User Identity Linkage Across Social Networks via Community Preserving Network Embedding

Xiaoyu Guo[1], Yan Liu[1(✉)], Lian Liu[2], Guangsheng Zhang[2], Jing Chen[1], and Yuan Zhao[1]

[1] State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450001, China
guoxy_ieu@outlook.com, ms_liuyan@aliyun.com, jingchen1101@yeah.net, zhao18703973381@163.com
[2] Investigation Technology Center PLCMC, Beijing 100000, China
donick@163.com, zhanggstide@163.com

Abstract. User Identity Linkage (UIL) across social networks refers to the recognition of the accounts belonging to the same individual among multiple social network platforms. Most existing network structure-based methods focus on extracting local structural proximity from the local context of nodes, but the inherent community structure of the social network is largely ignored. In this paper, with an awareness of labeled anchor nodes as supervised information, we propose a novel community structure-based algorithm for UIL, called CUIL. Firstly, inspired by the network embedding, CUIL considers both proximity structure and community structure of the social network simultaneously to capture the structural information conveyed by the original network as much as possible when learning the feature vectors of nodes in social networks. Given a set of labeled anchor nodes, CUIL then applies the back-propagation neural network to learn a stable cross-network mapping function for identities linkage. Experiments conducted on the real-world dataset show that CUIL outperforms the state-of-the-art network structure-based methods in terms of linking precision even with only a few labeled anchor nodes. CUIL is also shown to be efficient with low vector dimensionality and a small number of training iterations.

Keywords: User Identity Linkage · Community structure · Network embedding · Social network analysis

1 Introduction

Different social networks provide different types of services, people usually join multiple social networks simultaneously according to their needs of work or life [1]. Each user often has multiple separate accounts in different social networks. However, these accounts belonging to the same user are mostly isolated without any connection or correspondence to each other.

The typical aim of User Identity Linkage (UIL) is to detect that users from different social platforms are actually one and the same individual [2]. It is a crucial prerequisite

J. K. Liu and H. Cui (Eds.): ACISP 2020, LNCS 12248, pp. 621–630, 2020.
https://doi.org/10.1007/978-3-030-55304-3_32

for many interesting inter-network applications, such as friend recommendation across platforms, user behavior prediction, information dissemination across networks, etc.

Early research uses the public attributes and statistical features of users to solve the UIL problem [3, 4], such as username, user's hobbies, language patterns, etc. However, there is a lot of false information in the user's public attributes and user's statistics in different social networks are unbalanced. The correctness and richness of user's public attributes cannot be guaranteed.

Compared with user's attributes, the relationships between users are reliable and rich, and can also be directly used to solve the UIL problem. Therefore, the methods based on network structure are receiving more and more attention. Most of the existing methods [5–8] extract the local structural proximity from the context of nodes and focus on the microscopic structure of network. However, some typical properties of social network are ignored, such as community structure, etc.

Community structure is one of the most prominent features of social networks. A user primarily interacts with a part of the social network. Users in the same community are closely connected, but the connections among users from different communities are relatively sparse [9]. If a pair of friends connects closely to each other on Twitter and they exist in the same community because of common hobbies, then they should be closely connected and in the same community on Foursquare or Facebook.

In this paper, we introduce the community structure into user identity linkage across social networks and propose a novel model via community preserving network embedding, called CUIL. The contributions of this paper are as follows:

- CUIL applies network embedding and community structure to UIL problem simultaneously to retain the proximity structure and community structure to the vector representations of nodes; and learns a nonlinear mapping function between two networks through the BP neural network to achieve a unified model for UIL.
- We perform several experiments on a real-world dataset. The results show that CUIL can significantly improve the accuracy of user identity linkage compared to the state-of-the-art methods, e.g., up to 45% for *top-1* and more than 60% for *top-5* in terms of linking precision.

2 Preliminaries

2.1 Terminology Definition

We consider a set of social networks as $G^1, G^2, \ldots, G^n$, each of which is represented as an undirected and unweighted graph. Let $G = (V, E)$ represent the network, where V is the set of nodes, each representing a user, and E is the set of edges, each representing the relationship between two users.

In this paper, we take two social networks as an example, which are treated as source network, $G^s = (V^s, E^s)$, and target network, $G^t = \left(V^t, E^t\right)$ respectively. For ease of description, we have the following definitions.

