Supplementary Document for 'Deep Learning for Link Prediction in Realistic Biomedical Graphs: A Multidimensional Evaluation of Graph Embedding-based Approaches'

Gamal Crichton, Yufan Guo, Sampo Pyysalo and Anna Korhonen

1 Introduction

This document is supplementary to the paper: *Deep Learning for Link Prediction in Realistic Biomedical Graphs: A Multidimensional Evaluation of Graph Embedding-based Approaches*. It contains additional results and analysis which were left out of the main paper due to space constraints.

For SDNE, two implementations were tried: the one created by the authors (Wang et al., 2016) and one created by (Goyal and Ferrara, 2017). We used the parameters from (Goyal and Ferrara, 2017) because our attempted hyper-parameters did not give good results and, though we contacted both sets of authors, only they responded to our request for the hyper-parameters used in their experiments.

2 Results and Discussion

In the result tables, the number in **bold** represent the best score for a particular metric. The difference between the best and scores with an asterisk (*) are not statistically significant.

2.1 MATADOR

These results are in Table 1. The additional result is that SDNE is much worse than the other approaches for this dataset. This may be due to the fact that it is the deepest of all the deep learning approaches and so required more data to train properly. In the main paper, we already attribute the relatively poor performance of the deep learning models compared to the baselines to the small size of this dataset - that argument would hold even more so for SDNE.

Note also that LINE embeddings combined with Hadamard were on par with the best performer for precision at k.

2.2 BioGRID

The randomly sliced experiments on this dataset are in Table 2 and the time-sliced experiments are in Table 3.

2.2.1 Random-Slice

Node2vec embeddings combined with Hadamard were on par with the best performer for precision at k.

2.2.2 Time Slice

Section 3.1 of the paper explains why it is more difficult to perform link prediction in the time-slice setting. To recap: first, new nodes can be introduced to the graph at later time periods which will present little or no information to the link predictor to use as they will have no links to other nodes in the time period which the predictor uses to make predictions. Second, in evolving graphs, the easier links tend to form first and more difficult ones later, so the edges to be predicted in later time periods tend to be more difficult.

As expected, the majority of the approaches performed worse in all metrics than the randomly sliced experiments with this dataset. However there were some exceptions. DeepWalk embeddings combined by Weighted-L1 and L2, node2vec embeddings combined with Weighted-L1 and all baselines recorded

Method	Node Combination	AUC (ROC)	AUC (PR)	MAP	Avg. R-prec	Prec @ k
Deep-Walk	Average	95.93	95.82	89.81	86.86	98.77*
	Concat	94.97	94.83	88.30	84.63	98.34*
	Hadamard	90.21	91.55	86.65	82.59	97.56
	W-L1	80.45	82.74	69.27	62.56	93.74
	W-L2	85.67	88.12	77.31	71.57	97.44
LINE	Average	80.63	81.30	67.74	61.04	91.65
	Concat	81.16	81.82	68.53	61.42	92.00
	Hadamard	89.11	90.37	83.45	77.47	98.00
	W-L1	70.76	79.32	73.86	66.15	98.02*
	W-L2	69.52	76.37	70.94	63.33	92.38
node-2vec	Average	78.38	78.75	66.42	59.32	88.67
	Concat	77.62	77.54	65.44	58.40	87.25
	Hadamard	84.74	85.12	82.34	76.88	93.71
	W-L1	75.38	74.98	69.32	62.08	83.94
	W-L2	74.31	74.57	69.56	62.48	84.62
SDNE	Average	55.77	55.22	54.81	47.21	57.56
	Concat	54.88	54.17	53.37	46.14	56.41
	Hadamard	53.12	52.20	51.81	47.85	52.84
	W-L1	54.35	53.44	50.06	45.56	54.93
	W-L2	52.60	51.34	50.67	43.41	50.44
AA	N/A	91.97	88.40	87.16	85.06	86.87
CN	N/A	**97.27**	97.04*	**95.47**	**94.64**	98.74*
JC	N/A	97.23*	**97.10**	94.72	92.29	**98.96**

