Knowledge and Information Systems (2020) 62:4277–4296
https://doi.org/10.1007/s10115-020-01490-z

REGULAR PAPER

A joint optimization framework for better community detection based on link prediction in social networks

Shu-Kai Zhang[1] · Cheng-Te Li[2] · Shou-De Lin[1]

Received: 4 May 2019 / Accepted: 6 July 2020 / Published online: 17 July 2020

Abstract
Real-world network data can be incomplete (e.g., the social connections are partially observed) due to reasons such as graph sampling and privacy concerns. Consequently, communities detected based on such incomplete network information could be not as reliable as the ones identified based on the fully observed network. While existing studies first predict missing links and then detect communities, in this paper, a joint optimization framework, *Communities detected on Predicted Edges*, is proposed. Our goal to improve the quality of community detection through learning the probability of unseen links and the probability of community affiliation of nodes simultaneously. Link prediction and community detection are mutually reinforced to generate better results of both tasks. Experiments conducted on a number of well-known network data show that the proposed COPE stably outperforms several state-of-the-art community detection algorithms.

Keywords Social Network Analysis · Community Detection · Link Prediction · Joint Optimization

1 Introduction

Community structure is one of the essential tasks in social network analysis [14]. Community structure refers to that densely inter-connected nodes who share similar attributes tend to belong to the same communities. Network community detection has attracted significant attention in the last decade. Most of existing studies presume that the network structures are complete, i.e., edges can be fully observed, when detecting the communities. That said, all of the connections between nodes are available after obtaining them from online social networking platforms. However, it is not this case in the real world. The acquired network datasets are usually incomplete. Only partial connections are observed due to a variety of reasons. First, a complete social network data might be too costly to acquire. Hence, the

✉ Cheng-Te Li
chengte@mail.ncku.edu.tw

[1] Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan

[2] Institute of Data Science, National Cheng Kung University, Tainan, Taiwan

networks are usually collected by sampling methods, which produces considerable missing data. Second, the structures of social networks evolve over time. New connections are created as time proceeds. Hence, the collected network data is only a momentary reflection of the present situation. To this end, relationships appearing at different timestamps can be considered as missing information in the present network. For example, in an academic paper network, new citations will be created in the future (i.e., do not exist at present). Third, due to various privacy concerns, either users in online social platforms or the service providers provide only partial data of social connections, which is essentially an incomplete data.

It is straightforward to improve the performance of community detection by reconstructing the social connections from the missing data. Link prediction, which has been widely studied, is a solution to reconstruct the missing links [28]. In fact, the tasks of link discovery and community detection are correlated. It is because densely inter-connected nodes are more likely to be in the same community. Community affiliations among nodes are also crucial signals for the task of link prediction [41]. Although link prediction and community detection have been researched for decades, very few studies have tried to combine them together for specific tasks. The major technical aim of this paper is to use link prediction to improve the performance of community detection. The most relevant work might be the 2-stage algorithm [7] that first performs link prediction to infer the missing connections, and then detects communities based on the network with new predicted links. The drawback of such an approach is that link prediction and community detection are treated as two separate modules and performed sequentially. The detected communities cannot be utilized to improve the quality of link prediction. Besides, predicted links might further improve the goodness of detected communities. We think community detection and link prediction can be mutually reinforced to boost the performance of both tasks. In other words, iteratively performing these two tasks in an interweaving manner can probably achieve the effect of mutual reinforcement. One should carefully devise the objective that combines both tasks such that the predicted links and the detected communities can positively influence each other.

Given a social network, in this paper, a joint optimization framework, *Communities detected on Predicted Edges* (COPE), is proposed to bring a marriage between link prediction and community detection in a mutually-reinforced manner. The ultimate goal is to improve the quality of detected communities. An novel objective function is devised to not only guarantee the quality of detected communities, but also promise that the outcome can reconstruct the missing links of the given graph. Furthermore, the proposed objective in COPE has nice extensibility to accommodate external information, such as demographic attributes of nodes if accessible, in the joint learning framework. Moreover, COPE is composed of several meaningful parameters that reflect the mechanism of link prediction and community detection. Hence, COPE is equipped with explainability to understand which factors significantly influence the performance. We conduct the experiments on a number of well-known network datasets, and the results demonstrate that COPE can stably outperform the typical link prediction-based community detection method and a number of state-of-the-art community detection methods. We summarize the contributions of this paper as follows.

– Conceptually, we propose to better identify community structures in networks whose links are partially observed (i.e., incomplete). The idea is to simultaneously perform community detection and link prediction in a mutually-reinforced manner, rather than treating tasks separately.
– Technically, We devise a joint optimization framework, *Communities detected on Predicted Edges* (COPE). COPE optimize a novel objective function that jointly considers the trained social links and the proposed density measure of communities.

- Empirically, COPE is evaluated on a number of well-known network datasets. The results exhibit its superiority over a set of state-of-the-art methods. In addition, COPE is also able to accommodate external information, such as node attributes to further boost the quality of detected communities.

2 Related work

Detecting communities in a graph can allow a variety of applications, such as information diffusion [26], people search [27], and recommender systems [21]. The task of network community detection can be divided into two categories based on how nodes in communities overlap with each other: *disjoint community detection* [4,10,23,25,36,39] and *overlapping community detection* [1,34,46,47]. Although there has been a growing interest in overlapping community detection, the task has been still challenging and full of ambiguity. On the other hand, theoretical approaches in disjoint community detection have been widely studied. For proposing solid results, the disjoint issue is focused in this paper.

The solutions to community detection can be divided into three categories: node clustering, graph decomposition, and node embedding. Node clustering is designed to optimize a certain criterion, which represents the quality of clustering results [48], such as conductance and modularity. For example, Louvain [4] was designed to maximize the modularity score. Since modularity maximization is proved to be an NP-completed problem, Louvain finds the approximate solution for better efficiency. Hierarchical clustering [23] minimizes the number of cuts to separate communities. Through strategically transforming community detection problem to a random walk compression problem, Infomap [39] was designed to maximize compression ratio of the graph. In addition, OSLOM [25] maximizes the value of significance, which is the probability that a given community is found by a null model. SCD [36] maximizes Weighted Community Clustering, which is a metric based on triangle structures in a community. These methods can produce community structures in a given complete network (i.e., links are assumed to be fully observed). However, those models cannot be directly used to deal with networks with missing links (i.e., partially observed networks) since they cannot automatically learn and predict links.

In the methods of graph decomposition, the input graph is transformed to a low-rank representation. Singular value decomposition (SVD) [11] is a well-known generalization of the eigendecomposition of a positive semidefinite normal matrix. Matrix factorization-based methods have been widely studied and have good performance. Among these methods, non-negative matrix factorization (NMF) can be modified to be an approximation to maximize modularity and has a probabilistic interpretation [10]. The proposed COPE is inspired by the probabilistic meaning of the NMF technique. Bigclam [47] is also a NMF-based method for overlapping community detection. Through some probabilistic constraints, the output of those methods is explainable. However, those constraints limit the growth of latent vectors and the usage of the output. Since there are learning-based algorithms, the impotency of external information can be considered by a suitable design. The reconstructed graph can also be used to perform link prediction. Those learning-based algorithms take more time in computing.

In the node embedding methods for community detection, nodes are transformed into a latent vector space. After the transformation, one can perform conventional clustering algorithms on the derived embedding vectors of nodes. The well-known node embedding methods include node2vec [17], LINE [42], and MARINE [13]. Their obtained vectors had been validated to be somehow more accurate to describe the nodes. However, since such embedding vectors is not explainable. One cannot learn which factors have higher impact on the tasks of community detection and link prediction.

