Which Algorithm Performs Best: Algorithm Selection for Community Detection

Gaoyang Guo
School of Software, Tsinghua University, Beijing 100084, China
ggy16@mails.tsinghua.edu.cn

Chaokun Wang
School of Software, Tsinghua University, Beijing 100084, China
chaokun@tsinghua.edu.cn

Xiang Ying
School of Software, Tsinghua University, Beijing 100084, China
yingx14@mails.tsinghua.edu.cn

ABSTRACT

A myriad of community detection methods have been designed to discover communities based on specific network features in different disciplines, such as sociology, physics, and computer science. Consequentially, we have to face the problem of Algorithm Selection for Community Detection (**ASCD**): Given a specific network, which algorithm should we select to reveal its latent community structures? In this study, we propose a model called **CYDES** to address the **ASCD** problem. **CYDES** consists of two parts, namely feature matrix generation and algorithm classification. We combine three effective feature extraction methods with the idea of BOW model to construct a fixed-size feature matrix. After a nonlinear transformation to the feature matrix, a softmax regression model is utilized to generate a classification label representing the best community detection algorithm we select. Extensive experimental results demonstrate that **CYDES** has high algorithm selection quality for community detection in networks.

CCS CONCEPTS

• **Information systems** → *Social recommendation*; *Content analysis and feature selection*; • **Networks** → *Social media networks*;

KEYWORDS

algorithm selection, community detection, classification

ACM Reference Format:
Gaoyang Guo, Chaokun Wang, and Xiang Ying. 2018. Which Algorithm Performs Best: Algorithm Selection for Community Detection. In *WWW'18 Companion: The 2018 Web Conference Companion, April 23–27, 2018, Lyon, France.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3184558.3186912

1 MOTIVATION

This paper addresses the problem of Algorithm Selection for Community Detection (**ASCD**), which is illustrated in Figure 1. To the best of our knowledge, this is the first time that the problem is proposed.

We use an undirected graph $G = (V, E)$ to denote a network, where V is the node set and E is the edge set. Let $A = \{a^1, a^2, \ldots, a^k\}$ denote the community detection algorithm set consisting of k different algorithms. Given a network G, the **ASCD** problem is to select an algorithm $a \in A$ which has the best performance for G. Formally, the problem aims at learning a function T mapping a network G into the best algorithm $a \in A$ for G, i.e., $a = T(G)$. We use the popular metric NMI (Normalized Mutual Information) to measure the performance of algorithms for a network G in this paper. The algorithm with the highest NMI value is the best algorithm.

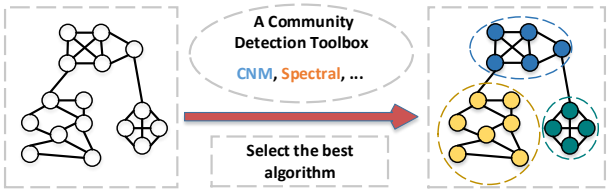

Figure 1: An illustration of the ASCD problem. A community detection toolbox, such as CoDeT [6], includes CNM, Spectral, and other algorithms. Given a network, the aim of the ASCD problem is to select a community detection algorithm from the toolbox which performs best on the network.

2 PROPOSED METHODS

The **CYDES** model consists of two parts. Given a network G, 1) it generates a feature matrix $\mathbf{D}$ for G, and then 2) it performs a nonlinear transformation on $\mathbf{D}$ and predicts the best community detection algorithm $a \in A$ with a softmax regression model.

2.1 Feature matrix generation

We propose three effective feature extraction methods, which are based on 2-hop neighbor structure, normalized neighbor structure and 2^{nd} order random walk [1], respectively.

2-hop neighbor structure. For each node $v \in V$, we construct a local feature vector $\mathbf{b}_{\mathbf{2-hop}}$ based on its 2-hop neighbor structure. We use BFS (Breadth First Search) strategy to extract the subgraph G' within 2 hops starting from v. $\mathbf{b}_{\mathbf{2-hop}}$ for v consists of three parts: 1) the size of G', 2) the density of G', 3) the maximum p degree values in G', where p is a constant. The three parts of values are concatenated to construct $\mathbf{b}_{\mathbf{2-hop}}$. Let $\mathbf{B}_{\mathbf{2-hop}}$ denote the set of all $|V|$ local feature vectors based on 2-hop neighbor structure in G.