Table 1: MATADOR random-slice results

Method	Node Combination	AUC (ROC)	AUC (PR)	MAP	Avg. R-prec	Prec @ k
DeepWalk	Average	97.69	97.62	79.24	73.86	99.30
	Concat	97.74	97.65	82.48	77.70	99.18
	Hadamard	95.76	96.54	79.63	74.87	99.25
	W-L1	79.17	80.57	51.96	46.50	91.71
	W-L2	79.73	81.08	52.81	47.39	92.12
LINE	Average	98.10*	97.80*	83.13*	78.22*	99.54*
	Concat	98.08	97.76	82.94	78.04	99.29
	Hadamard	94.45	95.35	80.17	75.17	99.30
	W-L1	92.41	92.06	70.88	65.21	97.07
	W-L2	91.80	91.55	71.80	66.39	96.56
node2vec	Average	98.32*	97.97*	85.70*	81.17*	99.38*
	Concat	**98.51**	**98.26**	**86.49**	**81.84**	99.49*
	Hadamard	97.19	97.17	81.53	76.54	99.33*
	W-L1	92.02	92.30	64.24	59.45	97.45
	W-L2	93.07	93.01	67.11	61.94	97.47
AA	N/A	86.10	90.75	70.97	57.65	96.13
CN	N/A	91.20	94.96	75.72	69.81	**99.64**
JI	N/A	90.80	93.95	73.93	68.79	98.59

Table 2: BioGRID random-slice results

better performance for MAP. DeepWalk embeddings combined by Weighted-L1 and L2, node2vec embeddings combined with Weighted-L1 and Adamic-Adar recorded better performance for averaged R-precision. Adamic-Adar also recorded increased performance for precision at k. There are several possible contributing factors here.

For MAP and averaged R-precision, if a particular node has no positives it is removed from the calculations as these metrics are only concerned with predicted true positives. In the time-sliced data, there are a much higher percentage of nodes which have no true positives in the test slice than is the case with randomly-sliced data. These nodes are also likely to have a small amount of links and are thus difficult nodes to perform well on, so it is not surprising that the approaches which performed poorest on the randomly-sliced version of this dataset benefited from having less and easier nodes in the evaluation. The poor embeddings created for this setting as explained above would contribute to decreased performance for the other methods but as all combination methods use the same embeddings, there is something about the DeepWalk embeddings combined with Weighted L1 and L2 which help in this setting.

Node2vec embeddings combined with Hadamard had performance that was not significantly worse than the best for AUPRC and precision at k.

2.3 PubTator

The randomly sliced experiments on this dataset can be seen in Table 4 and the time-sliced experiments can be seen in Table 5.

2.3.1 Random-Slice

Nothing much to add here except to note that Common Neighbours outperformed the lower neural network performers (Hadamard, Weighted-L1 and Weighted-L2) for most metrics.

2.3.2 Time Slice

As with the BioGRID data, the majority of the approaches performed worse in this setting than the random-sliced one, and there were again some exceptions. DeepWalk embeddings combined by Weighted-L1 and L2 had better performance in all metrics and Adamic-Adar again recorded increased

Method	Node Combination	AUC (ROC)	AUC (PR)	MAP	Avg. R-prec	Prec @ k
Deep-	Average	89.40	90.10	68.94	63.30	97.25*
Walk	Concat	92.12	92.78	71.61	65.96	98.04
	Hadamard	89.03	91.39	66.28	60.34	98.31
	W-L1	69.75	67.43	59.74	54.61	73.26
	W-L2	72.11	69.33	59.84	54.51	75.02
LINE	Average	91.86	92.31	72.85	67.76	97.40
	Concat	93.55	93.74	73.60	68.57	97.90
	Hadamard	77.70	82.51	67.78	61.33	96.05
	W-L1	82.36	81.32	66.66	60.93	88.54
	W-L2	79.79	78.82	66.53	60.75	86.76
node-	Average	**95.25**	**95.43**	74.91	**70.39**	98.26
2vec	Concat	93.66	94.66*	73.48	68.77	98.40*
	Hadamard	93.94	94.02*	71.81	66.57	97.59*
	W-L1	89.06	88.70	66.17	61.20	93.86
	W-L2	88.81	88.43	66.09	61.02	93.54
AA	N/A	77.46	87.69	74.84	61.39	98.10
CN	N/A	85.07	91.81	**76.20**	67.73	**99.38**
JC	N/A	84.74	90.20	75.60	67.49	97.45

