NMOEA: Node Embedding and Multi-Objective Evolutionary Algorithms for Co-authorship Prediction

Heeju Wi
School of Computing, KAIST
Daejeon, South Korea
bb0711@kaist.ac.kr

Junho Han
Graduate School of AI, KAIST
Daejeon, South Korea
jyuno426@kaist.ac.kr

ABSTRACT

In this work, node embedding and multi-objective evolutionary algorithms(MOEA) are combined to explicitly construct promising communities for co-authorship prediction. Both node embedding and evolutionary algorithms have been used in previous literature for community detection and co-author prediction, but works to combine them are limited. We utilize *node2vec* to get two features: homophily and structural equivalence in the network. Then use them to define two conflicting fitness values: (1)local similarity (2)structural role dissimilarity. We hypothesize that those attributes are important, especially in a large low-order network to predict future co-authorship. MOEA finds promising communities maximizing two objectives with sampling heuristics to reduce computation cost in a large network. Then nonlinear SVM finally performs binary classification with features from those communities. Our model NMOEA achieves accuracy 59.9% which is better than the baseline accuracy 50.0% (random guess).

1. INTRODUCTION

Co-authorship prediction task has been largely studied in the field of data mining, but there are few works that consider distributional difference in co-author networks. The most of existing works[1, 2, 9] are based on complex networks, where the order of a paper(the number of co-authors) is large and diverse. However, in the dataset like one given in this project, the order of paper is dominantly 2, 3 or 4 so that previous approaches may not work well. For example, Yoon et al.[9] claimed that capturing interactions between higher order author-sets is important to predict future co-authorship, however we observed that it is not helpful on this low-order network. We summarized our observations on it in Appendix sections, so please see them for more details.

Our main question is how to predict future co-authorship in a large but low-order co-author network effectively. For this, we hypothesized that local features (such as research field or affiliation) and structural role features (such as student or supervisor) are much more important. It is because there are limited direct connection between authors in low-order network dataset as we've shown in Appendix. Those features can be computed using *node2vec*[4], also called as node embedding.

So, to leverage both features, we devise two fitness functions (1)local similarity and (2)structural role dissimilarity by using two node embeddings. In particular, our objective here is to separate all authors into explicit communities while local similarity in a single community is maximized and structural role dissimilarity between communities is also maximized. What we can expect from this is to identify communities that are not only similar in local nature (research field or affiliation) but also diverse in combination of authors' roles. The resulting communities would have expressive power to distinguish true/false future co-authorship prediction. Indeed, those fitness can be conflicting each other, multi-objective optimization would be useful. So, our method NMOEA combines node embedding and multi-objective evolutionary algorithm. Up to our knowledge, there are no prior works that incorporate intrinsic node features in MOEA setting.

The contributions of this project are the following:

- We proposed new method NMOEA for detecting communities that uses learned intrinsic node embedding in MOEA setting.
- We also showed such discovered communities have expressive power to distinguish true/false co-authorship prediction only with simple SVM.
- The proposed method outperforms the baseline (random guess) for co-authorship prediction task in a large low-order network with tractable computation cost.

2. PROPOSED METHOD

2.1 Node embedding

Node embedding method can learn rich feature representations for nodes in a network[4]. Without any given features of the authors, such as age, college or jobs, we can build intrinsic feature vector for each node. Moreover, by tuning some parameters in building node embedding, we can build several kind of feature vectors. In this project, we utilize two feature vectors; one for representing relationship with its neighbors and the other for the structural characteristics in the global graph structure.

.

From the point of view of the graph structure, the n nodes of the arbitrary graph G are in one-to-one correspondence with the n nodes of the arbitrary graph G', and the edges of $G(i, j)$ are connected to the path between node i and node j of G'. It means that the path between the corresponding nodes is mapped. In this case, G' is mapped so that the path between deeply related nodes is shortened by considering characteristics such as connectivity between nodes in the graph G and structural equivalence.

