can be in range of $[0, 4]$ and $[0, 19]$ for $x$ and $y$ respectively. Consequently, for comparisons, we have to scale down $S$ values to $(-1, 1)$. If $S_{xy}$ has the maximum value of $S_x^{max}$ ($S_x^{max} = \max_{y:(x,y) \in E} S_{xy}$) for a particular node $x$. We divide the range $[-1, 1]$ into $S_x^{max} + 1$ equal length segments. Each $S$ value in the range of $[0, S_x^{max}]$ is then mapped to the center of $(n + 1)^{th}$ segment using equation:

$$w_{xy}^x = S_{xy} \frac{2}{S_x^{max} + 1} + \frac{1}{S_x^{max} + 1} - 1 \tag{2}$$

where $w_{xy}^x$ is the scaled value of $S_{xy}$ from the viewpoint of node $x$ (min-max normalization could also work). We can also scale $S_{xy}$ from the viewpoint of node $y$: $w_{xy}^y = S_{xy} \frac{2}{S_y^{max} + 1} + \frac{1}{S_y^{max} + 1} - 1$ where $S_y^{max} = \max_{x:(y,x) \in E} S_{xy}$. To decide whether we should trust $x$ or $y$, we need to look at the importance of each one in the network. Local clustering coefficient ($CC$) [28], given below, is a measure that reflects the importance of nodes and it can be computed even on large graphs, for instance with Mapreduce [15].

$$CC(x) = \frac{|\{e_{ij} : i, j \in N_x, e_{ij} \in E\}|}{\binom{d_x}{2}} \tag{3}$$

where $d_x$ and $N_x$ are respectively the degree and the set of neighbors of node $x$. $CC$ is in the range of [0,1] with 1 for nodes whose neighbors form cliques, and 0 for nodes whose neighbors are not connected to each other directly. Here, we

scale each edge from the viewpoint of the endpoint that is more likely to be in a dense neighborhood characterized by a large $CC$:

$$w_{xy} = \begin{cases} w_{xy}^x, & \text{if } CC(x) \geq CC(y) \\ w_{xy}^y, & \text{otherwise} \end{cases} \qquad (4)$$

### 2.3  SIWO Measure

The new measure that we propose encourages adding strong links into the communities while keeping the weak links outside of the communities (**S**trong **I**nside, **W**eak **O**utside). This measure is defined as follows:

$$SIWO = \sum_{i,j \in V} \frac{w_{ij}\delta(c_i, c_j)}{2} \qquad (5)$$

where $c_i$ is the community of node $i$ and $\delta(x,y)$ is 1 if $x = y$ and 0 otherwise. SIWO is the sum of weights of the edges that reside in the communities. This objective function provides a way to partition the set of nodes but it does not specify the conditions required by a subset of nodes to be a community. These conditions are defined in the following.

### 2.4  Community Definition

Following [21] we consider that a subgraph $C$ is a community in a weak sense if the following condition is satisfied:

$$\frac{1}{2}\sum_{v \in C} |N_v^C| > \sum_{v \in C} |N_v - N_v^C| \qquad (6)$$

where $N_v$ is the set of the neighbors of node $v$ and $N_v^C$ is the set of the neighbors of node $v$ that are also in community $C$. This condition means that the collective of the nodes in a community have more neighbors within the community than outside. In this paper, we expand this definition by adding one more condition. Given a partition $p = \{C_1, C_2, ..., C_t\}$ of a network, subgraph $C_i$ is considered as a **qualified community** if it satisfies the following conditions:

1. $C_i$ is a community in a weak sense (Eq. 6).
2. The number of links within $C_i$ exceeds the number of links towards any other subgraph $C_j$ ($j \neq i$) in the partition $p$ taken separately, such that:

$$\frac{1}{2}\sum_{v \in C_i} |N_v^{C_i}| > \sum_{v \in C_i} |N_v^{C_j}|, j \in [1..t], j \neq i \qquad (7)$$

## 3   The SIWO Method

This method has four steps: pre-processing, optimizing SIWO, qualified community identification, and post-processing. They are discussed in detail below.