Since the proposed joint optimization of link prediction and community detection can be considered as a kind of multi-task learning, i.e., combing two tasks together in the objective function. Hence, we also need to discuss the relevant studies on applying multi-task learning to graph data. LoNGAE [43] devises an autoencoder model whose allows the input of a graph, and aims at jointly predicting links and classifying nodes under a semi-supervised learning manner. Their goal is to generate node embeddings for downstream tasks, which is different from our end-to-end COPE model. A general graph multi-task learning framework [30] is developed to learn the correlation of multiple tasks via a graph neural network model. However, their goal is to construct the graph for multi-task learning, rather than utilizing multi-task learning for graph-related tasks. DEMO-Net [45] is a degree-specific graph neural network model that preserves various degree properties into multi-task graph convolution so that both tasks of node classification and graph classification can be better performed. It is apparent that DEMO-Net is neither applicable nor related to community detection and link prediction. FelMuG [35] is a task sensitive feature exploration and learning algorithm for multi-task graph classification. It presumes the labels of different graph-level classification tasks are available, and thus aims to distinguish task-specific features from each other. FelMuG requires multi-tasks' labels for feature learning, our COPE needs no label information as both links and community structure are naturally accessible in the given graph.

3 Problem statement

Given an incomplete graph with a set of nodes and a subset of links between nodes, our goal is to detect the network communities, together with a set of predicted missing links. Specifically, we are given an incomplete graph with missing links (not shown in the input graph), we aim at jointly inferring the missing links (shown in the output graph) and detecting network communities for nodes so that nodes belong to the same communities in the ground truth are grouped.

We define $G = (N, E)$ as a social network, where N is the set of nodes and E is the entire set of edges between nodes. We also define $H = (N, \hat{E})$ as the social network with missing edges, where $\hat{E} \subseteq E$. The problem can be described as: given the incomplete network H, the goal is to (a) detect the set of communities (denoted by S), $S = \{s_1, s_2, \ldots, s_k\}$, where s_i $(i = 1, 2, \ldots, k)$ is the i-th community, k is the number of communities, and $\bigcup_{i=1}^{k} s_i = N$, and (b) predict the set of unobserved links, $E \backslash \hat{E}$.

4 The COPE framework

We cast the joint community detection with link prediction into an optimization task. We propose a novel factorization-based model that jointly optimize the objectives of link prediction and community detection. The basic idea is threefold. (1) Since the given network contains missing links, which need to be considered in the detection of communities, we resort to estimate the link probability values between node pairs. Node pairs with higher link probability values tend to be the missing connections and are essential when detecting communities. Therefore, node pairs with the estimated link probabilities are exploited to find the communities. (2) The estimation of link probability values is automatically learned based on existing links in the network. The detection of network communities is based on *probabilistic modularity*, which is a goodness measure of network communities. (3) The iterative updating process is designed to simultaneously minimize the link prediction loss and maximize the goodness of detected communities. We implement such three ideas by first devising *adjacency factor* and *density factor*.

4.1 Adjacency factor

The adjacency factor is proposed to learn the link probabilities of node pairs given their adjacent structures in a social network. A network can be represented by a binary adjacency matrix $A = [A_{i,j}] \in \mathbb{R}^{n\times n}$, where $A_{i,j}$ is 1 if edge (i, j) exists, else $A_{i,j}$ is 0. We propose a feature-based sigmoid function to estimate the link probability $P_{i,j}$ between node i and node j, as written below:

$$P_{i,j} = \frac{1}{1 + \exp^{-wx_{ij}+b}}, \tag{1}$$

where $x_{i,j}$ is a list of features that depict the neighboring network structure surrounded by edge (i, j), w is a numerical vector representing the weights of features, and b is the bias term. The parameters w and b can be learned from the data. The *adjacency factor* denoted by AF, as described below, can be derived by minimizing the square error between the estimated probability $P_{i,j}$ and the edges in the network.

$$\mathrm{AF} = \sum_{i\in N, j\in N} \frac{1}{2}(P_{i,j} - A_{i,j})^2 \tag{2}$$

4.2 Density factor

The density factor is designed to guarantee that nodes in the detected communities are densely connected with each other and loosely connected with nodes in other communities. We consider that in a *strong* community each node has more connections within the same community than with nodes in the rest of the network [37]. Let S be the set of communities in the network, and let $C_i \in \mathbb{R}^{+|S|}$ be the latent vector of node i, where $C_i = \left\{C^i_1, C^i_2, \dots, C^i_{|S|}\right\}$, and C^i_s indicates how likely a node i belongs to a community s. So we can have the latent matrix $C = [C^i_s]$ consisting of a list of latent vectors $C_1, C_2, \dots, C_{|N|}$ for all of the nodes in the network, which is the learning target of our COPE model from the network data. To implement the idea of the density factor, we define the external density (D_E) and the internal density (D_I) between nodes i and node j in a probabilistic form:

$$D_I(i, j) = (P_{i,j} - \lambda)(C_i \cdot C_j) \tag{3}$$

$$\begin{aligned} D_E(i, j) &= (P_{i,j} - \lambda)\left(\frac{1}{2}(C_i \cdot (1 - C_j) + (1 - C_i) \cdot C_j)\right) \\ &= (P_{i,j} - \lambda)\left(\frac{1}{2}\sum C_i + \frac{1}{2}\sum C_j - C_i \cdot C_j\right) \end{aligned} \tag{4}$$

In the special case that both $\sum C_i$ and $\sum C_j$ equal to 1, the last term in Eq. 4 can be rewritten as $(1 - C_i \cdot C_j)$. It is apparent that to some extent Eq. 4 to be correlated with Eq. 3. Note that since we will be more confident on the estimated link probability when $P(i, j)$ is closer to either 0 or 1, the term λ is added into these equations so that the derived link probabilities can be more reliable in both density values. We set $\lambda = \frac{1}{2}$ by default, and will study how it affects the performance in the experiment.

Based on the concept of network community structure, the density factor DF can be obtained by minimizing Eq. 3 and maximize Eq. 4. Hence the density factor can be accordingly defined as below:

$$\mathrm{DF} = \sum_{i \in N, j \in N} D_E(i, j) - D_I(i, j) \tag{5}$$

4.3 Combine adjacency and density factor

By combining Eqs. 2 and 5, and the regularization terms in our joint optimization framework, we can have the following final objective function, which will be minimized so that the vector w of feature weights and the community matrix C can be derived.

$$\begin{aligned} \min_{w,C} F(w, C) = \min_{w,C} \alpha & \sum_{i \in N, j \in N} \frac{1}{2}(P_{i,j} - A_{i,j})^2 \\ & + (1-\alpha) \sum_{i \in N, j \in N} (D_E(i, j) - D_I(i, j)) \\ & + \gamma \left(\frac{1}{2} \|w\|^2 + \frac{1}{2} \|C\|^2 \right) \text{s.t. } C_s^i \geq 0 \ (\forall i, s) \end{aligned} \tag{6}$$

where α and γ are the hyperparameters for adjusting the contributions of different factors and the regularization term, respectively.

4.4 Pairwise features

The Link probability is designed to share the learning variables w and C between adjacency factor and density factor. To embed C into the feature vector x_{ij}, in this section, we need to consider C in the design of feature vector $x_{i,j}$, which depicts the potential correlation between each pair of node i and node j. In order to make both the adjacency factor and the density factor share both of the trainable variables C and w, we propose to use the latent matrix C in defining pairwise features. There are three basic ideas behind the design of pairwise features. First, it had been proven that the community information can improve the performance of link prediction [41]. Therefore, we propose to exploit the inner product of the latent community vectors C_i and C_j as a feature that reflects the strength of connection of two nodes. Second, aggregated features and topological features [2], which are widely used in feature-based link prediction tasks, are also considered here. Third, the bias term b is designed to train the intercept of the link probability. What follows provide the list of pairwise features x_{ij} depicting the potential connection between node i and node j, in which $\Gamma(i)$ is the set of neighbors of node i in the network.

