Link prediction with structural information

Shitao Lu and Jing Yang

Abstract

In this report, we introduce our solutions to missing link prediction tasks on ogbl-ppa and ogbl-ddi. For the PPA dataset, we combine scores of local similarity measures and the label information of nodes to get the state-of-the-art performance. For the DDI dataset, we use multiple anchor sets selected from random sampling to encode distance information for edges on graph. In addition, we modify the aggregation stage of GraphSAGE[1] to incorporate edge information. Experimental results show that our methods outperform existing methods by a large extent.

1 Introduction

1.1 PPA

The ogbl-ppa dataset is an undirected, unweighted graph. Nodes represent proteins from 58 different species, and edges indicate biologically meaningful associations between proteins, e.g., physical interactions, co-expression, homology or genomic neighborhood[2]. The task is to predict new association edges with the given training edges[3]. Existing methods like Common Neighbors[4], Resource Allocation[5] and Adamic Adar[6] assign a score for each edge respecting how possible it may exist but they only take care of the local structure information neglecting the label information of nodes in PPA. We take advantage of the neglected labels together with the scores.

1.2 DDI

The ogbl-ddi dataset is a homogeneous, unweighted, undirected graph, representing the drug-drug interaction network[7]. From the leaderboard we know that distance enhanced methods like Distance Encoding (DE)[8] can help to achieve better performance. Motivated by this, we use multiple anchor sets to encode distance information. Furthermore, we modify the aggregation stage of GraphSAGE[1] with distance information of edges to get better embeddings of nodes. Our method achieves the best performance on DDI.

2 Methods

2.1 Problem Definition

Let $G=(V,E)$ be the given undirected graph, where V and E are the sets of nodes and edges respectively. Each edge in E can be represented as node pair (u,v), where $u,v\in V$. In the link prediction problem, E is split into E_{train}, E_{valid}, and E_{test}. It is allowed to train with E_{train} and to choose the best model with E_{valid}. The final performance is tested on E_{test}.

2.2 PPA

Previous local similarity measures like Common Neighbors, Resource Allocation and Adamic Adar are all based on the local structural information contained in the training set. These methods are fast, parameter-free and effective on PPA, but the label information of PPA have not been used in such methods. We combine these heuristic methods with MLP and take advantage of the node labels in PPA.

For a pair of nodes (u,v) in *V*, aforesaid heuristic methods assign a score to each edge. To be specific, for a node pair (u,v), let s_{cn}, s_{ra}, s_{aa} be the score assigned by Common Neighbors, Resource Allocation and Adamic Adar respectively:

$$s_{cn} = |\Gamma(u)\cap\Gamma(v)|$$

$$s_{ra} = \sum_{z\in\Gamma(u)\cap\Gamma(v)} \frac{1}{k(z)}$$

$$s_{aa} = \sum_{z\in\Gamma(u)\cap\Gamma(v)} \frac{1}{\log k(z)}$$

where $\Gamma(x)$ donates the neighbors set of x and $k(x)=|\Gamma(x)|$.

We concatenate these scores with one-hot vectors of u and v, so the representation we construct for node pair (u,v) is:

$$h_{u,v} = concat(one-hot_u, one-hot_v, s_{cn}, s_{ra}, s_{aa})^T$$

We treat existing edges in training set as positive training edges. For negative training edges, we sample the same amount by randomly sampling node pairs from *V*. To train with the feature, we employ a multi-layer perceptron with two hidden layers to classify edges as positive or negative.

2.3 DDI

DE: Distance Encoding[8] intends to use graph distance measures such as shortest path distance (SPD) between the node set whose representation to be learned and each node in the graph, as extra node features. For large scale graph, it is difficult to get all needed SPDs of node pairs. In our experiments, we randomly select K_A nodes from *V* to be anchor nodes and then calculate the shortest path starting from these anchor nodes to any other nodes. After that, the distance between *u* and *v* can be estimated by:

$$d_{u,v} = \frac{1}{K_A} \sum_{i=1}^{K_A} d_{u,a_i} + d_{v,a_i}$$

To reduce the randomness, we use k anchor sets to generate multiple distance features, where k is a hyper parameter to tune. Specifically, we firstly choose K ($K > K_A$) nodes at random to generate anchors set V_{anchor} and calculate the SPDs from these nodes to any others. Then, K_A nodes are randomly chosen from V_{anchor} for k times to generate k extra distance features. In this report, we set K and K_A to be 500 and 200 respectively.