**Step 1. Pre-processing**

The first step calculates the edge strength weights $(w_{ij})$ needed during the SIWO optimization. Moreover, to reduce the computational time, we remove the dangling nodes temporally. Node $x$ is a dangling node if there exists node $y$ such that by removing $e_{xy}$, the network would be divided into two disconnected parts with $part_x$ (the part containing node $x$) being a tree. Since $part_x$ has a tree structure, it cannot form a community on its own. So all the nodes in $part_x$ belong to the same community as node $y$. In Fig. 2, nodes 1, 2, 3 and 4 are dangling nodes and they belong to the same community as node 5, unless we consider them outliers. Even though such tree-structured subgraphs attached to the network are very sparse and cannot be considered as communities, they satisfy Eq. (6) and (7) defined for qualified communities. So we do not need to consider them during the community detection process. To remove them (and the links incident to them), we need to investigate every node of the network in the first time to identify nodes with degree of 1. However, after the first visit, we only need to check the list of the neighbors of the nodes that are removed in the previous time.

**Step 2. Optimizing SIWO**

We use Louvain's optimization process to maximize SIWO since it has been proven to be very efficient but we replace the modularity by our criterion. This greedy optimization process has two main phases, iteratively performed until a local maximum of the objective function (SIWO measure) is reached. The first phase starts by placing each node of graph $G$ in its community. Then each node is moved to the neighbor community which results in the maximum gain of the SIWO value. If no gain can be achieved, the node stays in its community. In the second phase, a new weighted graph $G'$ is created in which each node corresponds to a community in $G$. Two nodes in $G'$ are connected if there exists at least one edge lying between their corresponding communities in $G$. Finally, we assign each edge $e_{xy}$ in $G'$ a weight equal to the sum of the weights of edges between the communities that match with $x$ and $y$. These two phases are repeated until no further improvement in the SIWO objective function can be achieved.

**Step 3. Qualified community identification**

This step determines qualified communities complying with Eq. (6) and (7) for the dense subgraphs discovered in the previous step. However, there may exist communities consisting of one node weakly connected to all of its neighbors $(S_x^{max} = 0)$ and that have links with non-positive weight incident to it, we call them Lone communities. Since the decision about the communities of such nodes can not be made on edge strength, we let the majority of their neighbors decide about their communities but, to reduce the computational time, like for dangling nodes, we temporarily remove these nodes in this step and bring them back in the final step. Then, we identify the unqualified communities which do not satisfy Eq. (6) or (7). We keep merging each unqualified community with

one of its neighboring communities (qualified or not) until no more unqualified community exists. For that, first, we assign a weight equal to 1 to each edge. Then, we repeat the two phases of Louvain. In phase 1, we create a new graph $G^*$ in which each node corresponds to a community identified in step 3 for the first iteration of in phase 2 for the next ones and where each edge $e_{xy}$ is assigned a weight equal to the sum of the weights of edges between the communities that correspond to x and y. We also add a self-loop to each node that has a weight equal to the sum of the weights of the edges that reside in its corresponding community. In phase 2, we visit all nodes in $G^*$. If a node $x$ has a self-loop with a weight that is larger than 1) half of sum of the weights of the edges incident to it and 2) weight of any edge connecting $x$ to another node in $G^*$, it means the community assigned to $x$ satisfies both the conditions in Eq. (6) and (7), we let $x$ stay in its community. Otherwise, we move node $x$ to the neighboring community that results in the maximum decrease in the sum of the weights of the edges that lie between communities of $G^*$.

**Step 4. Post-processing**

Finally, each lone community that was temporarily removed is sequentially added back to the network and merged with the community in which it has the most neighbors. If two or more communities tie and they have more than one connection to the node, then one is chosen at random. Otherwise, we choose the community of the most important neighbor, based on the largest degree of centrality within its community. Since we add lone nodes one after the other, the community that a former node is assigned to, might not be the best for that node. To resolve this issue, once all lone nodes are added to the network, we repeat moving each one of them to the community of the majority of its neighbors until no further movement can be made. Dangling nodes are also added to the network in the reverse order that they were removed and they are assigned to the community of their unique neighbor.

## 4     The resolution limit of SIWO

Fortunato and Barthélemy [11] used two sample networks, shown in Fig. 3, to demonstrate how Q-modularity is affected by the resolution limit. The first example is a ring of cliques where each clique is connected to its adjacent cliques through a single link. If the number of cliques is larger than about $\sqrt{m}$ with $m$ being the total number of edges in the network, then optimizing Q-modularity results in merging the adjacent cliques into groups of two or more, despite that each clique corresponds to a community. The second example is a network containing 4 cliques: 2 of size k and 2 of size p. If $k >> p$, Q-modularity similarly fails to find the correct communities and the cliques of size $p$ will be merged.