Normalized neighbor structure. For each node $v \in V$, we construct a fixed-length local feature vector $\mathbf{b}_{\mathbf{norm}}$ based on its normalized neighbor structure. We also utilize the BFS strategy to get the subgraph G' within 2 hops starting from v. We perform graph normalization [2] on G', which is an injective graph labeling procedure. According to the labeling results, the top q nodes are chosen to form a new subgraph G'', where q is a constant. $\mathbf{b}_{\mathbf{norm}}$ for v is constructed simply by expanding G'' into one dimension. The advantage of this method is that the graph normalization ensures a unified way to get the information of different neighbor structures.

WWW'18 Companion, April 23–27, 2018, Lyon, France

ACM ISBN 978-1-4503-5640-4/18/04.
https://doi.org/10.1145/3184558.3186912

Let $\mathbf{B_{norm}}$ denote the set of all $|V|$ local feature vectors based on the normalized neighbor structure in G.

The 2^{nd} order random walk. For each node $v \in V$, we construct a fixed-length local feature vector $\mathbf{b_{rw}}$ based on the 2^{nd} order random walk. We simulate a 2^{nd} order random walk of length l starting from v. Let N and M denote all nodes and edges in the random walk, respectively. $\mathbf{b_{rw}}$ for v consists of three parts: 1) all degree values of nodes in N, 2) the length of the longest path in the random walk, 3) all values of triangle number containing each edge $e \in M$. All above three parts of values are concatenated to construct $\mathbf{b_{rw}}$. Let $\mathbf{B_{rw}}$ denote the set of all $|V|$ local feature vectors based on the 2^{nd} order random walk in G.

Generate network feature matrices based on BOW model. Given a network G, three different local feature vector sets for G, i.e., $\mathbf{B_{2-hop}}$, $\mathbf{B_{norm}}$, and $\mathbf{B_{rw}}$, are extracted by three methods above. We utilize the idea of BOW (Bag of Words) model to transform these three sets of local feature vectors into three c-dimensional vectors. Let $\mathbf{B_m} = \{\mathbf{b_m^1}, \mathbf{b_m^2}, ..., \mathbf{b_m^{|V|}}\}$ denote one of three sets of local feature vectors, where $m \in \{2-hop, norm, rw\}$ represents the category. Our goal is to transform $\mathbf{B_m}$ into a c-dimensional feature vector $\mathbf{d_m}$. First, the K-means method is used to cluster all local feature vectors with category m from all training networks into c classes. Then, for the ith dimension of $\mathbf{d_m}$, we count the number of local feature vectors $\mathbf{b_m}$ belonging to the ith class according to the clustering results and assign this value to it. Last, $\mathbf{d_m}$ denotes a c-dimensional feature vector of category m for G. Now, $\mathbf{B_{2-hop}}$, $\mathbf{B_{norm}}$, and $\mathbf{B_{rw}}$ are transformed into $\mathbf{d_{2-hop}}$, $\mathbf{d_{norm}}$, and $\mathbf{d_{rw}}$, which are used to construct the network feature matrix $\mathbf{D}$ for G, i.e., $\mathbf{D} = (\mathbf{d_{2-hop}}, \mathbf{d_{norm}}, \mathbf{d_{rw}})$.

2.2 Algorithm classification

Given a network G and its network feature matrix $\mathbf{D}$, we firstly perform a nonlinear transformation on $\mathbf{D}$: $\mathbf{z} = \sigma(\mathbf{D^T w})$, where $\mathbf{w}$ is a c-dimensional weight vector, and $\sigma(x) = 1/(1+e^{-\alpha x})$ is the adaptive sigmoid function. Since the community detection algorithm set A consists of k algorithms [5], we can treat the **ASCD** problem as a k-classification problem. Then, we train a softmax regression model by minimizing the loss function L as

$$\min L = -\ln \frac{e^{(\mathbf{S^T z})_y}}{\sum_{i=1}^{k} e^{(\mathbf{S^T z})_i}} + \lambda ||\mathbf{w}||_2^2 + \gamma ||\mathbf{S}||_2^2 \quad (1)$$

where y denotes the target classification label representing the best algorithm, $\mathbf{S}$ denotes a weight matrix with size $(3*k)$, $(\mathbf{S^T z})_\mathbf{i}$ denotes the ith dimension of $\mathbf{S^T z}$, λ and γ denote two tunable regularization parameters. Finally, the output label of the softmax regression model just represents the best algorithm we select for G.