- Inner product of the learned embedding vectors: $C_i \cdot C_j$
- Preferential attachment: $|\Gamma(i)| \cdot |\Gamma(j)|$
- Length of the shortest path: $\min(|\mathrm{path}(i, j)|)$
- Number of common neighbors: $|\Gamma(i) \bigcap \Gamma(j)|$
- Adamic–Adar score: $\sum_{z \in \Gamma(i) \bigcap \Gamma(j)} \frac{1}{\log |\Gamma(z)|}$

– Jaccard's Coefficient: $\frac{|\Gamma(i)\bigcap\Gamma(j)|}{|\Gamma(i)\bigcup\Gamma(j)|}$

It is worthwhile noticing that our model allows the flexibility of feature design by adding more features based on other external information if accessible. Note that the listed features are replaceable, rather than absolutely adhering to the proposed method. Therefore, they are not the requirement of COPE. It is acceptable to use only the suggested inner product of the learned embedding, i.e., $C_i \cdot C_j$, as the pairwise feature.

5 Methodology

In this section, we demonstrate how to minimize the non-convex objective function and elaborate the techniques we use in the framework.

5.1 Stochastic gradient descent

Since Eq. 6 is differentiable, a *Stochastic Gradient Descent* (SGD) approach [6] can be implemented to minimize the objective function. We can derive the gradient by the following steps. First, the partial derivative of $P_{i,j}$ with respect to w leads to a standard sigmoid function gradient solver.

$$\frac{\partial P_{i,j}}{\partial w} = P_{i,j}(1 - P_{i,j})x_{i,j} \tag{7}$$

The partial derivative of $P_{i,j}$ with respect to C_i also leads to a standard sigmoid function gradient solver. We use symbol w_C to represent the weight corresponding to the inner product feature.

$$\frac{\partial P_{i,j}}{\partial C_i} = P_{i,j}(1 - P_{i,j})w_C C_j \tag{8}$$

Note that the partial derivative of $P_{i,j}$ with respect to C_j has the same form, in which C_j is replaced by C_i and C_i is replaced by C_j. Let Δ_w denote Eq. 7 and Δ_{C_i} denote Eq. 8. The partial derivative of Eq. 6 with respect to w can be written as follows.

$$\begin{aligned}\frac{\partial F_{i,j}(w, C)}{\partial w} &= \alpha(P_{i,j} - A_{i,j})\Delta_w \\ &+(1-\alpha)\left(\frac{1}{2}\sum C_i + \frac{1}{2}\sum C_j - 2C_iC_j\right)\Delta_w \\ &+\gamma w\end{aligned} \tag{9}$$

The partial derivative of Eq. 6 with respect to C_i can be written as the following.

$$\begin{aligned}\frac{\partial F_{i,j}(w, C)}{\partial C_i} &= \alpha(P_{i,j} - A_{i,j})\Delta_{C_i} \\ &+(1-\alpha)\left(\left(\frac{1}{2}\sum C_i + \frac{1}{2}\sum C_j - 2C_iC_j\right)\Delta_{C_i}\right. \\ &\left.+\left(P_{i,j} - \frac{1}{2}\right)\left(\frac{1}{2} - 2C_j\right)\right) \\ &+\gamma C_i\end{aligned} \tag{10}$$

Note that the partial derivative of Eq. 6 with respect to C_j has the same form, in which C_j is replaced by C_i and C_i is replaced by C_j. Eventually, the updating rules for each (w, C_i, C_j) are as follows:

$$w' = w - \eta \frac{\partial F_{i,j}(w, C)}{\partial w} \tag{11}$$

$$C_i' = \max\left(C_i - \eta \frac{\partial F_{i,j}(w, C)}{\partial C_i},\ \epsilon\right) \tag{12}$$

$$C_j' = \max\left(C_j - \eta \frac{\partial F_{i,j}(w, C)}{\partial C_j},\ \epsilon\right), \tag{13}$$

where ϵ is a small value 10^{-5} to maintain the non-negative constraint: $C_s^i \geq 0$ for every node i and every detected community s, and η is the learning rate.

5.2 Implementation details

5.2.1 Pre-training initialization

Since the two learning variables w and C interact with each other between the adjacency and density factors, we exploit an unsupervised pre-training initialization to help the model training and to make the framework more robust [12]. In this paper, variable C is pre-trained by the k-means algorithm [3] based on the adjacency matrix of the given network. The number of communities, k, is determined based on the mixture model approach [1], in which a hold out set is reserved to fit our model by varying k and evaluate by the likelihood.

5.2.2 Normalization

Normalization aims to meet the property of probability: the value is between 0 and 1. In disjoint community detection, we make sure that $\|C_i\|_1 = 1$ for all possible node i. In other words, the normalization operation $C_i = \left[\frac{C_s^i}{||C_i||}\right]$ is performed for all updated C_i in each optimization mini-batch.

5.2.3 Feature scaling

Feature scaling is known to improve the convergence rate in the process of stochastic gradient descent [18]. Therefore, we standardize the range of unchangeable features by performing the scaling operation: $x_{ij}' = \frac{x_{ij} - \bar{x}}{\sigma}$, where x_{ij}' is the scaled feature value.

5.2.4 Termination criteria

Since there is no validation set in unsupervised learning, we modify the *probabilistic modularity* [4], which is a probabilistic version of the goodness measure of derived communities, to determine when to terminate the optimization process. The modified probabilistic modularity is given by:

$$Q = \frac{1}{2m} \cdot \sum_{i,j} \left(A_{i,j}' - \frac{\left\|A_i'\right\|_1 \left\|A_j'\right\|_1}{2m} \right) \cdot \delta(C_i, C_j) \tag{14}$$

where $A'_{i,j} = 1$ if $(i, j) \in E$ and $A'_{i,j} = P_{i,j}$ if $(i, j) \notin E$. $A'_{i,j}$ represents the link probability between nodes i and node j. In addition, $m = \sum_{i,j} P_{i,j}$ indicates the sum of all probability values between nodes in the network. $\delta(C_i, C_j) = C_i \cdot C_j - \frac{1}{|S|}$, where $\frac{1}{|S|}$ represents the probability that node i and node j belong to the same community by random assigning nodes into communities in a disjoint scenario. Hence, $\delta(C_i, C_j)$ is the difference between the detected communities and the random assignment.

Higher values of probabilistic modularity refer to better quality of the detected communities. In our implementation for the evaluation, if the probabilistic modularity value is not increased from the current batch to the follow-up 20 updating batches, we terminate the learning process and return the community matrix C. Such termination is guaranteed to make the derived communities possess the local maximal value of the probabilistic modularity. According to the community matrix C, since we are detecting disjoint communities, we assign each node i to the community s^* that possesses highest probability, given by: $s^* = \arg\max_s \{C^i_s\}$.

5.3 Complexity analysis

It is apparent that the learning from all pairs of nodes is time-consuming. Both techniques of positive sampling and negative sampling are widely implemented on the graph-based optimization process. While positive samples are existing links in the network, we follow existing studies [2,2,41] to randomly sample non-existent links whose number is equal to positive ones as the negative samples. Consequently, the size of the learning instances is proportional to the number of edges and can be considered as $O(E)$. In addition, since there are inner product terms in the objective function, the time complexity of a training sample is $O(|S|+|w|)$. Eventually, the total complexity of the proposed COPE is $O(|E| \times (|S|+|w|))$. Note that $|S|$ and $|w|$ are usually far less than $|E|$.