To prove how SIWO resolves the resolution limit of Q-modularity, the exact structure of the network should be known ; which is not possible. So, we analyze whether SIWO is affected by the resolution limit on these networks Given the definition of SIWO, let us consider the edge $e_{xy}$ between two adjacent cliques in the first network. Since $x$ and $y$ do not have any common neighbors, the
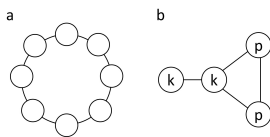
**Fig. 3.** Schematic examples a) a ring of cliques; adjacent cliques are connected through a single link b) a network with 2 cliques of size $k$ and 2 cliques of size $p$.

edge between them has a non-positive weight. Therefore, by maximizing SIWO measure in our algorithm, the adjacent cliques will not be merged. For the edge $e_{xy}$ between the cliques of size $p$ in the second network, since $x$ and $y$ have at most one common neighbor, the edge between them has a non-positive weight. Therefore, the cliques in the second network will not be merged either.

## 5    Experimental results

We compared the performance of our method with the most widely used and efficient algorithms, as pointed out in several recent state of art studies [8, 29], on both real and synthetic networks. The algorithms are: 1- Fastgreedy [6]; 2- Infomap; 3- Infomap+ which is Infomap to which we added the third step of our algorithm (to relieve its sensitivity to the field of view limit and demonstrate that our framework can be used to improve other algorithms); 4- Label Propagation [22]; 5- Louvain[3] [5]; 6- Walktrap[4] [20]. It should be noted that Infomap is the only algorithm that suffers from the filed of view limit among these algorithms.

The results are evaluated according to the Adjusted Rand Index (ARI) [14] and Normalized Mutual Information (NMI) [26]. As both ARI and NMI show similar results, we only present ARI results for lack of space. We also compared the results of different methods according to the ratio of the number of detected communities over the true number of communities in the ground-truth to observe how a method is affected by the resolution and the field of view limits.

### 5.1    Real networks

We used 5 real networks and the ground-truth communities are available for 4 of them. Table 1 presents the properties of these networks.

We compared SIWO and Louvain on Eurosis network [9] which represents scientific web pages from 12 European countries and the hyperlinks between them without known ground-truth communities. However, since each European country has its own language, web pages in different countries are sparsely connected to each other. Moreover, as reported in [9], some of the countries can be

---

[3] `https://github.com/taynaud/python-louvain`

[4] `https://www-complexnetworks.lip6.fr/~latapy/PP/walktrap.html`

[5] `http://www.orgnet.com`

**Table 1.** Properties of real networks

| Network | #nodes | #edges | #C | Network | #nodes | #edges | #C |
|---|---|---|---|---|---|---|---|
| Karate [30] | 34 | 78 | 2 | Eurosis [9] | 1218 | 5999 | - |
| Polbooks [5] | 105 | 441 | 3 | Polblogs [1] | 1222 | 16717 | 2 |
| Football [13] | 115 | 613 | 12 | | | | |

divided into smaller components e.g. Montenegro network includes three components: 1- Telecom and Engineering, 2- Faculties and 3- High Schools. Louvain detects 13 communities whereas SIWO detects 16 communities in this network. Louvain assigns all nodes in Montenegro network to one giant community. However, SIWO puts Faculties and High Schools in one community and Telecom and Engineering web pages in another community. These two communities are connected to each other with only 7 links. However, Louvain cannot separate them due to its resolution limit.

**Table 2.** Comparison of 7 algorithms according to ARI and the ratio of the number of detected communities over the true number of communities in the ground-truth on real networks. Tables shows the average results and standard deviation computed on 10 iterations of the algorithms on each network.