3 EXPERIMENTS

Since the public real network data sets are relatively limited to obtain, we use the widely used tool LFR to generate 500 synthetic networks, whose numbers of the nodes ($|V|$) is evenly distributed from 300 to 2,500 and average degrees are uniformly distributed over $[4, 40]$. We select ten community detection algorithms including most kinds of methods in recent years: CNM, SCP, Radicchi, M-DMF, Spectral, M-DSGE, LPA, gCluskeleton, HANP and Attractor. We conduct all ten community detection algorithms on the synthetic networks and compute their NMI values. The algorithm with the highest NMI value just represents the best one and is regarded as the target classification label. We select 350 networks randomly for training and use remaining 150 networks for testing.

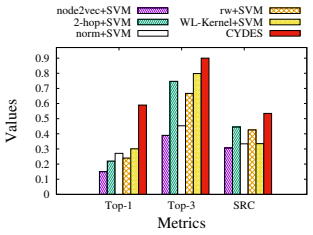

Figure 2: Comparison of the performance

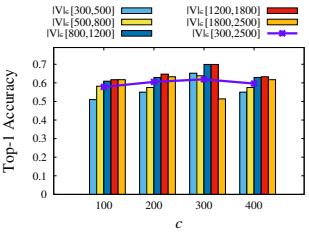

Figure 3: Sensitivity of dimension c and node number $|V|$

Figure 2 illustrates Top-1, Top-3 and average SRC (Spearman Rank Correlation) [4] values of all the baselines and **CYDES** on the testing networks. We separately combine $\mathbf{d_{2-hop}}$, $\mathbf{d_{norm}}$ and $\mathbf{d_{rw}}$ with the SVM classification model to act as three baselines, which are denoted by "2-hop+SVM", "norm+SVM", and "rw+SVM", respectively. We also use two SVM classification models based on node2vec [1] and WL-Kernel method [3] as the other two baselines, which are represented as "node2vec+SVM" and "WL-Kernel+SVM", respectively. We can see that **CYDES** has the best performance, which indicates that **CYDES** is effective for the **ASCD** problem. Furthermore, we analyze the sensitivity of feature dimension c and node number $|V|$ as shown in Figure 3. $|V|$ is divided into five intervals, which are denoted by five different colors of bars, respectively. The line shows the average performance on all testing networks. The results show that **CYDES** achieves the best top-1 accuracy when c is 300 and it is more effective for networks with $|V|$ between 800 and 1,800.

4 CONCLUSION

The problem of **ASCD** is presented in this study. A model called **CYDES** is proposed to deal with the problem. **CYDES** includes two parts, feature matrix generation and algorithm classification. The experimental results show **CYDES** is effective for the **ASCD** problem. In the future, more representative features of networks will be considered with the help of deep learning methods.

5 ACKNOWLEDGMENTS

This work is supported in part by the National Key Research and Development Program of China (No. 2017YFC0820402), the Intelligent Manufacturing Comprehensive Standardization and New Pattern Application Project of Ministry of Industry and Information Technology (Experimental validation of key technical standards for trusted services in industrial Internet), and the China National Arts Fund (No. 20164129). Chaokun Wang is the corresponding author.

REFERENCES

[1] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *SIGKDD*. ACM, 855–864.
[2] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. 2016. Learning convolutional neural networks for graphs. In *ICML*. 2014–2023.
[3] Nino Shervashidze, Pascal Schweitzer, Erik Jan van Leeuwen, Kurt Mehlhorn, and Karsten M Borgwardt. 2011. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research* 12, Sep (2011), 2539–2561.
[4] Charles Spearman. 1904. The proof and measurement of association between two things. *The American journal of psychology* 15, 1 (1904), 72–101.
[5] Meng Wang, Chaokun Wang, Jeffrey Xu Yu, and Jun Zhang. 2015. Community Detection in Social Networks: An In-depth Benchmarking Study with a Procedure-Oriented Framework. *Proceedings of the VLDB Endowment* 8, 10 (2015), 998–1009.
[6] Yifei Yue, Chaokun Wang, Xiang Ying, and Jun Qian. 2017. CoDeT: An Easy-to-Use Community Detection Tool. *International Journal of Data Mining and Bioinformatics* 19, 1 (2017), 52–74.