Given the co-author data, we can build a graph; an author is a node and there's an edge between two nodes if they co-worked together. By using this graph, we can compute two feature vectors for each node by adjusting parameters of *node2vec*; p : return hyper parameter, q : inout hyper parameter. One for the homophily, which means the similarity between nearby node, and the other for role in the global structure. We use $p = 1, q = 0.5$ and $p = 1, q = 2$ two build them.

2.2 Multi-Objective Evolutionary Algorithm

Multi-objective evolutionary algorithms(MOEA) are good at handling conflicting objectives [3, 5, 13] and have been used to handle attributed network clustering problems in very recent years[6, 7, 8, 11, 12]. Moreover, MOEA aims at finding a set of Pareto optimal solutions in a single run, which can provide a wide range of options for decision makers. The Pareto optimal solution is the generally used definition of the optimal solution in multi-objective optimization problems[3, 5, 13]. Pareto dominance is a criterion for evaluating two feasible solutions. x_1 dominates x_2 only when x_1 is not worse than x_2 in all O objectives, and x_1 is better than x_2 at least in one objective. Pareto optimality means there are no solutions better than the target solution. Multi-objective optimization problems usually have more than one Pareto optimal solution, and the set of solutions are named as Pareto set.

We use representation, initial population, crossover and mutation operator proposed in [10]. Our main contribution is to devise novel fitness functions with node embedding and to combine them with MOEA. For MOEA, we use NSGA-II framework. The pseudo code of *NMOEA* is in algorithm 1. In *fast-non-dominated-sort*, it uses the two fitness functions to sort population. This algorithm returns pareto front, which is a set of optimal solutions. Each solution is a set of separated communities of the given graph.

Algorithm 1 NMOEA

```
Input: given co-author dataset
Output: pareto front partitions
graph ← $makeGraph(data)$
S, D← $node2vec(graph)$ (two fitness functions)
Population ← $initialPopulation(graph)$
while $iter < MaxIter$ do
    while $iter2 < 0.5 * population\ size$ do
        $p_i, p_j = random$(Population) $(i < j)$
        Population ←(append) $Crossover(p_i, p_j)$
        Population ←(append) $Mutation(p_i)$
    Population ←fast-non-dominated-sort(Population)
    Population ← choose top-$k$ of Population
    ($k$ =population size)
return getParetoFront(Population)
```

2.2.1 Representation

The representation of each individual is a crucial part in evolutionary algorithm. Each individual is defined as a partition of all nodes. A partition can be thought as a collection of separated communities including nodes. So, each individual is represented as an array of the community number in which each node is contained. This is known as **character string representation**. So, its objective is to find good partitions that explicitly separated authors.

2.2.2 Fitness function

For the fitness function, we use two functions; (1)sum of local similarity in every community (S) and (2)structural role dissimilarity between all communities(D). By using the intrinsic homophilic feature vector(h_i) of each node from node embedding, we can compute the similarity(S_{ij}) between node i and node j:

$$S_{ij} = \frac{\vec{h_i} \cdot \vec{h_j}}{\left|\vec{h_i}\right| \times \left|\vec{h_j}\right|}.$$

Using similarity between two nodes, we can compute the similarity between all nodes in a cluster and by using it, we can compute similarity of each community(S_c):

$$S_c = \frac{\sum_{i,j \in c, i \neq j} S_{ij}}{n(c) \times (n(c) - 1)}.$$

where $n(c)$ means the number of nodes in the community c. High S_c means that community C has similar nodes in there. Finally the fitness function is

$$S = \frac{\sum_{c \subset C_{tot}} S_c}{n(C_{tot})}.$$

We can also compute the dissimilarity(D_{ij}) of two nodes by using global structural vector(g_i), which are from node embedding:

$$D_{ij} = 1 - \frac{\vec{g_i} \cdot \vec{g_j}}{|\vec{g_i}| \times |\vec{g_j}|}.$$

Then, the fitness function is

$$D = \frac{\sum_{i,j \in A_n, i \neq j} D_{ij} - \sum_{c \subset C_{tot}} \sum_{i,j \in c, i \neq j} D_{ij}}{l}$$

where A_n is a set of all nodes in the graph and l is the number of edges which are not inside any clusters.