6 Experiments

In the experiments, we aim to answer to the following questions. The first is whether the proposed COPE outperform the state-of-the-art methods using incomplete network datasets. The second is whether the predicted links by COPE tend to connect nodes within the same communities.

6.1 Datasets and competitors

We follow existing studies [9,22] on community detection to utilize six typical network datasets. The used real-world network dataset are listed in Table 1. Each of the datasets is associated with the ground-truth communities. Every compared method will be evaluated using ground-truth communities under various metrics, described in the following paragraph. We compare the proposed COPE with the following community detection methods: BPRNMF [10,38], interactive 2-stage link prediction-based community detection (LPCD) [7], the Louvain algorithm [4], Infomap [39], OSLOM [25], SCD [36], and K-means on node2vec embedding (km-node2vec) [3,17]. As for the proposed COPE, we set $[\alpha, \gamma] = [0.5, 0.001]$ so that the adjacency factor and the density factor have equal contribution. Note that it is worthwhile to mention the competing method LPCD [7] in detail. To verify the effectiveness

Table 1 Data statistics

Dataset	# Nodes	# Edges	# Communities
Dolphins [33]	62	159	2
Polbooks [5]	105	441	3
Railway [8]	301	1224	21
Football [16]	115	613	12
Workplace [15]	92	755	5
Karate [49]	34	78	2

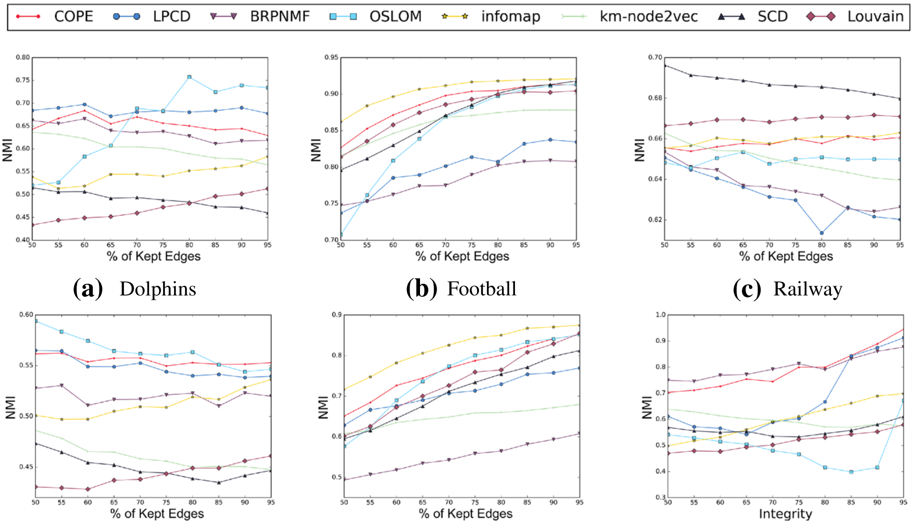
of joint optimization of link prediction and community detection, LPCD runs Link Prediction first to create missing links, and then runs Community Detection on those new graphs. By comparing with LPCD, we aim to prove that joint optimization leads to better performance, comparing to treating such two tasks separately in the manner of one after the other.

6.2 Evaluation settings

The evaluation settings consist of two parts corresponding to two scenarios of missing data in the given social networks: missing links and missing nodes. In the scenario of **missing links**, we randomly remove edges up to different percentages 5%, 10%, ..., and 50%, i.e., keeping 95%, 90%, ..., and 50% of the entire set of edges. The random sampling for edge removal is performed 100 times, so a set Q of 100 different subgraphs can be obtained for each percentage of edge removal. On the other hand, in the scenario of **missing nodes**, we consider two different node removal strategies. One is to remove nodes with 5% highest degree values from the network. The other is to randomly remove 15% nodes from the network. For the latter case, we perform 100 times and obtain 100 incomplete networks. In both strategies, all edges connected with the removed nodes are also removed.

6.3 Evaluation metrics

We consider two evaluation metrics to measure the goodness of the detected communities. First, we make use of *Normalized Mutual Information* (NMI) [20,22,24,44], whose scores are between 0 and 1. NMI is widely used in disjoint community detection. A higher NMI value (close to 1) represents better quality of the detected communities. A recent study [31] has shown that NMI-based metrics cannot properly deal with the finite size effect of communities, ignore the importance of small communities, and cannot be used to evaluate a single community of interest. However, our proposed COPE method is neither to deal with the size of communities, nor to target at any single community of interest. Therefore, we choose the typical NMI as a proper evaluation. The design of incorporating the community size and the single community of interest into our COPE model is left as the future work. Second, we utilize *F1 score*, which is widely used to examine the effectiveness of data clustering algorithm. Higher F1 scores indicate better results of detected communities. Third, given a certain community detection method $\mathtt{M}$ and the set Q of incomplete networks, we define the *Average Rank* (AvgRank), given by $AvgRank(\mathtt{M}, Q) = \frac{\sum_{q \in Q} Rank_q(\mathtt{M})}{|Q|}$, where $Rank_q(\mathtt{M})$ is the rank of method $\mathtt{M}$ among all competing methods in terms of their NMI values for incomplete network q. Lower AvgRank values refer to higher ranking, and indicate better performance of the method.



(a) Dolphins **(b)** Football **(c)** Railway

(d) Polbooks **(e)** Workplace **(f)** Karate

Fig. 1 The average NMI scores by varying different percentages of kept edges

Table 2 The values of average rank for different percentages of kept edges in the missing link scenario

Method	50%	55%	60%	65%	70%	75%	80%	85%	90%	95%
COPE	2.67	2.67	2.33	2.33	2.67	2.67	2.50	2.17	2.33	2.83
LPCD	4.00	4.17	4.50	5.00	5.17	4.83	4.83	4.50	4.67	4.67
BPRNMF	4.50	5.00	5.17	5.17	5.33	5.33	5.50	6.00	5.83	5.83
OSLOM	6.00	5.00	4.67	4.50	3.83	3.83	3.67	3.67	3.83	3.17
Infomap	4.00	4.00	3.67	3.33	3.33	3.17	3.33	3.33	3.17	3.00
km-node2vec	3.83	4.33	4.83	5.00	5.33	5.83	5.83	5.83	6.00	6.67
SCD	5.17	5.33	5.17	5.17	5.00	5.00	5.00	5.17	5.00	5.00
Louvain	5.83	5.50	5.67	5.50	5.33	5.33	5.33	5.33	5.17	4.83

6.4 Community detection results

The results of missing links by varying the percentage of kept edges are shown in Fig. 1. We can find that although the proposed COPE does not lead to the highest NMI values in the six datasets, it is still able to stably maintain higher NMI values, i.e., around the top three, in all datasets. The competitors work better in one dataset but perform worse in other datasets. For example, in the Dolphins data, the NMI values of LPCD is higher than our COPE, but LPCD leads to quite lower NMI values in most of the remaining datasets. Such results exhibit the stability of COPE; that says, the competitors are unstable across datasets. The results by Average Rank, as exhibited in Table 2, can further demonstrate the superior of our COPE over the competitors. It is apparent that COPE has the highest average rank in all datasets. Such results imply that by jointly predicting links and detecting communities, the performance of community detection can be truly boosted.