GraphSAGE: GraphSAGE[1] is an inductive GNN model to generate node embeddings. Let h_v^l donate the hidden representation of node v at the l-th layer of a GNN model. The core aggregation stage of GraphSAGE can be defined as:

$$h_v^l = W_1^l h_v^{l-1} + W_2^l \text{mean}_{u \in \Gamma(v)} h_u^{l-1}$$

To take full advantage of the edge information, we modified the aggregation operator of GraphSAGE as follows:

$$h_v^l = W_1^l h_v^{l-1} + W_2^l \text{mean}_{u \in \Gamma(v)} \text{ReLU}(h_u^{l-1} + W h_{u,v})$$

which is similar as GINE convolution operator in [9]. During the training phase, we sample equal numbers of negative edges.

3 Experiments and Results

As required, we conduct our experiments for 10 times with random seeds from 0 to 9 and calculate the mean value with standard deviation.

3.1 PPA

We adopt a simple MLP with two hidden layers which have 256 and 512 dimensions respectively. During training, we set the parameters as listed in Table 1.

Table 1. Detailed settings of our experiments

Parameter	PPA	DDI
runs	10	10
epochs	20	400
learning rate	0.005	0.003
optimizer	SGD	Adam
dropout	0.5	0.3
batch size	4096	65536
parameters	163330	3761665
node embedding	-	512
hidden channels	256,512	512
num layers	2	2

The results on PPA are reported in Table 2. To better learn the effect of additional label information added to local structure methods, we conduct extra experiments with Common Neighbor, Resource Allocation and Adamic Adar respectively. It shows that simple local similarity measures can actually achieve higher performance with the help of label information.

Table 2. Results on PPA with our algorithm

Algorithm	Test Hits@100	Val Hits@100
CN	0.2765±0.0000	0.2832±0.0000
RA	0.4933±0.0000	0.4722±0.0000
AA	0.3245±0.0000	0.3268±0.0000
MLP+CN	0.3064±0.0116	0.3161±0.0070
MLP+RA	0.4896±0.0048	0.4794±0.0029
MLP+AA	0.3459±0.0033	0.3454±0.0029
MLP+CN&RA&AA	**0.5062±0.0035**	**0.4906±0.0029**

3.2 DDI

The detailed parameter settings for the DDI dataset are listed in Table 1. Results on DDI are showed in Table 3, which indicates that our methods can outperform existing methods shown in the leaderboard by a large extent.

Table 3. Results on DDI with our algorithm

Algorithm	Test Hits@20	Val Hits@20
GraphSAGE + Edge Attr (k =1)	0.8633±0.0313	0.7916±0.0324
GraphSAGE + Edge Attr (k =3)	**0.8781±0.0474**	**0.8044±0.0404**
GraphSAGE + Edge Attr (k =5)	0.8527±0.0247	0.7839±0.0278

4 References

[1] Hamilton W L, Ying R, Leskovec J. Inductive Representation Learning on Large Graphs[J]. 2017.

[2] Damian Szklarczyk, Annika L Gable, David Lyon, et al. STRING v11: protein–protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets. Nucleic Acids Research, 47(D1):D607–D613, 2019.

[3] Weihua Hu, Matthias Fey, et al. Open graph benchmark: Datasets for machine learning on graphs, 05 2020.

[4] Barabasi, A.-L. and Albert, R. Emergence of scaling in random networks. science, 286(5439):509–512, 1999.

[5] Tao Zhou & Linyuan Lü & Yi-Cheng Zhang, 2009. "Predicting missing links via local information," The European Physical Journal B: Condensed Matter and Complex Systems, Springer; EDP Sciences, vol. 71(4), pages 623-630, October.

[6] L. A. Adamic, E. Adar, *Social Networks* 25, 211 (2003).

[7] David S Wishart, Yannick D Feunang, et al. DrugBank 5.0: a major update to theDrugBank database for 2018. Nucleic Acids Research, 46(D1): D1074–D1082, 2018.

[8] Pan Li, Yanbang Wang, Hongwei Wang, and Jure Leskovec. Distance encoding: Design provably more powerful neural networks for graph representation learning, 2020.

[9] W. Hu*, B. Liu*, J. Gomes, M. Zitnik., P. Liang, V. Pande, J. Leskovec International Conference on Learning Representations (*ICLR*), 2020.