In particular, we use sampling to calculate the fitness function due to large computation cost.

What we can expect from this is to identify communities that are not only similar in local nature (research field or affiliation) but also diverse in combination of authors' roles. The resulting communities would have expressive power to distinguish true/false future co-authorship prediction.

2.2.3 Initial population / Operators

In this section, we introduce initial population, crossover operator and mutation operator briefly. These methods are proposed in [10].

Initially, a graph is constructed as same as stated in section 2.1. To construct each individual, first each node is assign to different communities, i.e., at first the number of communities is n =# of nodes. For each iteration, pick random node v and say its community v_c. Then, move its all neighbors into v_c. Repeat this iteration $0.4 \times n$ times.

For crossover operator, let's say p_1 and p_2 be two partitions to be crossovered. As similar to initial population scheme, we pick a random node u in p_1 and v in p_2, set community number of v and its neighbors into u's community number.

For mutation operator, it's just crossover of one partition, i.e., $\text{mutation}(p) = \text{crossover}(p, p)$.

2.3 Binary Classification with SVM

We will make our training data using 70% ground truth data(query public and answer public data)and our validation data using 30% of ground truth data.

Before predicting probability, we extract features for each co-authors:

- # of co-authors
- size of each community in all pareto fronts that contains a single author of given co-authors. Then compute its count and arithmetic, geometric, harmonic means.
- size of each community in all pareto fronts that contains a pair of authors of given co-authors. Then compute its count and arithmetic, geometric, harmonic means.

To leverage the above features, then use SVM with rbf kernel to classify each co-authors into true or false. We tested that just simple linear kernel shows lower performance.

3. EXPERIMENTS

3.1 Dataset Split

We construct the graph only with true dataset (paper_author and query_public_true). Since we have to use some part of dataset to train and validate SVM, we split the dataset as the following:

- Test set: query_private
- Dataset for SVM: query_public_false and the same number of random subset of query_public_true + paper_author. So, true and false data are 50%:50%.
- Train set for SVM: 70% of *data set for SVM*
- Valid set for SVM: 30% of *data set for SVM*
- Dataset for graph construction: the remaining part of true dataset after removing *dataset for SVM.*

3.2 Model Parameters

$p = 1, q = 0.5$ and $p = 1, q = 2$ are used to build two node embeddings. Population size during MOEA iteration is set to 50 and the model is iterated 500 times. For final prediction, C=1 and gamma=1 is used for SVM with rbf kernel.

3.3 Results

Our baseline model is the random guess. The expected binary accuracy measured by random guess should be 50% since our train and valid sets have exactly the same number of true and false sets.

Model Name	Train Acc.	Valid Acc.
Random Guess	50.0%	50.0%
NMOEA Iter 0	55.4%	54.8%
NMOEA Iter 500	65.8%	**61.8%**

Table 1: Train and validation accuracy for random guess and our model(NMOEA).

4. CONCLUSIONS

We proposed new method NMOEA for co-authorship prediction, which detects communities explicitly by combination of node embedding and MOEA, and then extracts simple features from them to classify each co-authors into true/false binary class. It outperforms our baseline random-guess model in tractable time, which shows it has expressive power to classify co-authors into binary class.

However, our proposed method has some limitations and have a room for improvement because we didn't have much time to execute. First, we ran our algorithm with only few iterations which is very small number in usual Multi-objective algorithms. Second, we didn't put much time for tuning our fitness functions. So, we might get better results if we tune these parameters.

5. REFERENCES

[1] Jaideep Srivastava Ankit Sharma and Abhishek Chandra. Predicting multi-actor collaborations using hypergraphs, 2014.

[2] MT Schaub A Jadbabaie AR Benson, R Abebe and J Kleinberg. Simplicial closure and higher-order link prediction. In *Proceedings of the National Academy of Sciences of the United States of America*, volume 48, 2018.

[3] C. M. Fonseca and P. J. Fleming. Multiobjective optimization and multiple constraint handling with evolutionary algorithms. i. a unified formulation. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 28(1):26–37, 1998.

