Social Relation Inference via Label Propagation

Anonymous

Anonymous

Abstract. Collaboration networks are a ubiquitous way to characterize the interactions between people. In this paper, we consider the problem of inferring social relations in collaboration networks, such as the fields that researchers collaborate in, or the categories of projects that Github users work on together. Social relation inference can be formalized as a multi-label classification problem on graph edges, but many popular algorithms for semi-supervised learning on graphs only operate on the nodes of a graph. To bridge this gap, we propose a principled method which leverages the natural homophily present in collaboration networks. First, observing that the fields of collaboration for two people are usually at the intersection of their interests, we transform an edge labeling into node labels. Second, we use a label propagation algorithm to propagate node labels in the entire graph. Once the label distribution for all nodes has been obtained, we can easily infer the label distribution for all edges. Experiments on two large-scale collaboration networks demonstrate that our method outperforms the state-of-the-art methods for social relation inference by a large margin, in addition to running several orders of magnitude faster.

Keywords: label propagation, social relation inference, social network

1 Introduction

In collaboration networks, edges, or social relations [12], are formed between people with shared interests. Social relations in networks are complex and nuanced, which often cannot be characterized by a single label. Consider a co-author network between researchers where the social relations between two researchers are the research areas they collaborate in. Since collaborations can occur in different research areas, the social relation between researchers is inherently multifaceted. Many applications on collaboration networks can benefit from an awareness of social relations, such as node classification [15], recommendation [11] and anomaly detection [14]. However, in many networks, such label information (social relations) is far from complete. It is thus desirable to learn to infer social relations associated with the unlabeled edges.

We formalize the task of social relation inference as a semi-supervised multi-label edge classification problem on networks. Given the network structure and a limited amount of labeled edges, our goal is to infer the labels of the rest of the edges. There are several previous studies on inferring social ties from social networks, which is similar to our definition of social relations [11, 9]. However, these works assume that each edge corresponds to a single relation type, which may not be the case in collaboration networks. Moreover, they only consider first-order or second-order relationships between nodes, but fails to model higher-order relationships that play an important role in network inference tasks [2].

Another relevant area is network embeddings [6, 8, 4], which aim at learning low-dimensional latent representations of nodes in a network. Also, representations of larger-scale components of networks (such as edges and subgraphs) can be composed from these node representations. These representations can then be used as features for a wide range of downstream tasks on networks, including social relation inference. As a pioneering work, DeepWalk [6] generates fixed-length random walk sequences in networks and trains a skip-gram model [5] on these sequences to obtain node embeddings. While achieving state-of-the-art results on a handful of network inference tasks such as node classification and link prediction [6, 4], the semantics of edges in networks are seldom exploited by network embedding models. Moreover, we find that they usually ignore the unique properties possessed by different types of networks and by different downstream tasks. Also, many of them are computationally expensive: learning network embeddings of a one-million node network can take several days on a single CPU.

In this paper, we propose a simple but effective method for social relation inference on collaboration networks. Our method is based on the observation that social relations between people in collaboration networks are determined by their shared interests. As such, the networks are highly homophilous and there is a natural connection between the (hidden) labels of the nodes, and the provided edge labels. Using this relationship, we first transform the edge labels into a node labeling. Next, to alleviate any data sparsity problem, we perform label propagation on the input network to obtain label distribution for all nodes. Label propagation [16, 13] represents a class of semi-supervised learning methods which find numerous applications in graph mining. For social relation inference, we find that label propagation has several desirable properties compared to the neural methods mentioned before: it is extremely efficient and it makes good use of the high level of homophily exhibited in collaboration networks [7]. Finally, once node labels have been obtained, the label distribution of edges can be easily inferred from the label distribution of their endpoints. Experimental results on real-world networks show that our method outperforms state-of-the-art methods by a large margin.