Definition 1 (Anchor Link). *Link* (v_i^s, v_k^t) *is an anchor link between* G^s *and* G^t *iff.* $\left(v_i^s \in V^s\right) \wedge \left(v_k^t \in V^t\right) \wedge$ (v_i^s *and* v_k^t *are accounts owned by the same user in* G^s *and* G^t *respectively*).

Definition 2 (Anchor Users). *Users who are involved in two social networks simultaneously are defined as the anchor users (nodes) while the other users are non-anchor users (nodes).*

2.2 Problem Definition

Based on the definitions of the above terms, we formally define the problem of user identity linkage across social networks. The UIL problem is to determine whether a pair of accounts, $(v_i^s, v_k^t), v_i^s \in V^s, v_k^t \in V^t$, corresponds to the same real natural person, which can be formally defined as:

$$\Phi_V\left(v_i^s, v_k^t\right) = \begin{cases} 1 & v_i^s = v_k^t, \\ 0 & otherwise. \end{cases} \tag{1}$$

where $\Phi_V\left(v_i^s, v_k^t\right) = 1$ means v_i^s and v_k^t belong to the same individual.

3 CUIL: The Proposed Model

As shown in Fig. 1, CUIL consists of three main components: Cross Network Extension, Network Embedding, and BP Neural Network-based Mapping Learning, which will be introduced in detail later.

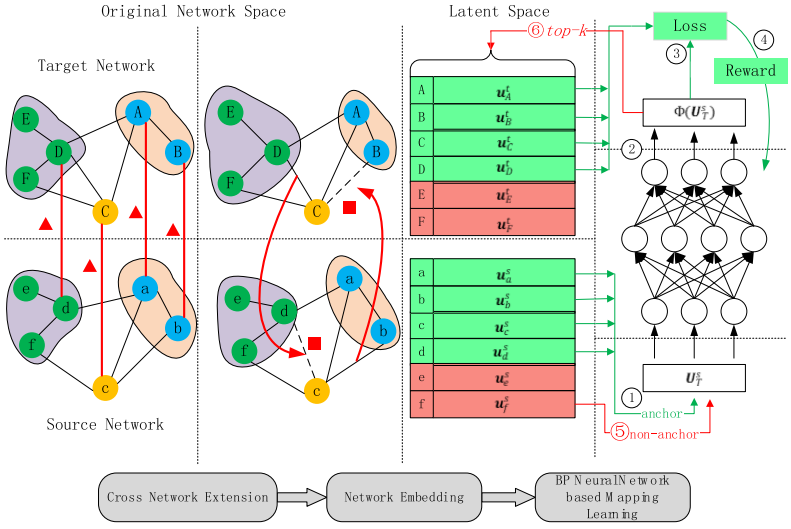


Fig. 1. The framework of CUIL. In the process of mapping learning, the training process is as ①–④, and the testing process is as ⑤⑥.

3.1 Cross Network Extension

For a real-world social network dataset, some edges that exist in practice may be unobserved, as they have not been explicitly built or failed to be crawled. These missing edges can lead to unreliable representations when embedding networks into latent vector spaces. In order to solve this problem, we apply *Cross Network Extension* to extend the source network and target network respectively according to the observed anchor links.

Usually, if two anchor nodes in the source network are connected, then their counterparts in the target network should also be connected [10]. Based on such an observation, we can perform *Cross Network Extension* by the following strategy. Given two social networks G^s, G^t, and a set of anchor links T, the extended network $\widetilde{G}^s = \left(\widetilde{V}^s, \widetilde{E}^s\right)$ can be described as:

$$\widetilde{V}^s = V^s \tag{2}$$

$$\widetilde{E}^s = E^s \cup \left\{\left(v_i^s, v_j^s\right) : (v_i^s, v_k^t) \in T, (v_j^s, v_l^t) \in T, (v_k^t, v_l^t) \in E^t\right\} \tag{3}$$

Similarly, the target network G^t is extended into $\widetilde{G}^t$.

3.2 Network Embedding

The first-order and second-order proximity describe social networks from the microscopic level, while the community structure constrains the network representation from a mesoscopic perspective. M-NMF [11] integrates the community structure into network embedding, which preserves both the first-order/second-order proximity structure and community structure of social networks. Here we use M-NMF model to learn the vector representation of nodes.