Table 3 The average rank values for two node removal strategies

Method	S1: 15% Random	S2: 5% Highest-degree
COPE	1.7	1.7
LPCD	4.7	4.7
BPRNMF	4.3	4.8
OSLOM	2.7	3.3
Infomap	6.8	5.8
km-node2vec	5.0	5.8
SCD	3.3	4.2
Louvain	7.5	5.7

S1: the removal of 15% random nodes; S2: the removal of 5% highest-degree nodes

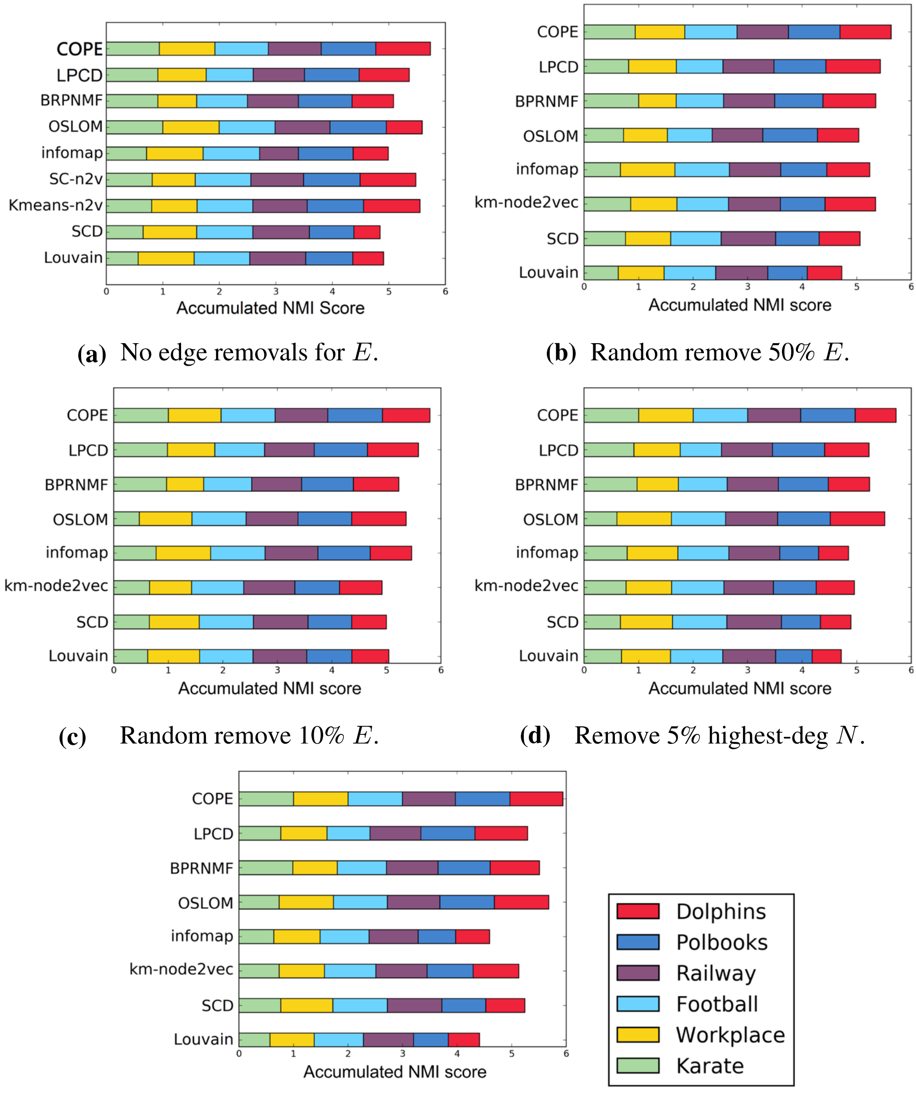
As for the evaluation of missing nodes, the results are shown in Table 3. COPE clearly outperforms all competitors in both random removal and highest-degree removal. The effectiveness of COPE can be hereby verified. Note that we only report the Average Rank here due to page limit. The NMI values and the AvgRank of different kept percentages exhibit similar trends as the scenario of missing links. Moreover, we also report the accumulated NMI and F1 scores over six datasets, as shown in Figs. 2 and 3, in which we (a) randomly remove 50% edges, (b) 10% edges, and (d) 15% nodes, and (c) remove 5% highest-degree nodes. The results again show the superior of COPE.

6.5 In-community link prediction

We want to analyze why the predicted links in COPE can lead to better quality of detected communities. It is believed that a good community tends to have densely connected members. Hence, we hypothesize that given a network with missing links or nodes, if more links within communities can be predicted, the performance of detected communities can get boosted. To verify this hypothesis, we examine whether the in-community links can be accurately predicted by COPE. We compare COPE with LPCD [7], Support Vector Regression (SVR) (linear and RBF versions) [40], and the classical logistic regression [19] in terms of the Area Under ROC curve (AUC) [32]. If both end nodes of a predicted link belong to the same ground-truth community, this predicted link is considered as a successful prediction. In other words, the class label $y_{i,j}$ is positive if both node i and node j belong to the same community in the ground truth, otherwise $y_{i,j}$ is negative. We perform in-community link prediction under the scenario of edge missing. Note similar results can be derived in the node missing scenario, but do not report due to page limit. The resulting AUC scores by varying the percentage of kept edges are show in Fig. 4. It can be clearly observed that COPE significantly outperforms the competitors in most of the six datasets, except Railway. Such results, along with Table 2, imply that accurate prediction of in-community links in an incomplete network can boost the performance of community detection. Note that since SVR with RBF kernel performs best on the Railway data, we may want to develop a non-linear extension of our COPE, which is considered as our future work.

6.6 Hyperparameter sensitivity

We analyze how two hyperparameters α and λ affect the performance of community detection. Hyperparameter α is used to determine the contribution between adjacency factor and density



(a) No edge removals for E.

(b) Random remove 50% E.

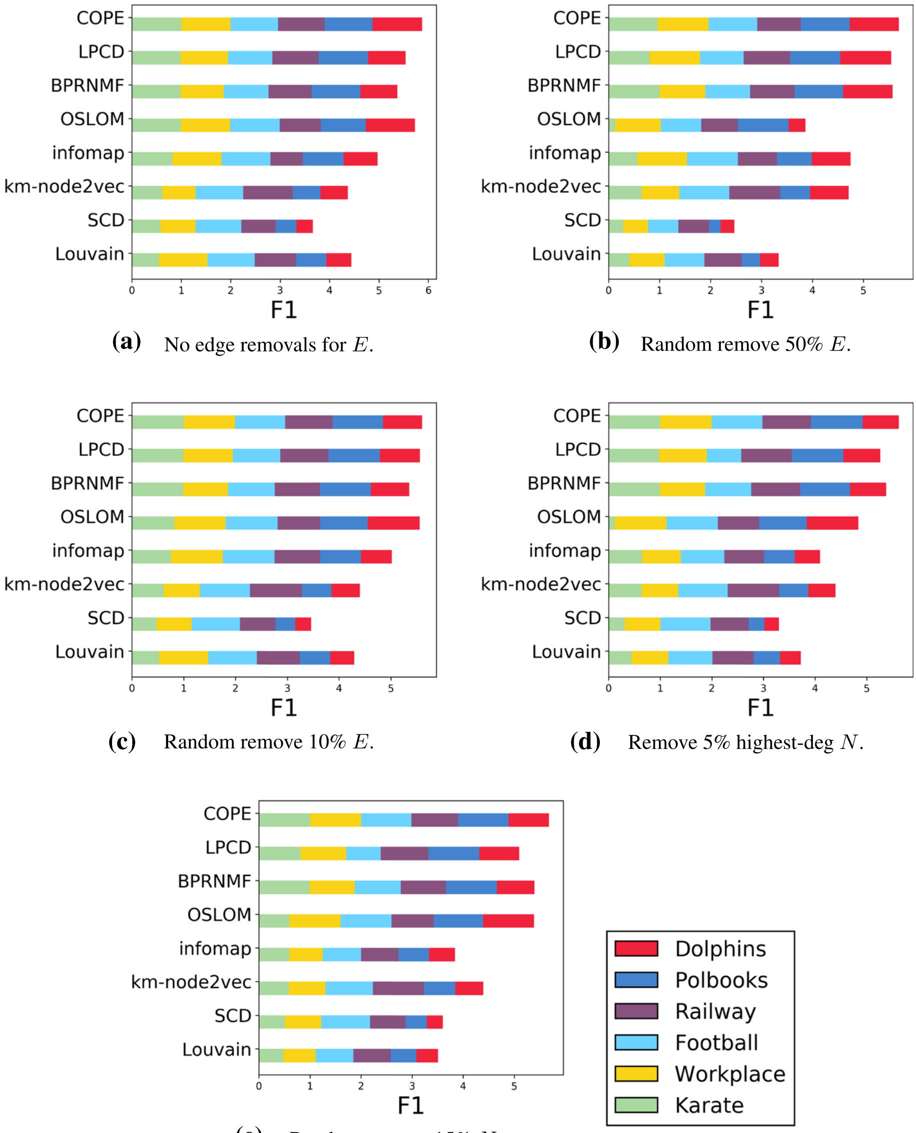
(c) Random remove 10% E.