2 Problem Definition and Notation

We hereby formalize the problem of social relation inference in collaboration networks. Let $G = (V, E)$ be an undirected graph, where V are the nodes in the graph and E represent its edges. Let A be the adjacency matrix of G. Let $L = (l_1, l_2, \cdots, l_k)$ be the set of relation types (labels). A partially labeled network is then defined as $G = (V, E_L, E_U, Y_L)$, where E_L is the set of labeled edges, E_U is the set of unlabeled edges with $E_L \cup E_U = E$. Y_L represents the relation types associated with the labeled edges in E_L, with $\forall Y_L(i) \in Y_L : Y_L(i) \subseteq L$.

Algorithm 1 LabelProp(G, P)

```
Input: graph $G$, initial node label distribution $P$, rounds of iteration $k$
Output: node label distribution after propagation $\hat{Y_V} \in \mathbb{R}^{|V|\times|L|}$
Compute the degree matrix $D$: $D_{ii} \leftarrow \sum_j A_{ij}$
Compute the transition matrix: $Q \leftarrow D^{-1}A$
$Y^{(0)} \leftarrow P$
for $i = 0$ to $k-1$ do
    $Y^{(i+1)} \leftarrow QY^{(i)}$
end for
$\hat{Y_V} = Y^{(k)}$
return $\hat{Y_V}$
```

The objective of social relation inference is to predict the relation types Y_U of the unlabeled edges E_U:

$$f : G = (V, E_L, E_U, Y_L) \rightarrow Y_U \tag{1}$$

We denote the i-th row and ij-th element of a matrix M as M_i and M_{ij}.

3 Method

3.1 Step 1: From Edge Labels to Node Labels

One challenge with social relation inference is that the labels we seek to predict are associated with edges, instead of nodes. However, most machine learning algorithms on graphs only operate on nodes. To bridge this gap, we note that collaboration networks possess a unique property: edges are typically formed between two people which have *shared interests.* Such shared interests can very well be characterized by the labels of edges. This means that we should be able to infer the latent interests of nodes based on their corresponding edge labels.

Formally, we seek to estimate the probability distribution matrix $P \in \mathbb{R}^{|V|\times|L|}$ for all nodes over the label space L. For ease of presentation, we assume that the training data is given in the form of triplets $t = (u, v, l)$, where $u, v \in V, l \in L$. In other words, if an edge has several labels, then we construct one triplet for each label. We define the set of all training triplets as T. Assume the label distribution of u and v are independent, the strength of relation l between u and v can be estimated as:

$$Pr(l|u, v) = P_{ul} \cdot P_{vl} \tag{2}$$

Our objective is to maximize the probability of observing the relations in T as given by:

$$\ell = \prod_{u \in V} \prod_{\substack{(v,l) \\ (u,v,l)\in T}} Pr(l|u, v) \tag{3}$$

Then, for a certain $u \in V$, our goal is to minimize the following objective:

$$-\log \ell_u = - \sum_{\substack{(v,l) \\ (u,v,l) \in T}} (\log P_{ul} + \log P_{vl}) \tag{4}$$

Since P is the probability distribution of labels, we have the constraint $\sum_{l \in L} P_{ul} = 1$. The Lagrangian function of Eq. (4) is:

$$\mathcal{L}(P_u, \lambda) = - \sum_{\substack{(v,l) \\ (u,v,l) \in T}} (\log P_{ul} + \log P_{vl}) + \lambda(\sum_{l \in L} P_{ul} - 1) \tag{5}$$

For all $l \in L$, we take the derivative of Eq. 5 w.r.t. P_{ul} and set it to zero:

$$-\frac{\#(u,l)}{P_{ul}} + \lambda = 0 \tag{6}$$

where $\#(u,l)$ is the number of co-occurrences of u and l in T, with v being marginalized out. It is now clear that $P_{ul} = \frac{\#(u,l)}{\lambda}$. Combined with the constraint $\sum_{l \in L} P_{ul} = 1$, we have $\lambda = \sum_{l \in L} \#(u,l)$. Finally, the closed-form estimation of P_{ul} is calculated as: $P_{ul} = \#(u,l) / \sum_{l \in L} \#(u,l)$.

