GraphReform^{CD}: Graph Reformulation for Effective Community Detection in Real-World Graphs

Jiwon Hong
Hanyang University
Seoul, Republic of Korea
nowiz@hanyang.ac.kr

Dong-hyuk Seo
Hanyang University
Seoul, Republic of Korea
hyuk125@hanyang.ac.kr

Jeewon Ahn
Hanyang University
Seoul, Republic of Korea
dkswldnjs@hanyang.ac.kr

Sang-Wook Kim
Hanyang University
Seoul, Republic of Korea
wook@hanyang.ac.kr

ABSTRACT

Community detection, one of the most important tools for graph analysis, finds groups of strongly connected nodes in a graph. However, community detection may suffer from *misleading information* in a graph, such as a nontrivial number of inter-community edges or an insufficient number of intra-community edges. In this paper, we propose GraphReform^{CD} that reformulates a given graph into a new graph in such a way that community detection can be conducted more accurately. For the reformulation, it builds a k-nearest neighbor graph that gives a node k opportunities to connect itself to those nodes that are likely to belong to the same community together with the node. To find the nodes that belong to the same community, it employs the structural similarities such as Jaccard index and SimRank. To validate the effectiveness of our GraphReform^{CD}, we perform extensive experiments with six real-world and four synthetic graphs. The results show that our GraphReform^{CD} enables state-of-the-art methods to improve their accuracy significantly up to 40.6% in community detection.

CCS CONCEPTS

• **Theory of computation → Unsupervised learning and clustering**; *Social networks*; • **Information systems → Clustering**.

KEYWORDS

social networks, community detection, clustering, nearest neighbor graph, graph reformulation

ACM Reference Format:
Jiwon Hong, Dong-hyuk Seo, Jeewon Ahn, and Sang-Wook Kim. 2022. GraphReform^{CD}: Graph Reformulation for Effective Community Detection in Real-World Graphs. In *Companion Proceedings of the Web Conference 2022 (WWW '22 Companion), April 25–29, 2022, Virtual Event, Lyon, France.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3487553.3524240

1 INTRODUCTION

Real-world graphs, including social networks, have communities inside them. *Communities* are subsets of nodes, with the connections within a subset (i.e., *intra-community edges*) being dense and the connections between different subsets (i.e., *inter-community edges*) being fairly sparse. *Community detection* (CD) is an operation of detecting such a community structure in a given graph [1, 10]; it is one of core graph analysis techniques and has various applications.

CD has been studied for a long time. There are some categories of CD algorithms found in the literature: vertex clustering (e.g., SHRINK, BlackHole [7]), substructure detection (e.g., SCAN), divisive ones (e.g., [8]), quality optimization (e.g., Louvain [1]), and model-based ones (e.g., Infomap [10]). Below are the existing CD algorithms widely used in the literature. SHRINK uses agglomerative hierarchical clustering [4]. BlackHole [7] embeds a graph into a set of points on a low-dimensional space and uses spectral clustering to detect communities. SCAN is a variant of DBSCAN [4], a well-known density-based clustering algorithm, for CD. Louvain [1] is a well-known modularity maximization algorithm that employs agglomerative hierarchical clustering. Infomap [10] finds communities based on information theory.

In general, an edge between two nodes in a graph implies a relationship between them. *From the CD perspective, the existence of an edge between two nodes could be interpreted as a high probability that they belong to the same community* [1, 10]. A node in a real-world graph determines whether to connect to other nodes by only considering their attributes rather than taking the entire community structure in the graph into account. For example, on a social network, one in a community of computer scientists could have a friendship with someone else whom s/he met on a trip, even if they do not belong to the same CS community. In this way, a real-world graph contains some information that could *mislead* CD algorithms in the wrong way: a node may not connect itself to other nodes within its community (i.e., *absence of intra-community edges*), and it may connect itself to other nodes outside its community (i.e., *presence of inter-community edges*). As a graph contains more misleading information, it gets more difficult to detect the community structure accurately on the graph: i.e., misleading information in a graph could cause inaccurate CD results even if a good CD algorithm is employed [7].