Modeling Community Structure. Modularity is a commonly used metric to measure the strength of network community structure [12]. If a network G is divided into two communities, the modularity is defined as:

$$Q = \frac{1}{4m}\sum\nolimits_{ij}\left(A_{ij} - \frac{k_i k_j}{2m}\right)h_i h_j \tag{4}$$

where $h_i = 1$ if node v_i belongs to the first community, otherwise, $h_i = -1$ and k_i is the degree of node v_i. And $\mathrm{m} = \frac{1}{2}\sum_i k_i$ is the number of relations in network G, $\frac{k_i k_j}{2m}$ is the expected number of edges between nodes v_i and v_j if edges are placed at random.

By defining the modularity matrix $\mathbf{B} = \left[B_{ij}\right] \in \mathbb{R}^{n*n}$, where $B_{ij} = A_{ij} - \frac{k_i k_j}{2m}$, then the modularity can be written as $\frac{1}{4m}\mathbf{h}^T\mathbf{B}\mathbf{h}$, where $\mathbf{h} = \left[h_{ij}\right] \in \mathbb{R}^n$ indicates the community to which each node belongs. When the network is divided into $K(K > 2)$ communities, the community membership indicator matrix $\mathbf{H} \in \mathbb{R}^{n*K}$ with one column for each community is introduced. In each row of $\mathbf{H}$, only one element is 1 and all the others are 0, so we have the constraint $tr\left(\mathbf{H}^T\mathbf{H}\right) = n$. Finally, we have:

$$Q = tr\left(\mathbf{H}^T\mathbf{B}\mathbf{H}\right), \quad s.t. \quad tr\left(\mathbf{H}^T\mathbf{H}\right) = n \tag{5}$$

where $tr(\mathbf{X})$ is the trace of matrix $\mathbf{X}$.

Modeling Proximity Structure. Modeling proximity structure mainly uses the first-order and second-order proximity. The first-order proximity indicates the similarity between two nodes connected directly and it is a direct expression of network structure. But in social networks, two nodes that have no direct connection do not mean there is no similarity. Therefore, in order to make full use of the proximity structure of social networks, the abundant second-order proximity is used to compensate for the sparse problem of first-order proximity.

The first-order proximity $\mathbf{S}^{(1)}$ is characterized by the adjacency matrix, then it can be defined as:

$$\mathbf{S}^{(1)} = [S_{ij}^{(1)}] \in \mathbb{R}^{n*n},\ s.t.\ S_{ij}^{(1)} = A_{ij} = 0\, or\, 1 \tag{6}$$

Let $N_i = \left(S_{i1}^{(1)}, \ldots, S_{in}^{(1)}\right)$, the i - th row of $\mathbf{S}^{(1)}$, be the first-order proximity between node v_i and other nodes. The second-order proximity $\mathbf{S}^{(2)}$ of a pair of nodes is the similarity between their neighborhood structures, which can be described as:

$$\mathbf{S}^{(2)} = [S_{ij}^{(2)}] \in \mathbb{R}^{n*n},\ s.t.\ S_{ij}^{(2)} = \frac{N_i * N_j}{\|N_i\| \left\|N_j\right\|} \in [0, 1] \tag{7}$$

Let similarity matrix $\mathbf{S} = \mathbf{S}^{(1)} + \eta \mathbf{S}^{(2)}$ to combine the first-order and second-order proximity together, where $\eta > 0$ is the weight of the second-order proximity. Using $\mathbf{U} \in \mathbb{R}^{n*d}$ to represent the node vector space, d is the dimensionality of representation, and introducing a nonnegative basis matrix $\mathbf{M} \in \mathbb{R}^{n*d}$, the objective function is described as:

$$\min \left\| \mathbf{S} - \mathbf{M}\mathbf{U}^T \right\|_F^2 \quad s.t. \quad \mathbf{M} \geq 0, \quad \mathbf{U} \geq 0 \tag{8}$$

The United Network Embedding Model. In order to integrate the proximity structure and community structure in a unified framework, the community representation matrix $\mathbf{C} \in \mathbb{R}^{K*d}$ is introduced, where the r - th row $\mathbf{C}_r$ corresponding to the community r. If node v_i belongs to community r, formulated as $\mathbf{U}_i\mathbf{C}_r$, then the representation of v_i should be highly similar to that community r. As the community indicator matrix $\mathbf{H}$ offers a guide for all the nodes, $\mathbf{U}\mathbf{C}^T$ is expected to be as closely consistent as possible with $\mathbf{H}$. Then the overall objective function is described as:

$$\begin{aligned} &\underset{\mathbf{M},\mathbf{U},\mathbf{H},\mathbf{C}}{\text{Min}} \|\mathbf{S} - \mathbf{M}\mathbf{U}^T\|_F^2 \ + \ \alpha \|\mathbf{H} - \mathbf{U}\mathbf{C}^T\|_F^2 - \beta tr\left(\mathbf{H}^T\mathbf{B}\mathbf{H}\right), \\ &s.t. \quad \mathbf{M} \geq 0, \mathbf{U} \geq 0, \mathbf{H} \geq 0, \mathbf{C} \geq 0, tr\left(\mathbf{H}^T\mathbf{B}\mathbf{H}\right) = n, \alpha > 0, \beta > 0 \end{aligned} \tag{9}$$