Concretely, we can simply compute the relative frequency that each node co-occur with each label, which gives us the initial label distribution P of all nodes.

3.2 Step 2: Label Propagation

Labeled edges are often scarce in real-world collaboration networks. As a result, using the procedure outlined above, we may get an empty label distribution for most of the nodes (as they have no edges). To alleviate this problem, we propose using label propagation [16] on G to spread the information from labeled edges around the graph. Algorithm 1 details the process. We start from the initial label distribution obtained in Step 1 and repeatedly distribute node labels to the neighboring nodes.

3.3 Step 3: From Node Labels to Edge Labels

Once we have obtained the label distribution for all nodes, we can easily compute the label distribution for edges by reusing Eq. 2. For each edge $e = (u, v)$, the strength of relation l is $P_{ul} \cdot P_{vl}$. The ranking of relation strengths serves as our prediction of social relations.

3.4 Time Complexity Analysis

The majority of time complexity is contributed by Algorithm 1, which takes $O(k \cdot (|E| + |V| \cdot |L|))$. In our experiments, it is further shown that a small value of k is sufficient for our model to converge: empirically, we take $k = 5$. We provide detailed running time comparison against baseline methods in Section 4.

4 Experiment

In this section, we describe the datasets for social relation inference and compare our method against a number of baselines.

4.1 Dataset

Table 1. Statistics of the networks used in our experiments.

Dataset	# Vertices	# Edges	# Train	# Test	# Valid	# Classes
Arnet-Small	187,939	1,619,278	1,579,278	20,000	20,000	100
Arnet-Medium	268,037	2,747,386	2,147386	300,000	300,000	500
Arnet-Large	945,589	5,056,050	3,856,050	600,000	600,000	500

We use the processed ArnetMiner [10] datasets provided by TransNet [12]. ArnetMiner is a large-scale co-author network with over a million authors and four million collaboration relations. The social relations between researchers can be reflected by the research areas or topics they collaborate in. Concretely, for each co-author relationship, the authors of TransNet extract representative research interest phrases from the abstracts of co-authored papers as edge labels. Two collaboration networks of different scales and different amount of labels are provided in this dataset to better investigate the characteristics of different models. The statistics of the datasets are presented in Table 1.

4.2 Baseline Methods

The baseline methods we use are as follows: **(1) DeepWalk** [6]: This is a network embedding method that learns latent representations of nodes in a graph. **(2) LINE** [8]: This is a network embedding method that preserves both first-order and second-order proximities in networks. **(3) node2vec** [4]: This is a network embedding method that improves DeepWalk with a biased random walk phase. **(4) TransE** [1]: This is a knowledge base embedding method which simultaneously learns latent representations of nodes and relations. Since TransE models each relation separately, we split each edge with k labels into k training instances, one for each label. **(5) TransNet** [12]: This method is an extension to TransE which explicitly models edges with multiple labels. It is also the state-of-the-art method for social relation inference.

We follow the experimental setup as in TransNet [12]. For all baseline methods, we use the hyperparameter settings as described in their papers. For TransE, we use the similarity-based method to predict social relations as described in [1]. For TransNet, we follow the inference algorithm in their paper. For the three network embedding methods, we concatenate node representations as the feature vector for edges. For social relation inference, we train a one-vs-rest logistic regression model with L2 regularization implemented in LibLinear [3].

Table 2. Relation inference results on **Arnet-Small**.

Algorithm	**Metrics(%)**		
	hits@1	*hits*@5	*hits*@10
DeepWalk	13.88	36.80	50.57
LINE	11.30	31.70	44.51
node2vec	13.63	36.60	50.27
TransE	39.16	78.48	88.54
TransNet	47.67	86.54	92.27
Proposed	**48.89**	**90.13**	**93.90**

Table 3. Relation inference results on **Arnet-Large**.