| | | Karate | Polbooks | Football | Polblogs |
|---|---|---|---|---|---|
| SIWO | ARI | $\mathbf{1 \pm 0}$ | $\mathbf{0.67 \pm 0}$ | $0.79 \pm 0$ | $0.77 \pm 0$ |
| | $\overline{C}/C_r$ | $\mathbf{1 \pm 0}$ | $1.3 \pm 0$ | $\mathbf{1 \pm 0}$ | $\mathbf{1.5 \pm 0}$ |
| Fastgreedy | ARI | $0.68 \pm 0$ | $0.63 \pm 0$ | $0.47 \pm 0$ | $0.78 \pm 0$ |
| | $\overline{C}/C_r$ | $1.5 \pm 0$ | $1.3 \pm 0$ | $0.5 \pm 0$ | $5 \pm 0$ |
| Infomap | ARI | $0.7 \pm 0$ | $0.64 \pm 0$ | $0.84 \pm 0$ | $0.68 \pm 0$ |
| | $\overline{C}/C_r$ | $1.5 \pm 0$ | $1.6 \pm 0$ | $0.9 \pm 0$ | $17.5 \pm 0$ |
| Infomap+ | ARI | $0.70 \pm 0$ | $0.66 \pm 0$ | $\mathbf{0.84 \pm 0}$ | $0.76 \pm 0$ |
| | $\overline{C}/C_r$ | $1.5 \pm 0$ | $1.3 \pm 0$ | $0.9 \pm 0$ | $1.5 \pm 0$ |
| Label_prop | ARI | $0.66 \pm 0.3$ | $0.66 \pm 0$ | $0.73 \pm 0$ | $\mathbf{0.8 \pm 0}$ |
| | $\overline{C}/C_r$ | $1.2 \pm 0.35$ | $\mathbf{1.1 \pm 0.1}$ | $0.8 \pm 0.1$ | $2.1 \pm 0$ |
| Louvain | ARI | $0.46 \pm 0$ | $0.55 \pm 0$ | $0.8 \pm 0$ | $0.77 \pm 0$ |
| | $\overline{C}/C_r$ | $2 \pm 0$ | $1.3 \pm 0$ | $0.8 \pm 0$ | $4.5 \pm 0$ |
| Walktrap | ARI | $0.32 \pm 0$ | $0.65 \pm 0$ | $0.81 \pm 0$ | $0.76 \pm 0$ |
| | $\overline{C}/C_r$ | $3 \pm 0$ | $1.3 \pm 0$ | $0.8 \pm 0$ | $5.5 \pm 0$ |

Table 2 presents the comparison with respect to ARI and $\overline{C}/C_r$, the ratio of the number of detected communities over the true number of communities (both ARI and $\overline{C}/C_r$ should be as close to 1 as possible) in the ground-truth, on real networks with ground-truth communities. It shows that SIWO performs better on Karate and Polbooks based on ARI. It also outperforms the others methods on Karate, Football, and Polblogs networks according to $\overline{C}/C_r$ measure (SIWO could detect the exact communities with respect to the ground-truth on these networks). Infomap detects a considerably larger number of communities in Polblogs network which indicates this algorithm is sensitive to the field of view

limit [25]. However, Infomap+ is much less sensitive to this limit which implies the third step of SIWO, added to Infomap+, is effective in resolving the field of view limit. Considering results for all networks, SIWO is the top performer among these algorithms on a variety of networks.

## 5.2   Synthetic networks

To analyze the effect of the resolution and field of view limit, it is important to test how community detection algorithms perform on networks with small/large communities. Therefore, in this work we generated two sets of networks using LFR [16] to test the different algorithms: one with large communities and one with small communities. The first set is in favor of algorithms that suffer from resolution limit such as Louvain and the second set is in favor of algorithms with field of view limit such as Infomap. Each set includes networks with a varying number of nodes and mixing parameter. The mixing parameter controls the fraction of edges that lie between communities. We do not generate networks with mixing parameter $\geq 0.5$ since beyond this point and including 0.5, the communities in the ground truth no longer satisfy the definition of community. The input parameters used to generate these two sets are presented in Table 3. Fig. 4 and 5 present respectively ARI or the ratio of the number of detected communities over the true number of communities ($\overline{C}/C_r$). Panels correspond to networks with a specific number of nodes (1000 to 100000) and they are divided into two parts; the lower (respectively upper) part illustrates the average ARI (or $\overline{C}/C_r$) (respectively standard deviation) computed over 20 graphs (10 small and 10 large communities) as a function of the mixing parameter.

**Table 3.** Input parameters of LFR benchmark: Set 1 contains networks with large communities and Set 2 contains networks with small communities. For each combination of parameters we generated 10 networks.

|  | Set 1 | Set 2 |
|---|---|---|
| #nodes ($N$) | $[1, 10, 50, 100] \times 10^3$ | $[1, 10, 50, 100] \times 10^3$ |
| average and max degrees | 20 - $N/10$ | 20 - $\sqrt{N}$ |
| mixing parameter | $[1, \ldots, 7] \times 0.1$ | $[1, \ldots, 7] \times 0.1$ |
| min and max community sizes | $N/20$ - $N/10$ | default - by default $\sqrt{N}$ |