(d) Remove 5% highest-deg N.

(e) Random remove 15% N.

Fig. 2 The accumulated NMI scores over the six network datasets. Note that a perfect community detection method can make the accumulated NMI value be 6

factor, and hyperparameter λ is created to ensure the derived link probabilities can be more reliable in both density values. We use Dophins and Railway datasets for this experiment, and both NMI and F1 are used as the metrics. The results are shown in Fig. 5. We can find that the change of α does not affect the performance too much, i.e., the change of both NMI and F1 values is quite small. Such a result implies that both adjacency factor (estimating link probabilities) and density factor (estimating the assignment of communities) are effective in generating communities. On the other hand, by changing λ, we can find when λ is around 0.5 to 0.6, both NMI and F1 values are higher. This proves that choosing $\lambda = 0.5$ is acceptable



(a) No edge removals for E.

(b) Random remove 50% E.

(c) Random remove 10% E.

(d) Remove 5% highest-deg N.

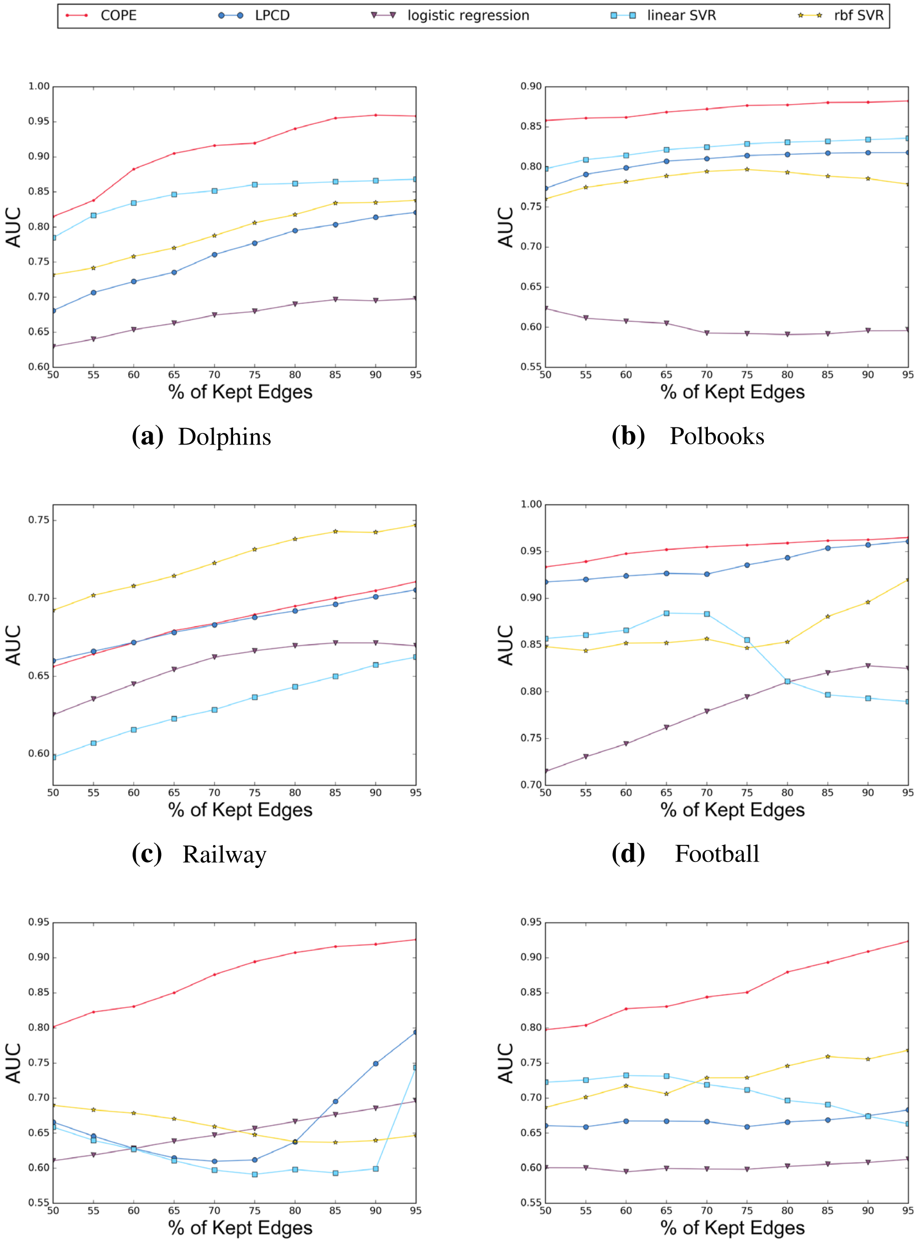
(e) Random remove 15% N.

Fig. 3 The accumulated F1 scores over the six network datasets. Note that a perfect community detection method can make the accumulated F1 value be 6

for better results of community detection. Too larger (e.g., $\lambda = 0.2$ or too smaller (e.g., $\lambda = 0.8$) values of λ lead to worse performance.

7 Conclusions and discussions

This work boosts the performance of disjoint community detection for social networks with partially observed edges, rather than considering complete networks in conventional stud-