In this paper, we propose a method of *graph reformulation*, called GraphReform^{CD} to address this problem of misleading information. Suppose each node of a graph determines whether to create an edge to another node by considering the possibility that the two nodes belong to the same community (i.e., instead of considering only their individual characteristics). In this case, the graph will have less misleading information in the perspective of CD, which is what our GraphReform^{CD} aims at. It transforms a given graph into a new one in such a way that each node connects itself to those nodes that are likely to belong to its community together. In this way, if the graph has been reformulated to include less misleading information, the communities detected with this new graph should

WWW '22 Companion, April 25–29, 2022, Virtual Event, Lyon, France

ACM ISBN 978-1-4503-9130-6/22/04…$15.00
https://doi.org/10.1145/3487553.3524240

be more accurate than those with the original one. In order to show the effectiveness of GraphReform^{CD}, we perform extensive experiments with five popular CD algorithms on synthetic and real-world graphs: we apply existing state-of-the-art algorithms to both the original and reformulated graphs and measure their accuracies. The results show that the reformulated graphs improve the accuracies of CD significantly by up to 40.6%.

2 GRAPHREFORM^{CD}

GraphReform^{CD} first removes *all* edges from the original graph and then creates new edges that help achieve more accurate CD. In this process, a newly created edge should connect a pair of nodes that are likely to belong to the same community. In this section, we discuss how to measure the degree of the likelihood that a pair of nodes belong to the same community and construct a reformulated graph by using the newly created edges.

2.1 Structural Similarity Measures

For the first mission of GraphReform^{CD}, we need a method to evaluate *the likelihood of two nodes belonging to the same community*. Towards this end, we use a *structural similarity measure* that computes the degree of how *close* the given two nodes are in a graph by taking into account the *topological structure* of the graph.

Structural similarity measures could be classified into three categories based on how they evaluate the closeness of a given pair of nodes in a graph:

- *1-hop-neighbor based ones*: they consider only the *direct* neighbors of a given pair of nodes. The similarity between the two nodes becomes higher as the number of common nodes directly connected to the two nodes increases. Typical examples include Jaccard index [4], Adamic/Adar index, and cosine similarity [4].
- *Multi-hop-neighbor based ones*: they take into account the *indirect* neighbors as well as the *direct* neighbors of a given pair of nodes. The similarity between a pair of nodes becomes higher as the number of common nodes directly and indirectly connected to the two nodes increases. Here, the indirect neighbors are made to impact the similarity less than the direct neighbors. Typical examples include SimRank and Random Walk with Restart (RWR) [3, 12].
- *Graph-embedding based ones*: graph embedding [3] aims to represent each node of a graph as a vector in low dimensional space where a pair of nodes are located more closely to each other as the number of nodes *directly/indirectly* connected to the two nodes increases. Graph-embedding based similarities of a pair of nodes could be computed by using the distance between their corresponding vectors in the embedding space. Typical graph embedding techniques include Deepwalk, LINE [12] and Node2Vec [3].

According to the definition of a community, there are relatively many edges connecting the nodes in the same community and relatively few edges connecting the nodes in different communities. Therefore, a pair of nodes within a community has more nodes (directly/indirectly) connecting *both* nodes than a pair of nodes from different communities. Based on these characteristics, we suppose that the higher the structural similarity between a pair of nodes, the more likely the two nodes belong to the same community.

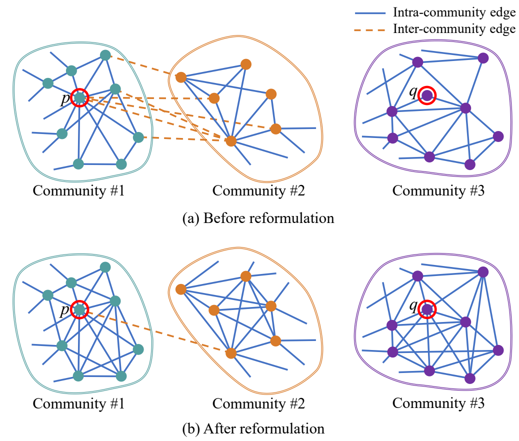

Figure 1: Example of a graph reformulation.