[4] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks, 2016.

[5] J. Horn, N. Nafpliotis, and D. E. Goldberg. A niched pareto genetic algorithm for multiobjective optimization. In *Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence*, pages 82–87 vol.1, 1994.

[6] Z. Li, J. Liu, and K. Wu. A multiobjective evolutionary algorithm based on structural and attribute similarities for community detection in attributed networks. *IEEE Transactions on Cybernetics*, 48(7):1963–1976, 2018.

[7] C. Liu, J. Liu, and Z. Jiang. A multiobjective evolutionary algorithm based on similarity for community detection from signed social networks. *IEEE Transactions on Cybernetics*, 44(12):2274–2287, 2014.

[8] J. Liu, W. Zhong, H. A. Abbass, and D. G. Green. Separated and overlapping community detection in complex networks using multiobjective evolutionary algorithms. In *IEEE Congress on Evolutionary Computation*, pages 1–7, 2010.

[9] Kijung Shin Se-eun Yoon, Hyungseok Song and Yung Yi. Separatedhow much and when do we need higher-order informationin hypergraphs? a case study on hyperedge prediction. In *WWW*, 2020.

[10] Mursel Tasgin, Amac Herdagdelen, and Haluk Bingol. Community Detection in Complex Networks Using Genetic Algorithms. *arXiv e-prints*, page arXiv:0711.0491, November 2007.

[11] X. Wen, W. Chen, Y. Lin, T. Gu, H. Zhang, Y. Li, Y. Yin, and J. Zhang. A maximal clique based multiobjective evolutionary algorithm for overlapping community detection. *IEEE Transactions on Evolutionary Computation*, 21(3):363–377, 2017.

[12] L. Zhang, H. Pan, Y. Su, X. Zhang, and Y. Niu. A mixed representation-based multiobjective evolutionary algorithm for overlapping community detection. *IEEE Transactions on Cybernetics*, 47(9):2703–2716, 2017.

[13] Q. Zhang and H. Li. Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, 2007.

APPENDIX

A. EXPLANATORY DATA ANALYSIS

A.1 Low-order Network

We checked the distribution of the order(the number of coauthors) of a paper in the dataset which is shown in the following graph:

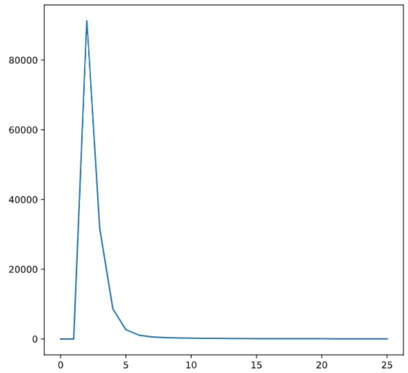

Figure 1: Here, the x-axis represents the number of coauthors and the y-axis represents the number of papers.

It shows that the given dataset is low-order network.

A.2 Higher-order interactions on Low-order Network

We first defined a hypergraph based on paper-author-set. Then, we computed geometric mean(GM), harmonic mean(HM) and arithmetic mean(AM) of 3-order expansions described on [9] for public-true-set and public-false-set. We observed that most of measurements are zero due to low-order. That means we cannot distinguish arbitrary low-order papers are true or false. The following shows the resulted statistics.

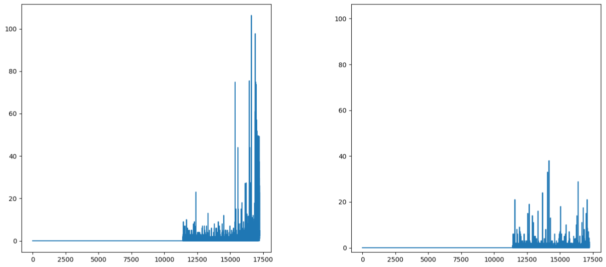

Figure 2: Geometric mean of 3-order expansions for true(left) and false(right) public-sets. More than 50% of papers have zero value. Other metrics(HM and AM) are omitted since they have identical distribution to GM.