Table 3: BioGRID time-slice results

performance for precision at k. Similar explanations hold for this situation as well. In this case only the DeepWalk vectors were better and they were better in all metrics and the previous explanations pertained only to the node-level metrics. These results provide strong indication that DeepWalk embeddings combined with Weighted-L1 and Weighted-L2 perform better in the time sliced setting than the random slice one, but their performances are still significantly worse than the best performers in these settings.

References

Palash Goyal and Emilio Ferrara. 2017. Graph embedding techniques, applications, and performance: A survey. *arXiv preprint arXiv:1705.02801*.

Daixin Wang, Peng Cui, and Wenwu Zhu. 2016. Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 1225–1234, New York, NY, USA. ACM.

Method	Node Combination	AUC (ROC)	AUC (PR)	MAP	Avg. R-prec	Prec @ k
Deep-Walk	Average	98.85	99.01	83.67	75.97	99.93*
	Concat	**99.20**	**99.30**	**91.01**	85.46	99.94*
	Hadamard	98.44	98.68	84.67	77.84	99.88
	W-L1	88.96	89.63	60.76	51.21	97.64
	W-L2	89.25	89.90	62.10	52.57	97.67
LINE	Average	99.10*	99.23*	90.36*	84.56	**99.97**
	Concat	99.13	99.24	90.07	84.03	99.95*
	Hadamard	98.30	98.49	86.40	79.28	99.90
	W-L1	93.93	94.16	78.25	69.48	98.97
	W-L2	94.23	94.51	77.97	69.00	99.13
node-2vec	Average	98.71	98.90	82.98	75.29	99.94*
	Concat	99.16*	99.21	88.94	82.14	99.92*
	Hadamard	98.81	98.91	86.40	79.07	99.87
	W-L1	88.07	87.28	87.28	48.95	94.08
	W-L2	88.85	88.26	88.26	50.72	94.90
AA	N/A	92.92	84.56	56.48	66.38	83.33
CN	N/A	98.40	98.28	79.84	**87.10**	99.94*
JI	N/A	92.36	87.59	65.44	59.74	91.21

Table 4: PubTator random-slice results

Method	Node Combination	AUC (ROC)	AUC (PR)	MAP	Avg. R-prec	Prec @ k
Deep-Walk	Average	93.86*	95.51*	70.78*	62.16*	**99.89**
	Concat	**93.99**	**95.70**	**71.11**	**62.65**	**99.89**
	Hadamard	87.23	91.33	54.72	46.22	99.70
	W-L1	92.06	93.23	66.47	57.29	98.77
	W-L2	91.81	93.06	65.89	56.66	98.76
LINE	Average	88.68*	92.27*	55.61*	46.41*	**99.89**
	Concat	90.32	93.01	62.51	53.21	**99.89**
	Hadamard	87.09	89.98	51.97	42.43	99.10
	W-L1	83.58	86.55	47.71	38.11	97.26
	W-L2	82.81	85.79	47.07	37.49	96.78
node-2vec	Average	88.40	92.07	55.72	46.48	99.87
	Concat	88.13	91.83	53.24	43.69	99.84
	Hadamard	85.24	90.63	47.76	38.84	99.81*
	W-L1	84.68	89.08	44.69	35.34	98.57
	W-L2	84.48	89.12	44.68	35.49	98.67
AA	N/A	85.10	80.24	35.49	40.13	90.56
CN	N/A	88.37	88.83	43.67	46.59	99.84
JI	N/A	86.08	83.52	38.66	38.75	94.27

Table 5: PubTator time-slice results