2.2 k-Nearest Neighbor Graph

Based on the similarities thus computed, we build the *k-Nearest Neighbor* (kNN) graph [11] as a reformulated graph. In a kNN graph, each node has the same chance to create k *outgoing edges* connected to other nodes that are most similar to it. This kNN graph is beneficial to accurate CD in two aspects below.

First, the high-degree nodes in the original graph are more likely to have inter-community edges. In a kNN graph, such nodes have a chance of connections *limited to* k, making themselves likely to create *much fewer inter-community edges* connected to those nodes in other communities. Thus, this limitation to high-degree nodes contributes to the decreased chance of making their inter-community edges.

Second, the low-degree nodes in the original graph are more likely to have only a few intra-community edges. In a kNN graph, such nodes have a chance of more (i.e., k) connections to other nodes, making themselves likely to create *more intra-community edges* connected to those nodes in the same community. Thus, this limitation to low-degree nodes contributes to the increased chance of making their intra-community edges. As a result, the reformulated graph has significantly reduced misleading information, meaning more intra-community edges and fewer inter-community edges; this would help CD algorithms identify communities in a graph more accurately.

We note, in the reformulated kNN graph, it is likely that high-degree nodes lose a lot of intra-community edges as well. In such a case, however, since they will have much more than k intra-community edges (i.e., k *outgoing* edges + more *incoming* edges) after the reformulation, the accuracy of CD would not be affected significantly.

Figure 1 demonstrates the advantage of GraphReform^{CD}. Figures 1-(a) and 1-(b) show two graphs before and after its application, respectively. Note that they show only the subgraphs centered on nodes p and q. A circle represents a node, a solid line represents an intra-community edge, and a dotted line represents an inter-community edge. In Figure 1-(a), a high-degree node p has a large number of intra-community edges but has some inter-community edges as well. In Figure 1-(b), however, p has inter-community edges significantly reduced in a 3-NN graph obtained after the reformulation. In Figure 1-(a), a low-degree node q has only two intra-community edges. However, in Figure 1-(b), in a 3-NN graph obtained after the reformulation, q has a more number of intra-community edges. We observe that there are those nodes with a degree greater than three, despite $k = 3$. Such a node gets excessive edges from other nodes that are not its 3-NNs but think of it as their 3-NNs.

In summary, GraphReform^{CD} finds the kNNs of every node in the original graph by using a structural similarity and constructs a kNN graph composed of the edges connecting the node and each of its kNNs. As a result, we obtain the kNN graph in this way, which is more beneficial to accurate CD than the original one.

3 EXPERIMENTAL SETUP FOR EVALUATION

GraphReform^{CD} has variants according to the following parameters.

- *Structural similarity measures*: We use the Jaccard index, Adamic/Adar index, Simrank, and Node2Vec [3] for calculating a score of structural similarity between a pair of nodes.
- k: We should determine the value of k to build a kNN graph. In our evaluation, we use the multiples of the average degree, D_{avg}, of each original graph (i.e., $k = D_{avg}, 2D_{avg}, \ldots, 5D_{avg}$).
- *k-Nearest Neighbor (kNN) graph and Mutual k-Nearest Neighbor (MkNN) graph*: The mutual kNN graph [11] is a variant of kNN graphs. In kNN graphs, an edge is created between two nodes when *either one of them* considers the other as its kNN. In MkNN graphs, an edge is created between two nodes only when *both of them* consider each other as their kNNs. In our evaluation, we use these two versions for GraphReform^{CD}[1].