3.3 BP Neural Network-Based Mapping Learning

After obtaining the latent vector space of each social network, CUIL applies the BP neural network (BPNN) to learn the mapping function Φ from G^s to G^t. Given any pair of anchor nodes (v_i^s, v_k^t) and their vector representations $(\boldsymbol{u}_i^s, \boldsymbol{u}_k^t)$, we firstly use the

mapping function $\Phi(\boldsymbol{u}_i^s)$ map node vector $\boldsymbol{u}_i^s$ to another vector space, and then minimize the distance between $\Phi(\boldsymbol{u}_i^s)$ and $\boldsymbol{u}_k^t$. In this paper, the *Cosine Distance* is selected and the loss function can be formally described as:

$$\ell\left(\boldsymbol{u}_i^s, \boldsymbol{u}_k^t\right) = 1 - \cos\left(\Phi(\boldsymbol{u}_i^s), \boldsymbol{u}_k^t\right) \tag{10}$$

The set of known anchor links is T, and the sub-vector spaces composed of anchor nodes are $\mathbf{U}_T^s \in \mathbb{R}^{|T|\times d}$ and $\mathbf{U}_T^t \in \mathbb{R}^{|T|\times d}$ respectively. Then the objective function of the mapping learning can be formally described as:

$$\ell\left(\mathbf{U}_T^s, \mathbf{U}_T^t\right) = \arg\min_{\mathbf{W},\mathbf{b}}\left(1 - \cos\left(\Phi\left(\mathbf{U}_T^s\right), \mathbf{U}_T^t\right); \mathbf{W}, \mathbf{b}\right) \tag{11}$$

where $\mathbf{W}$ and $\mathbf{b}$ are the weight parameters and bias parameters obtained by the back-propagation algorithm respectively. We minimize the loss function by stochastic gradient descent algorithm using the known anchor links as supervised information.

Construct the *top - k* for non-anchor nodes. For a non-anchor node v_x^s in the source network, firstly we input its vector representation $\boldsymbol{u}_x^s$ into the BPNN model trained above and get the mapping vector $\Phi(\boldsymbol{u}_x^s)$, like ⑤ in Fig. 1. Then we find k nodes that are most similar to the mapping vector $\Phi(\boldsymbol{u}_x^s)$ from the target network to form the *top - k* of node v_x^s, like ⑥ in Fig. 1.

4 Experiments

4.1 Datasets, Baselines and Parameter Setup, and Evaluation Metrics

Datasets. The real-world dataset is provided by [7], which contains two social networks, Twitter and Foursquare. Table 1 summarizes the statistics of this dataset.

Table 1. Statistics of twitter-foursquare dataset.

Networks	#Users	#Relations	#Anchor users
Twitter	5120	164919	1609
Foursquare	5313	76792	

Baselines and Parameter Setup. The model we proposed in this paper is based on network structure, so we compare CUIL with several structure-based methods for UIL.

- **PALE:** Predicting Anchor Links via Embedding [6] employs network embedding to capture the major and specific structural regularities and further learns a stable cross-network mapping for predicting anchor links.
- **IONE:** Input Output Network Embedding [7] tries to model followers/followees as different context vectors. With hard/soft constraints of anchor users, IONE learns a unified vector space by preserving second-order structural proximity.

- **DeepLink:** A Deep Learning Approach for User Identity Linkage [8] samples networks by random walks and learns to encode network nodes into vector representations to capture the local and global network structures. Finally, a deep neural network model is trained through the dual learning to realize user identity linkage.
- **PUIL:** Proximity Structure-based User Identity Linkage (PUIL) is based only on the proximity structure while without considering community structure.

Parameter Setup. The baselines are implemented according to the original papers. For CUIL (PUIL), we employ a four-layer neural network (2 hidden layers) to capture the non-linear mapping function between the source and target networks: 500 d (first hidden layer), 800 d (second hidden layer) and 300 d (input and output layer). The learning rate for training is 0.001, and the batch size is set to 16.