Fig. 4 shows the performance of Fastgreedy decreases as the mixing parameter increases. Louvain and Walktrap perform well on the smallest networks in the set; however, its performance drops when we apply it to the networks with sizes 50000 and larger. Label propagation, Infomap and Infomap+ perform well up to when the mixing parameter reaches 0.3. However, a larger mixing parameter causes a rapid decrease in the ARI value when applying these algorithms to the two largest networks in the set. These three algorithms have a large standard deviation and their outputs are not stable on these networks. SIWO correctly detects the communities when the mixing parameter is less than or equal to 0.3 (ARI $\simeq 1$) regardless of size of the network and has the best performance overall.
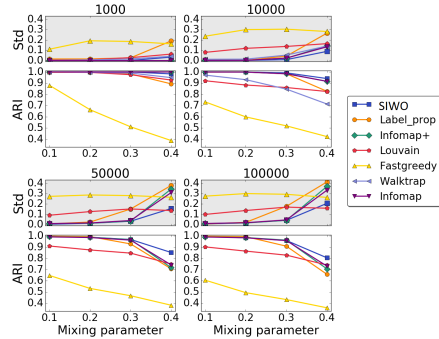
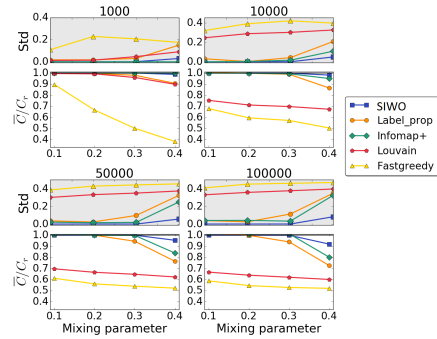**Fig. 4.** Evaluation according to ARI on synthetic networks generated with LFR.

**Fig. 5.** Evaluation of SIWO, Label propagation, Infomap+, Louvain and Fastgreedy according to $\overline{C}/C_r$ on synthetic networks generated with LFR.

Fig. 5 clearly shows the resolution limit of Louvain and Fastgreedy as they underestimate the number of communities. SIWO is the best performer in terms of the number of communities and it has a very small standard deviation whereas, Infomap+ and Label propagation have a large standard deviation and fail to find the correct number of communities when the mixing parameter exceeds 0.3.

## 6  Scalability

We analyze how the computational cost of SIWO varies with the size of the network. The pre-processing step has two phases: removing dangling nodes which requires a time of the order of $n$ where $n$ is the number of nodes, and calculating the edge strength weights which requires a time of the order of $nd^2 = 2md$ where $m$ is the number of edges and $d$ is the average degree. In many real networks $d$ is much smaller than $n$ and it does not grow with $n$ [10]. The second and third step follows the same greedy process as Louvain does. Louvain is theoretically cubic but was demonstrated experimentally to be quasi-linear [3] and has been applied with success to handle large size networks having several million nodes, and 100 million links. The time complexity of the post-processing step depends on the number of Lone communities and if all the nodes are in Lone communities, it requires a time $O(nd^2)$. Overall, the time complexity of SIWO is $O(n + md)$, which is similar to Louvain due to the fact that $d$ is small and $n = 2m/d$. SIWO can detect communities in a networks with 100000 nodes and 1 million edges, in about 1 minute on a commodity i7 and 8GB RAM laptop. The current implementation of SIWO is in Python[6], derived from python-louvain.

---

[6] SIWO Code and datasets available at https://www.dropbox.com/sh/eehjt5qblll0yvg/AACW2XjHJjHX2Q876Vbk0e4Ya?dl=0

## 7 Conclusion

This paper introduces SIWO, a novel objective function based on edge strength for community detection, and a formal definition of community, that we use to lead the community detection process after optimizing the objective function. This framework can also be applied to other community detection methods to remedy their inability that causes the resolution or the field of view limit. Our extensive experiments using both small and large networks confirm that our algorithm is consistent, effective and scalable for networks with either large or small communities demonstrating less sensitivity to the resolution limit and field of view limit that most community mining algorithms suffer from. As a future direction, we will generalize the proposed algorithm for weighted/directed networks. Notably, SIWO algorithm can be easily generalized to handle weighted graphs. It requires only to adjust the pre-processing step by combining the weights from the input graph and the weights computed by SIWO to evaluate the edge strength.