We conduct experiments for all possible combinations of the parameters of GraphReform^{CD}by employing five state-of-the-art CD algorithms: SHRINK, SCAN, Louvain [1], Infomap [10], and BlackHole [7]. For the parameter setting of each CD algorithm, we perform the grid search with various combinations of parameter values and show the best one with the highest NMI among their results. Note that some CD algorithms have the *randomness* in their nature, producing different results for different experiments even with the same set of parameter values. We show the average NMI for *five runs* for those algorithms to address this issue.

We use six real-world datasets and four synthetic benchmark data-sets for our evaluation. The real-world datasets are Football, Polbooks, Karate, Email, Cora, and PubMed that provide ground-truth community structures [6, 9]. The synthetic datasets are generated by the LFR-benchmark. The ratio of inter-community edges to all edges in a graph determines the amount of *misleading information* (in terms of CD) contained in the graph. The LFR-benchmark generates synthetic datasets with various characteristics by controlling the ratio of inter-community edges over all edges via a mixing parameter, μ, and the number of nodes via $|V|$.

Modularity [8] is a quality function for measuring how well a graph is divided into communities. We first evaluate the modularity on the ground truth communities of each graph dataset before and after the reformulation. This is to verify that our reformulation successfully reduces the misleading information. We then evaluate the

[1]We also employed other variants of MkNN graphs [11] that allow some additional non-mutual kNN edges. However, we do not include them here since they made no meaningful difference in the accuracy.

Table 1: Inter-community edge ratios and modularities before and after reformulation.

Dataset	$\|V\|$	$\|E^{inter}/E\|$		Q	
		Orig.	Ours	Orig.	Ours
Karate	34	**0.128**	0.156	**0.371**	0.343
Football	115	0.357	**0.207**	0.554	**0.704**
Polbooks	105	**0.158**	0.224	**0.415**	0.358
Email	986	0.664	**0.295**	0.288	**0.644**
Cora	2,708	0.190	**0.109**	0.640	**0.725**
PubMed	19,717	0.198	**0.176**	0.432	**0.477**
$B_{\mu=0.6}$	5,000	0.599	**0.329**	0.376	**0.646**
$B_{\mu=0.7}$	5,000	0.700	**0.300**	0.275	**0.676**
$B_{\mu=0.8}$	5,000	0.798	**0.674**	0.177	**0.300**
$B_{\mu=0.9}$	5,000	0.899	**0.893**	0.076	**0.081**

CD accuracy using the *Normalized Mutual Information* (NMI) [2], which is widely used in CD research. NMI measures how much information the ground-truth community structure and the predicted one have in common.

4 EXPERIMENTAL RESULTS

To evaluate the effectiveness of GraphReform^{CD}, we try to answer the following key questions via our experiments:

- (Q1) Does the reformulation reduce misleading information in the original graph?
- (Q2) Does the graph reformulated by GraphReform^{CD} improve the CD accuracy, compared with the original one?

4.1 Changes in Inter-Community Edge Ratio and Modularity (Q1)

Table 1 shows the ratio of inter-community edges to all edges (i.e., $|E^{inter}|/|E|$) and the modularity (i.e., Q) of the ground-truth community structure before (i.e., Orig.) and after (i.e., Ours) the reformulation. Here, E^{inter} refers to the set of inter-community edges in the graph. The lower inter-community edge ratio and higher modularity are shown in boldface for each dataset. In Table 1, we observe that GraphReform^{CD} successfully reduces the ratio of inter-community edges to all edges in most of the graphs, hence increasing the modularity of the ground-truth community structure. The datasets with *a very small number of nodes* (34 for Karate and 105 for Polbooks) only show lower modularities after the reformulation. However, even in such datasets, we found that the reformulated graphs have *more intra-community edges* than the original ones, resulting in higher accuracies (as explained in Section 4.2). We thus conclude that the graphs have been reformulated by our GraphReform^{CD} so that their community structure could be revealed more clearly.