Evaluation Metrics. Inspired by *the Success at rank k* proposed in [13], we use $Precision@k (P@k)$ as the evaluation metric of user identity linkage.

$$P@k = \sum_i^n \mathbb{I}_i\{success@k\} / n \tag{12}$$

where n is the number of testing anchor nodes and $\mathbb{I}_i\{success@k\}$ measures whether the counterpart of v_i^s exists in top - $k (k \leq n)$.

4.2 Experiments

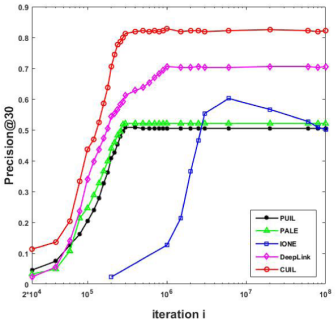
We firstly evaluate the influence of the parameters on the performance of algorithms, such as the training iteration i, the percentage r of anchor nodes used for training, and the vector dimensionality d. We set the basic experimental environment as: r is 0.8, i is 1 million, and d is 800. We change one parameter at a time while keeping the other two parameters constant.

As can be seen from Fig. 2(a), there is no overfitting problem for CUIL compared to IONE. By comparison, CUIL can not only get better results, but also reach the convergence faster.

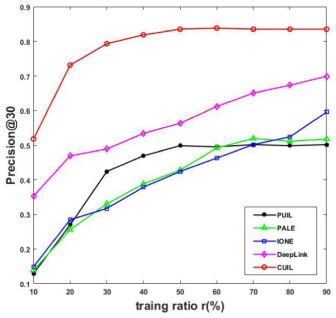
The percentage r of anchor nodes used for training is an important parameter. As shown in Fig. 2(b), with the increase of training ratio r from 0.1 to 0.9, the performance of CUIL is always superior to other baselines. CUIL performs excellently even though the training ratio r is only 0.1 or 0.2.

The impact of the vector dimensionality d on the results is shown in Fig. 2(c). IONE, DeepLink, and CUIL all perform well on low-dimensional vector spaces. When the dimensionality is below 100, DeepLink performs best. But when the dimensionality reaches up to 200, the performance of CUIL is significantly better than other methods.

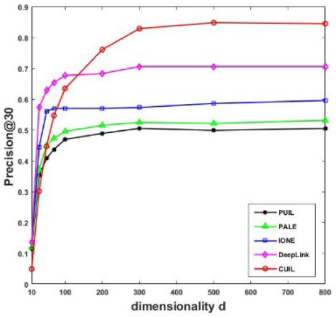
Finally, we conduct experiments for each method with the most appropriate parameters: the training ratio r is 0.8 and the vector dimensionality d is 300. The training iteration i is 3 hundred thousand for CUIL (PUIL) and PALE, 1 million for DeepLink, and 6 million for IONE. And we randomly select 6 different k values between 0 and 30 to compare the performance of different algorithms, as illustrated in Table 2. In order to compare and analyze the results intuitively, we show the results in a line chart, as shown in Fig. 2(d).



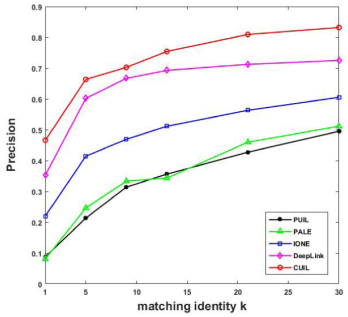
(a) Precision@30 on different *iteration*



(b) Precision@30 on different *ratio*



(c) Precision@30 on different *dimensionality*



(d) Precision on different *k*

Fig. 2. Result analysis on twitter-foursquare dataset.

Table 2. Comparisons of user identity linkage on twitter-foursquare dataset.