# Bibliography

[1] Adamic, L.A., Glance, N.: The political blogosphere and the 2004 U.S. election. In: Proc. of the 3rd Inter. WS on Link Discovery. pp. 36–43 (2005)

[2] Arenas, A., Fernandez, A., Gomez, S.: Analysis of the structure of complex networks at different resolution levels. New J. Phys. 10(5), 053039 (2008)

[3] Aynaud, T., Blondel, V.D., Guillaume, J.L., Lambiotte, R.: Multilevel Local Optimization of Modularity, pp. 315–345. John Wiley & Sons, Ltd (2013)

[4] Barnes, E.R.: An algorithm for partitioning the nodes of a graph. SIAM J. Alg. Discr. Meth.s 3(4), 541–550 (1982)

[5] Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment 2008(10), P10008 (2008)

[6] Clauset, A., Newman, M.E.J., Moore, C.: Finding community structure in very large networks. Phys. Rev. E 70, 066111 (2004)

[7] Coscia, M., Giannotti, F., Pedreschi, D.: A classification for community discovery methods in complex networks. Stat. Anal. Data Min. 4(5), 512–546 (2013)

[8] Emmons, S., Kobourov, S., Gallant, M., Börner, K.: Analysis of network clustering algorithms and cluster quality metrics at scale. PLOS ONE 11(7), 1–18 (2016)

[9] EUROSIS Final Report: Webmapping of science and society actors in europe, final report (2018-06-01), www.eurosfaire.prd.fr/7pc/documents/1274371553\_finalreporteurosis3\_1.doc

[10] Fortunato, S.: Community detection in graphs. Phy. Rep. 486(3-5), 75–174 (2010)

[11] Fortunato, S., Barthélemy, M.: Resolution limit in community detection. PNAS 104(1), 36–41 (2007)

[12] Fortunato, S., Hric, D.: Community detection in networks: A user guide. Phys. Rep. 659, 1–44 (2016)

[13] Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. PNAS 99(12), 7821–7826 (2002)

[14] Hubert, L., Arabie, P.: Comparing partitions. Journal of Classification 2(1), 193–218 (Dec 1985)

[15] Kolda, T.G., Pinar, A., Plantenga, T.D., Seshadhri, C., Task, C.: Counting triangles in massive graphs with mapreduce. SIAM J. Scientific Computing 36(5) (2014)

[16] Lancichinetti, A., Fortunato, S., Radicchi, F.: Benchmark graphs for testing community detection algorithms. Phys. Rev. E 78(4), 1–5 (2008)

[17] Li, Z., Zhang, S., Wang, R.S., Zhang, X.S., Chen, L.: Quantitative function for community detection. Phys. Rev. E 77(3), 36109 (2008)

[18] Newman, M., Girvan, M.: Finding and evaluating community structure in networks. Phys. Rev. E 69, 026113 (2004)

[19] Newman, M.: Fast algorithm for detecting community structure in networks. Phys. Rev. E - Stat. Nonlin. Soft Matter Phys. 69, 066133 (2004)

[20] Pons, P., Latapy, M.: Computing Communities in Large Networks Using Random Walks. In: Computer and Information Sciences. pp. 284–293 (2005)

[21] Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., Parisi, D.: Defining and identifying communities in networks. PNAS 101(9), 2658–63 (3 2004)

[22] Raghavan, N., Albert, R., Kumara, S.: Near linear time algorithm to detect community structures in large-scale networks. Phys. Rev. E - Stat. Nonlin. Soft Matter Phys. 76, 036106 (2007)

[23] Reichardt, J., Bornholdt, S.: Statistical mechanics of community detection. Phys. Rev. E 74, 16110 (2006)

[24] Rosvall, M., Bergstrom, C.: Maps of Random Walks on Complex Network Reveal Community Structure. PNAS 105(4), 1118–1123 (2008)

[25] Schaub, M.T., Delvenne, J.C., Yaliraki, S.N., Barahona, M.: Markov dynamics as a zooming lens for multiscale community detection: Non clique-like communities and the field-of-view limit. PLOS ONE 7, e32210 (2012)

[26] Strehl, A., Ghosh, J.: Cluster ensembles — a knowledge reuse framework for combining multiple partitions. J. Mach. Learn. Res. 3, 583–617 (2003)

[27] W., K.B., S., L.: An efficient heuristic procedure for partitioning graphs. Bell System Technical Journal 49(2), 291–307 (1970)

[28] Watts, D.J., Strogatz, S.H.: Collective dynamics of 'small-world' networks. Nature 393(6684), 440–442 (1998)

[29] Yang, Z., Algesheimer, R., Tessone, C.J.: A comparative analysis of community detection algorithms on artificial networks. Sci. Rep. 6(30750) (2016)

[30] Zachary, W.: An information flow model for conflict and fission in small groups. Journal of Anthropological Research 33, 452–473 (1977)