4.2 Changes in Accuracies (Q2)

Table 2 summarizes the accuracy (in terms of NMI) of five algorithms performed on six real-world datasets. A boldface value indicates the highest NMI *for each dataset*, and an italicized value with '*' indicates the highest NMI *for each dataset before the reformulation.* For instance, Infomap shows the highest NMI of 0.711 among all CD algorithms with the original graph in a Karate dataset. With our reformulated graph, SHRINK, Louvain, and Infomap show perfect

Table 2: Accuracies (NMIs) before and after reformulation with five CD methods on six real-world datasets.

Method	Karate			Football		
	Orig.	Ours	Gain (%)	Orig.	Ours	Gain (%)
SHRINK	0.124	**1.000**	707.2	0.849	0.916	7.9
SCAN	0.410	0.862	110.1	0.733	0.937	21.9
Louvain	0.605	**1.000**	65.2	0.908	0.927	2.1
Infomap	*0.711**	**1.000**	40.6	0.924	0.927	0.3
BlackHole	0.560	0.929	27.6	*0.933**	**0.939**	0.6
Method	**Polbooks**			**Email**		
	Orig.	Ours	Gain (%)	Orig.	Ours	Gain (%)
SHRINK	0.126	**0.636**	406.5	0.193	0.593	206.7
SCAN	0.454	0.567	24.9	0.564	0.687	21.8
Louvain	0.460	0.607	31.8	*0.665**	**0.736**	10.7
Infomap	0.541	0.612	13.1	0.642	0.721	12.3
BlackHole	*0.547**	0.620	13.4	0.641	0.694	7.2
Method	**Cora**			**PubMed**		
	Orig.	Ours	Gain (%)	Orig.	Ours	Gain (%)
SHRINK	0.184	0.419	127.4	0.052	0.228	339.3
SCAN	0.468	0.476	1.3	0.262	0.283	7.7
Louvain	*0.477**	**0.484**	1.5	*0.263**	**0.322**	22.7
Infomap	0.471	**0.484**	2.9	0.245	0.318	29.7
BlackHole	0.379	0.439	15.9	0.205	0.230	12.5

Table 3: Accuracies (NMIs) before and after reformulation with five CD methods on four benchmark datasets.

Method	$B_{\mu=0.6}$			$B_{\mu=0.7}$		
	Orig.	Ours	Gain (%)	Orig.	Ours	Gain (%)
SHRINK	0.082	0.340	312.6	0.084	0.323	284.9
SCAN	0.491	0.604	23.1	0.447	0.586	31.0
Louvain	0.590	0.691	17.0	0.434	**0.667**	53.7
Infomap	*0.618**	**0.699**	13.1	*0.450**	0.656	45.7
BlackHole	0.195	0.474	143.1	0.122	0.331	171.6
Method	$B_{\mu=0.8}$			$B_{\mu=0.9}$		
	Orig.	Ours	Gain (%)	Orig.	Ours	Gain (%)
SHRINK	0.082	0.213	159.0	0.082	0.197	139.5
SCAN	*0.401**	0.567	41.5	*0.364**	0.554	52.4
Louvain	0.373	**0.654**	75.6	0.356	**0.647**	81.7
Infomap	0.310	0.625	101.6	0.295	0.612	107.7
BlackHole	0.106	0.263	148.6	0.097	0.233	139.5

accuracies (i.e., NMI of 1.000) where the gain is 707.2% for SHRINK, 65.2% for Louvain, and 40.6% for Infomap. Since Infomap is the best performer with the original graph in a Karate dataset, we would say (conservatively) that the gain obtained from the reformulation is 40.6% in a Karate dataset. We observe that the CD result on the reformulated graph shows higher accuracy than that on the original graph *consistently and universally with all methods on all datasets*. The gain is 40.6%, 0.6%, 16.3%, 10.7%, 1.5%, and 22.7% in Karate, Football, Polbooks, Email, Cora, and PubMed datasets, respectively. Notably, the existing CD algorithms already using the structural similarity (i.e., SHRINK, SCAN, and BlackHole) also achieve significant accuracy improvements, thanks to our GraphReform$^{\text{CD}}$.