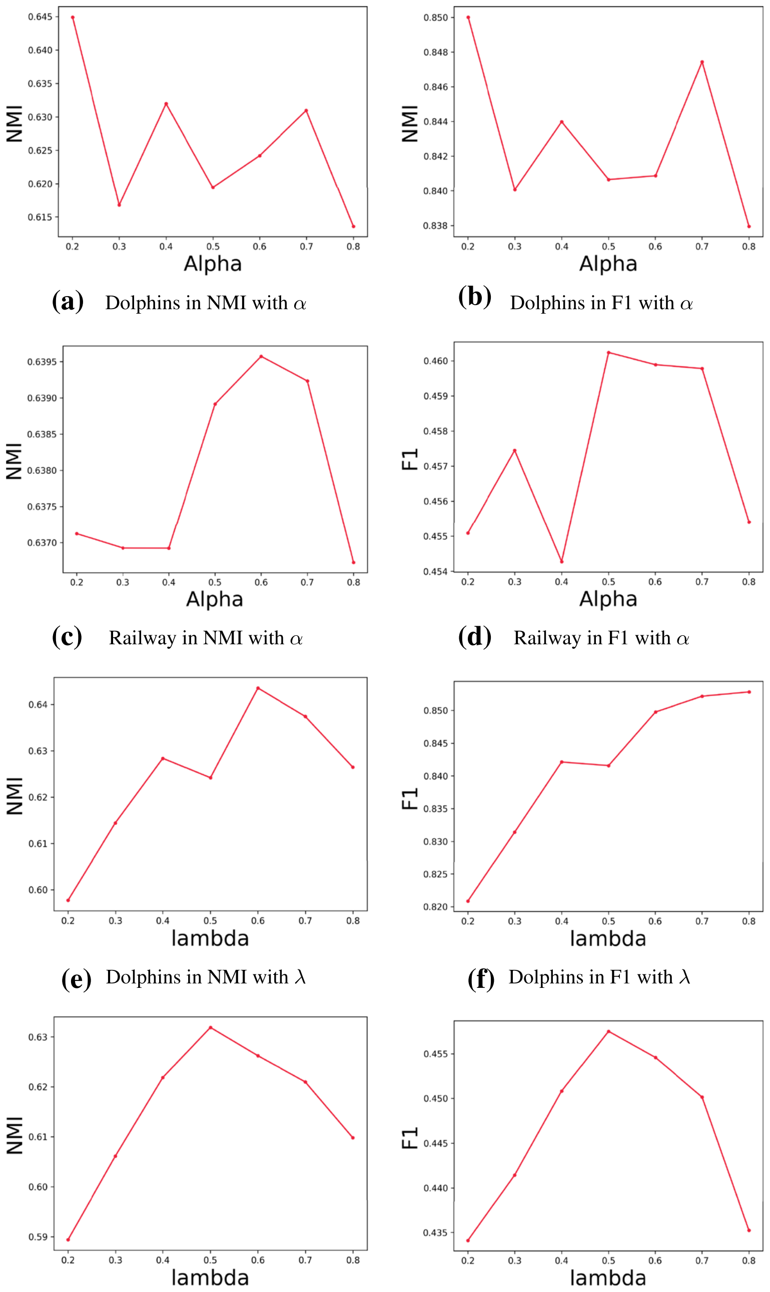


(a) Dolphins **(b)** Polbooks

(c) Railway **(d)** Football

(e) Workplace **(f)** Karate

Fig. 4 The AUC values by varying the percentages of kept edges in community-based link Prediction scenario



(a) Dolphins in NMI with α

(b) Dolphins in F1 with α

(c) Railway in NMI with α

(d) Railway in F1 with α

(e) Dolphins in NMI with λ

(f) Dolphins in F1 with λ

(g) Railway in NMI with λ

(h) Railway in F1 with λ

Fig. 5 Hyperparameter analysis of COPE community detection in terms of NMI and F1 scores using Railway and Dolphins datasets

ies. We successfully bring a tight-coupled marriage between community detection and link prediction, in which the latter is used to infer the missing links within communities. The proposed link probabilities of node pairs not only contribute to the density in a community, but also allow extensive features defined for inferring links, and thus eventually lead to a nice design of the final objective. Such formulation and fusion via joint optimization can be a good attempt for solving network analysis tasks with incomplete set of links. For example, the performance of node label classification and information diffusion prediction in the graph with missing links might be boosted via a joint optimization of link inference, instead of the concatenation of two tasks. This is an alternative but essential finding of this paper.

We have two ongoing follow-up tasks based on COPE. First, we aim at extending COPE to deal with overlapping communities detection by learning multiple affiliations from the community matrix C. Second, scalability is an important issue of COPE. Nevertheless, we have to emphasize that the proposed COPE is not to better detect communities efficiently in large-scale networks, but to verify whether link prediction can improve the effectiveness of community detection. We successfully prove it under the proposed joint optimization model COPE. To enable the scalability, the optimization method needs to be enhanced since we need to examine the adjacency factor and the density factor based on all pairs of nodes in the objective function (Eq. 6), whose computational cost is high. A strategy to remedy such an efficiency issue is to sample pairs of nodes i and j in Eqs. 3 and 4. We leave how to improve the optimization as the future work. We plan to combine COPE with block stochastic gradient ascent [29] to enhance the scalability.

Acknowledgements This work is supported by Taiwan Ministry of Science and Technology (MOST) under Grants 109-2636-E-006-017 (MOST Young Scholar Fellowship), 109-2634-F-002-033 and 108-2218-E-006-036, Academia Sinica under Grant AS-TP-107-M05, Microsoft Research Asia Collaborative Project Funding (2019), and National Center for High-Performance Computing in Taiwan.

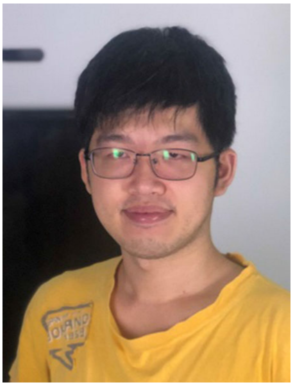
References

1. Airoldi EM, Blei DM, Fienberg SE, Xing EP (2008) Mixed membership stochastic blockmodels. J Mach Learn Res 9(Sep):1981–2014
2. Al Hasan M, Chaoji V, Salem S, Zaki M (2006) Link prediction using supervised learning. In: SDM06: workshop on link analysis, counter-terrorism and security
3. Arthur D, Vassilvitskii S (2007) k-means++: the advantages of careful seeding. In: Proceedings of the eighteenth annual ACM-SIAM symposium on discrete algorithms. Society for Industrial and Applied Mathematics, pp 1027–1035
4. Blondel VD, Guillaume JL, Lambiotte R, Lefebvre E (2008) Fast unfolding of communities in large networks. J Stat Mech Theory Exp 10:P10008
5. Books about us politics. http://networkdata.ics.uci.edu/data.php?d=polbooks
6. Bottou L (2010) Large-scale machine learning with stochastic gradient descent. In: Proceedings of COMPSTAT'2010. Springer, pp 177–186
7. Burgess M, Adar E, Cafarella M (2016) Link-prediction enhanced consensus clustering for complex networks. PLoS ONE 11(5):e0153384
8. Chakraborty T, Srinivasan S, Ganguly N, Mukherjee A, Bhowmick S (2014) On the permanence of vertices in network communities. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp 1396–1405
9. Chen W, Liu Z, Sun X, Wang Y (2016) Community detection in social networks through community formation games. In: Proceedings of the twenty-second international joint conference on artificial intelligence (IJCAI-16). pp 2576–2581
10. Ding C, He X, Simon H.D (2005) On the equivalence of nonnegative matrix factorization and spectral clustering. In: Proceedings of the 2005 SIAM international conference on data mining. SIAM, pp 606–610
11. Drineas P, Frieze A, Kannan R, Vempala S, Vinay V (2004) Clustering large graphs via the singular value decomposition. Mach Learn 56(1):9–33