Algorithm	**Metrics(%)**		
	hits@1	*hits*@5	*hits*@10
DeepWalk	5.41	16.17	23.33
LINE	4.28	13.44	19.85
node2vec	5.39	16.23	23.47
TransE	15.38	41.87	55.54
TransNet	28.85	66.15	75.55
Proposed	**29.91**	**72.32**	**80.86**

4.3 Results and Analysis

In Tables 2 and 3, we summarize the experimental results using the same data split as TransNet. Results for all baseline methods (including TransNet) are taken from the TransNet paper. We can clearly see that our simple method outperforms all baseline methods by a large margin. The performance gain over the best baseline method, TransNet, is at least 3.5% and up to 8.4% in terms of hits@5. We note that the TransNet data split uses 98%, 76% and 78% edges as training data for Arnet-Small, Arnet-Medium and Arnet-Large respectively. With such a large amount of training data, our algorithm achieves the reported performance even without performing label propagation, which proves the effectiveness of the node label inference algorithm. Moreover, our algorithm is orders of magnitude faster than all baseline methods. Using a single CPU core at 2.0GHz, our method finishes in 5 minutes on Arnet-Small while all baseline methods take more than 24 hours.

The only hyperparameter in our algorithm is the number of rounds of iterations k for label propagation, which is tuned on the validation set. We observe that even with only 1% of labeled edges, our label propagation algorithm converges within five iterations.

5 Conclusion

We study the problem of inferring social relations in collaboration networks, formulated as a semi-supervised learning problem on graphs where edges have multiple labels. Observing that edges in collaboration networks represent the shared interests of two people, we transform edge labels to node labels and perform label propagation to deal with the label sparsity problem. Experimental results on real-world collaboration networks show the superiority of our method in terms of both accuracy and efficiency.

References

1. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Advances in neural information processing systems. pp. 2787–2795 (2013)
2. Cao, S., Lu, W., Xu, Q.: Grarep: Learning graph representations with global structural information. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. pp. 891–900. ACM (2015)
3. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: Liblinear: A library for large linear classification. The Journal of Machine Learning Research **9**, 1871–1874 (2008)
4. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2016)
5. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. pp. 3111–3119 (2013)
6. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: Online learning of social representations. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 701–710. ACM (2014)
7. Powell, W.W., White, D.R., Koput, K.W., Owen-Smith, J.: Network dynamics and field evolution: The growth of interorganizational collaboration in the life sciences. American journal of sociology **110**(4), 1132–1205 (2005)
8. Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q.: Line: Large-scale information network embedding. In: Proceedings of the 24th International Conference on World Wide Web. pp. 1067–1077. International World Wide Web Conferences Steering Committee (2015)
9. Tang, J., Lou, T., Kleinberg, J.: Inferring social ties across heterogenous networks. In: Proceedings of the fifth ACM international conference on Web search and data mining. pp. 743–752. ACM (2012)
10. Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., Su, Z.: Arnetminer: extraction and mining of academic social networks. In: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 990–998. ACM (2008)
11. Tang, W., Zhuang, H., Tang, J.: Learning to infer social ties in large networks. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 381–397. Springer (2011)
12. Tu, C., Zhang, Z., Liu, Z., Sun, M.: Transnet: translation-based network representation learning for social relation extraction. In: Proceedings of International Joint Conference on Artificial Intelligence (IJCAI), Melbourne (2017)
13. Wang, F., Zhang, C.: Label propagation through linear neighborhoods. IEEE Transactions on Knowledge and Data Engineering **20**(1), 55–67 (2008)
14. Xiang, B., Liu, Z., Zhou, J., Li, X.: Feature propagation on graph: A new perspective to graph representation learning. arXiv preprint arXiv:1804.06111 (2018)
15. Xu, L., Wei, X., Cao, J., Philip, S.Y.: On exploring semantic meanings of links for embedding social networks (2018)
16. Zhu, X., Ghahramani, Z.: Learning from labeled and unlabeled data with label propagation (2002)