Table 3 summarizes the accuracy (in terms of NMI) on the synthetic datasets generated by the LFR-benchmark [5]. The accuracy from the reformulated graph is *consistently and universally* higher than that from the original one *for all combinations of datasets and CD algorithms*. Also, as the μ value of the original graph increases, the CD accuracy on the original graph decreases *rapidly*; on the other hand, the accuracy on the reformulated graph decreases *slowly*. We note, the higher the μ value, the more misleading information the graph has in the LFR-benchmark dataset. We thus conclude that the proposed GraphReform$^{\text{CD}}$ effectively improves the accuracy of CD even in more difficult cases by successfully addressing the issue of misleading information.

5 CONCLUSIONS

In this paper, we have proposed GraphReform$^{\text{CD}}$ that reformulates a given graph into a new one that enables more-accurate CD with the same algorithm. GraphReform$^{\text{CD}}$ constructs a kNN graph by making each node in the original graph have new k chances of connecting itself to other nodes that are most likely to belong to the same community together with the node. As a result, the reformulated graph contains misleading information much less than the original one in the perspective of CD. The results of extensive experiments demonstrate that using the graph reformulated by GraphReform$^{\text{CD}}$ enables a CD algorithm to find a more-accurate community structure than using the original graph.

ACKNOWLEDGMENTS

This work was supported by the National Research Foundation of Korea (NRF) and Institute of Information & Communications Technology Planning & Evaluation (IITP) (NRF-2018R1A5A7059549, NRF-2020R1A2B5B03001960, and IITP No. 2020-0-01373) grant funded by the Korean Ministry of Science and ICT. Also, we thank the Naver Corporation for their support including computing environment and data, which helped us greatly in performing this research successfully.

REFERENCES

[1] Vincent D. Blondel et al. 2008. Fast Unfolding of Communities in Large Networks. *Journal of Statistical Mechanics: Theory and Experiment* 2008, 10 (2008), P10008.
[2] Leon Danon et al. 2005. Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment* 2005, 09 (2005), P09008.
[3] Aditya Grover and Jure Leskovec. 2016. Node2vec: Scalable Feature Learning for Networks. In *Proc. of ACM SIGKDD*. 855–864.
[4] Jiawei Han, Micheline Kamber, and Jian Pei. 2011. Data Mining Concepts and Techniques. *The Morgan Kaufmann Series in Data Management Systems* 5, 4 (2011), 83–124.
[5] Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. 2008. Benchmark Graphs for Testing Community Detection Algorithms. *Physical Review E* 78 (2008), 046110. Issue 4.
[6] Jure Leskovec and Andrej Krevl. 2014. SNAP Datasets: Stanford Large Network Dataset Collection. http://snap.stanford.edu/data.
[7] Sungsu Lim, Junghoon Kim, and Jae-Gil Lee. 2016. BlackHole: Robust Community Detection Inspired by Graph Drawing. In *Proc. of IEEE ICDE*. 25–36.
[8] Mark E. J. Newman. 2006. Modularity and Community Structure in Networks. *Proceedings of the National Academy of Sciences* 103, 23 (2006), 8577–8582.
[9] Mark E. J. Newman. 2013. Network Data. http://www-personal.umich.edu/~mejn/netdata/.
[10] Martin Rosvall and Carl T. Bergstrom. 2008. Maps of Random Walks on Complex Networks Reveal Community Structure. *Proceedings of the National Academy of Sciences* 105, 4 (2008), 1118–1123.
[11] Divya Sardana and Raj Bhatnagar. 2014. Graph Clustering using Mutual K-Nearest Neighbors. In *International Conference on Active Media Technology*. 35–48.
[12] Jian Tang et al. 2015. LINE: Large-Scale Information Network Embedding. In *Proc. of WWW*. 1067–1077.