P@k	Precision					
	P@1	*P@5*	*P@9*	*P@13*	*P@21*	*P@30*
PALE	0.0906	0.2848	0.3625	0.3981	0.4628	0.5178
PUIL	0.0874	0.2136	0.3139	0.3592	0.4272	0.4984
IONE	0.2201	0.4142	0.4692	0.5113	0.5631	0.6052
DeepLink	0.3526	0.6019	0.6667	0.6926	0.7120	0.7249
CUIL	**0.4660**	**0.6634**	**0.7023**	**0.7540**	**0.8091**	**0.8317**

4.3 Discussions

With the experiments on the twitter-foursquare dataset, we have the following discussions:

- Through horizontal comparisons, CUIL proposed in this paper outperforms PALE, IONE, and DeepLink, even the $P@1$ can reach more than 45%. And through longitudinal comparisons, CUIL performs better than PUIL which only uses the proximity structure.
- The percentage of anchor nodes used for training greatly affects the performance of all algorithms, while CUIL achieves much better than other baselines even with only a few labeled anchor nodes. It is well known that the number of known anchor nodes is very limited and difficult to obtain. Therefore, our method is more advantageous in the practical applications.
- When the dimensionality reaches up to 200, the performance of CUIL has a significant improvement. With the rapid development of computing power and the continuous optimization of machine learning algorithms, the vector dimensionality is no longer a hard problem that restricts the performance of algorithm. In order to get better results, it is acceptable that the vector dimensionality reaches 200 or more for CUIL.

5 Conclusion

In this paper, we studied the problem of user identity linkage across social networks and proposed a novel community structure-based method, called CUIL. Many previous studies extracted the proximity structure of social networks from the local content of nodes while ignoring the important community structure. Therefore, we introduced the community structure and network embedding to UIL problem simultaneously. CUIL applied the embedding method, which preserves the microscopic proximity structure and the mesoscopic community structure, to map the original social network space into the vector space. Then based on the labeled anchor nodes, CUIL employed BP neural network to learn a stable mapping across different social networks. We conducted extensive experiments on the real-world dataset and the results showed that CUIL achieved superior performance over the state-of-the-art baseline methods that are based on the network structure.

Acknowledgements. This work was supported by the National Natural Science Foundation of China (U1636219, 61602508, 61772549, U1736214, 61572052, U1804263, 61872448) and Plan for Scientific Innovation Talent of Henan Province (No. 2018JR0018).

References

1. Zhang, J., Yu, P., Zhou, Z.: Meta-path based multi-network collective link prediction. In: The 20th International Conference on Knowledge Discovery and Data, pp. 1286–1295. ACM (2014)
2. Shu, K., Wang, S., Tang, J., Zafarani, R., Liu, H.: User identity linkage across online social networks: a review. In: SIGKDD Explorations Newsletter, pp. 5–17. ACM (2017)
3. Liu, J., Zhang, F., Song, X., Song, Y., Lin, C., Hon, H.: What's in a name? An unsupervised approach to link users across communities. In: The 6th International Conference on Web Search Data Mining, pp. 495–504. ACM (2013)

4. Zafarani, R., Liu, H.: Connecting users across social media sites: a behavioral-modeling approach. In: The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 41–49. ACM (2013)
5. Wang, C., Zhao, Z., Wang, Y., Qin, D., Luo, X., Qin, T.: DeepMatching: a structural seed identification framework for social network alignment. In: The 38th International Conference on Distributed Computing Systems, pp. 600–610. IEEE (2018)
6. Man, T., Shen, H., Liu, S., Jin, X., Cheng, X.: Predict anchor links across social networks via an embedding approach. In: The 25th International Joint Conference on Artificial Intelligence, pp. 1823–1829. IJCAI (2016)
7. Liu, L., Cheung, W., Li, X., Liao, L.: Aligning users across social networks using network embedding. In: The 25th International Joint Conference on Artificial Intelligence, pp. 1774–1780. IJCAI (2016)
8. Zhou, F., Liu, L., Zhang, K., Trajcevski, G., Wu J., Zhong, T.: DeepLink: a deep learning approach for user identity linkage. In: INFOCOM, pp. 1313–1321. IEEE (2018)
9. Girvan, M., Newman, M.: Community structure in social and biological networks. Proc. Natl. Acad. Sci. U.S.A. **99**(12), 7821–7826 (2002)
10. Bayati, M., Gerritsen, M., Gleich, D., Saberi, A., Wang, Y.: Algorithms for large, sparse network alignment problems. In: ICDM, pp. 705–710. IEEE (2009)
11. Wang, X., Cui, P., Wang, J., Pei, J., Zhu, W., Yang, S.: Community preserving network embedding. In: The 31st AAAI, pp. 203–209. AAAI (2017)
12. Newman, M.: Modularity and community structure in networks. Proc. Natl. Acad. Sci. **103**(23), 8577–8582 (2006)
13. Iofciu, T., Fankhauser, P., Abel, F., Bischoff, K.: Identifying users across social tagging systems. In: 5th International AAAI Conference on Weblogs and Social Media, pp. 522–525. ACM (2011)