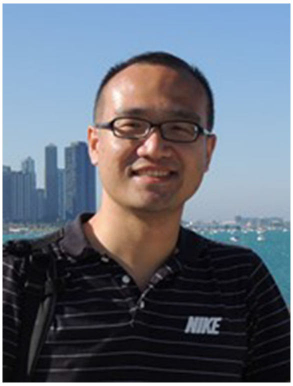
12. Erhan D, Bengio Y, Courville A, Manzagol PA, Vincent P, Bengio S (2010) Why does unsupervised pre-training help deep learning? J Mach Learn Res 11(Feb):625–660
13. Feng MH, Hsu CC, Li CT, Yeh MY, Lin SD (2019) Marine: multi-relational network embeddings with relational proximity and node attributes. In: The world wide web conference. WWW '19, pp 470–479
14. Fortunato S (2010) Community detection in graphs. Phys Rep 486(3):75–174
15. Génois M, Vestergaard CL, Fournet J, Panisson A, Bonmarin I, Barrat A (2015) Data on face-to-face contacts in an office building suggest a low-cost vaccination strategy based on community linkers. Netw Sci 3(3):326–347
16. Girvan M, Newman ME (2002) Community structure in social and biological networks. Proc Natl Acad Sci 99(12):7821–7826
17. Grover A, Leskovec J (2016) node2vec: scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp 855–864
18. Grus J (2015) Data science from scratch: first principles with python. O'Reilly Media Inc, Sebastopol
19. Hosmer DW Jr, Lemeshow S, Sturdivant RX (2013) Applied logistic regression, vol 398. Wiley, Hoboken
20. Huang L, Wang CD, Chao HY (2019) Higher-order multi-layer community detection. In: Proceedings the thirty-third AAAI conference on artificial intelligence (AAAI-19). pp 9945–9946
21. Jiang JY, Li CT, Chen Y, Wang W (2018) Identifying users behind shared accounts in online streaming services. In: The 41st international ACM SIGIR conference on research and development in information retrieval, SIGIR '18. pp 65–74
22. Jin D, You X, Li W, He D, Cui P, Fogelman-Soulie F, Chakraborty T (2019) Incorporating network embedding into markov random field for better community detection. In: Proceedings the thirty-third AAAI conference on artificial intelligence (AAAI-19), pp 160–167
23. Johnson SC (1967) Hierarchical clustering schemes. Psychometrika 32(3):241–254
24. Jonnalagadda A, Kuppusamy L (2018) Overlapping community detection in social networks using coalitional games. Knowl Inf Syst 56(3):637–661
25. Lancichinetti A, Radicchi F, Ramasco JJ, Fortunato S (2011) Finding statistically significant communities in networks. PLoS ONE 6(4):e18961
26. Li CT, Lin YJ, Yeh MY (2018) Forecasting participants of information diffusion on social networks with its applications. Inf Sci 422:432–446
27. Li CT, Shan MK, Lin SD (2011) Context-based people search in labeled social networks. In: Proceedings of the 20th ACM international conference on information and knowledge management, CIKM '11, pp 1607–1612
28. Liben-Nowell D, Kleinberg J (2007) The link-prediction problem for social networks. J Assoc Inf Sci Technol 58(7):1019–1031
29. Lin CJ (2007) Projected gradient methods for nonnegative matrix factorization. Neural Comput 19:2756–2779
30. Liu P, Fu J, Dong Y, Qiu X, Cheung JCK (2019) Learning multi-task communication with message passing for sequence learning. In: Proceedings the thirty-third AAAI conference on artificial intelligence (AAAI-19), pp. 4360–4367
31. Liu X, Cheng H, Zhang Z (2019) Evaluation of community detection methods. IEEE Trans Knowl Data Eng, pp 1–1
32. Lobo JM, Jiménez-Valverde A, Real R (2008) AUC: a misleading measure of the performance of predictive distribution models. Glob Ecol Biogeogr 17(2):145–151
33. Lusseau D, Schneider K, Boisseau OJ, Haase P, Slooten E, Dawson SM (2003) The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. Behav Ecol Sociobiol 54(4):396–405
34. Palla G, Derényi I, Farkas I, Vicsek T (2005) Uncovering the overlapping community structure of complex networks in nature and society. arXiv:physics/0506133
35. Pan S, Wu J, Zhu X, Long G, Zhang C (2017) Task sensitive feature exploration and learning for multitask graph classification. IEEE Trans Cybern 47(3):744–758
36. Prat-Pérez A, Dominguez-Sal D, Larriba-Pey JL (2014) High quality, scalable and parallel community detection for large real graphs. In: Proceedings of the 23rd international conference on World wide web. ACM, pp 225–236
37. Radicchi F, Castellano C, Cecconi F, Loreto V, Parisi D (2004) Defining and identifying communities in networks. Proc Natl Acad Sci USA 101(9):2658–2663
38. Rendle S, Freudenthaler C, Gantner Z, Schmidt-Thieme L (2009) BPR: Bayesian personalized ranking from implicit feedback. In: Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence. AUAI Press, pp 452–461
39. Rosvall M, Bergstrom CT (2008) Maps of random walks on complex networks reveal community structure. Proc Natl Acad Sci 105(4):1118–1123

40. Smola AJ, Schölkopf B (2004) A tutorial on support vector regression. Stat Comput 14(3):199–222
41. Soundarajan S, Hopcroft J (2012) Using community information to improve the precision of link prediction methods. In: Proceedings of the 21st international conference on world wide web. ACM, pp 607–608
42. Tang J, Qu M, Wang M, Zhang M, Yan J, Mei Q (2015) Line: Large-scale information network embedding. In: Proceedings of the 24th international conference on world wide web. International world wide web conferences steering committee, pp 1067–1077
43. Tran P.V (2018) Learning to make predictions on graphs with autoencoders. In: 2018 IEEE 5th international conference on data science and advanced analytics (DSAA), pp 237–245
44. Vinh NX, Epps J, Bailey J (2010) Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance. J Mach Learn Res 11(Oct):2837–2854
45. Wu J, He J, Xu J (2019) Demo-net: degree-specific graph neural networks for node and graph classification. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery and data mining, KDD '19, pp 406–415
46. Xie J, Kelley S, Szymanski BK (2013) Overlapping community detection in networks: the state-of-the-art and comparative study. ACM Comput Surv (CSUR) 45(4):43
47. Yang J, Leskovec J (2013) Overlapping community detection at scale: a nonnegative matrix factorization approach. In: Proceedings of the sixth ACM international conference on web search and data mining. ACM, pp 587–596
48. Yang J, Leskovec J (2015) Defining and evaluating network communities based on ground-truth. Knowl Inf Syst 42(1):181–213
49. Zachary WW (1977) An information flow model for conflict and fission in small groups. J Anthropol Res 33(4):452–473

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Shu-Kai Zhang received the B.S. and the M.S. degree in Computer Science respectively from National Taiwan University, Taipei, Taiwan. His research interests include social network analysis, data science, and machine learning.



Cheng-Te Li is now an Associate Professor at Institute of Data Science and Department of Statistics, National Cheng Kung University (NCKU), Tainan, Taiwan. Prof. Li received his Ph.D. degree (2013) from Graduate Institute of Networking and Multimedia, National Taiwan University. Before joining NCKU, he was an Assistant Research Fellow (2014–2016) at CITI, Academia Sinica. Prof. Li's research interests target at Machine Learning, Deep Learning, Data Mining, Social Networks and Social Media Analysis, Recommender Systems, and Natural Language Processing. He had published a series of papers in top conferences, including, KDD, TheWebConf, SIGIR, CIKM, ACL, IJCAI, and ACM Multimedia. His academic recognitions include: 2019 K. T. Li Young Researcher Award, 2018 MOST Young Scholar Fellowship (The Columbus Program), and 2016 Exploration Research Award of Pan Wen Yuan Foundation.

Shou-De Lin is currently a full professor in the CSIE department of National Taiwan University. He holds a BS degree in EE department from National Taiwan University, an MS-EE degree from the University of Michigan, an MS degree in Computational Linguistics and PhD in Computer Science both from the University of Southern California. He leads the Machine Discovery and Social Network Mining Lab in NTU. Before joining NTU, he was a post-doctoral research fellow at the Los Alamos National Lab. Prof. Lin's research includes the areas of machine learning and data mining, social network analysis, and natural language processing. His international recognition includes the best paper award in IEEE Web Intelligent conference 2003, Google Research Award in 2007, Microsoft research award in 2008, 2015, 2016 merit paper award in TAAI 2010, 2014, 2016, best paper award in ASONAM 2011, US Aerospace AFOSR/AOARD research award winner for 5 years. He is the all-time winners in ACM KDD Cup, leading or co-leading the NTU team to win 5 championships. He also leads a team to win WSDM Cup 2016. He has served as the senior PC for SIGKDD and area chair for ACL. He is currently the associate editor for International Journal on Social Network Mining, Journal of Information Science and Engineering, and International Journal of Computational Linguistics and Chinese Language Processing. He is also a freelance writer for Scientific American.