ICANN19

28th International Conference on Artificial Neural Networks

Multitask Learning on Graph Neural Networks: Learning Multiple Centrality Measures with a Unified Network

Pedro H.C. Avelar, Henrique Lemos, Marcelo O.R. Prates, Luis C. Lamb

Contact Author: phcavelar@inf.ufrgs.br

INSTITUTO DE INFORMÁTICA UFRGS

CAPES CNPq Conselho Nacional de Desenvolvimento Científico e Tecnológico

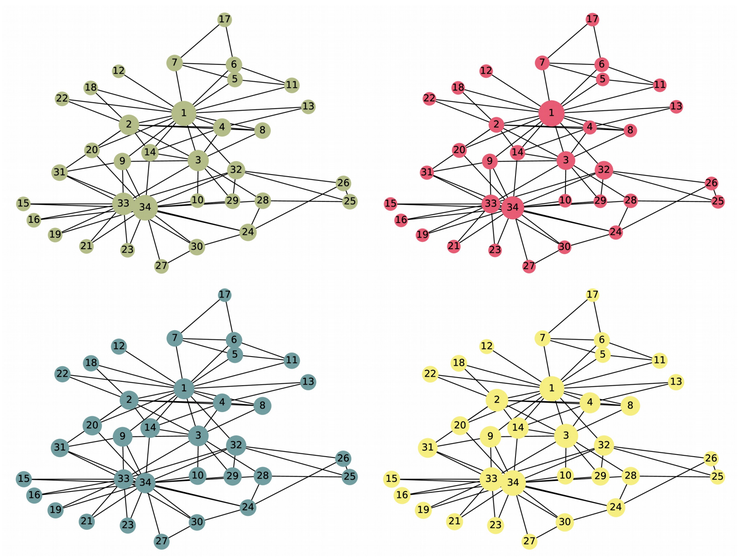
Research Questions

- Can GNNs benefit from Multitask Learning?
- Can GNNs approximate centrality measures solely from the network structure?

Network Centrality Measures

In general, node-level centralities attempt to summarise a node's contribution to the network cohesion. Several centralities have been proposed and many models and interpretations have been suggested, namely: autonomy, control, risk, exposure, influence, etc. Despite their myriad of applications and interpretations, in order to calculate some of these centralities one may face both high time and space complexity, thus making it costly to compute them on large networks. Although some studies pointed out a high degree of correlation between some of the most common centralities [6], it is also stated that these correlations are attached to the underlying network structure and thus may vary across different network distributions [8]. Therefore, techniques to allow faster centrality computation are topics of active research [1].

Here, however, we are not concerned as much with the time complexity of computing the centrality measure per se, but with whether a neural network can infer a node's centrality solely from the network structure – that is, without any numeric information about the node, its neighbourhood, or the network itself – even if the complexity of the methods presented here is similar to that of matrix operations, as the underlying procedures are based on these, and is polynomial with the size of the input. With this in mind, we selected four well-known node centralities to investigate in our study: **degree –** which simply calculates to how many neighbours a node is connected; **betweenness** – which calculates the number of shortest paths which cross by the given node, indicating that a node is more important to the graph's cohesion; **closeness** – which is a distance-based centrality measuring the average geodesic distance between a given node and all other reachable nodes; **eigenvector** – which uses the largest eigenvalue of the adjacency matrix to compute its eigenvector and assigns to each node a score based on the scores of its neighbours, being usually computed via a power iteration method without convergence guarantees.



Approximating Centralities with ANNs

[1,2,3,4,5] all use neural networks to estimate centrality measures. However, in [1,2,3] they use a priori knowledge of other centralities to approximate a different one, on [4] they also produce a ranking of the centrality measures, but do so using the degree and eigenvector centralities as input, and in [5] local features such as number of vertices in a network, number of edges in a network, degree and the sum of the degrees of all of the vertex's neighbours are used. These contributions differ from ours in that we feed our neural network solely with the network structure – that is, they use a simple MLP which receives numeric information about a specific node and outputs an approximation of one desired centrality measure, while our builds a message-passing procedure with only the network structure and no numeric information whatsoever. [7] also uses GNNs to compute rankings for the PageRank centrality measure, but does for a single graph, and does not focus on other centralities nor analyses the transfer between centralities.

Approximating vs Ranking

Most of the cited previous work focus on the rankings produced by their estimators, instead of the approximation of the centrality values per se. In this work we build models that approximate the normalised centralities (AN), those that approximate the normalised centralities but have the normalisation done outside the ANN (AM), and those that produce rankings natively by comparing the embeddings of two different nodes with a learned comparison function (RN), effectively producing a fuzzy comparison matrix as exemplified below.

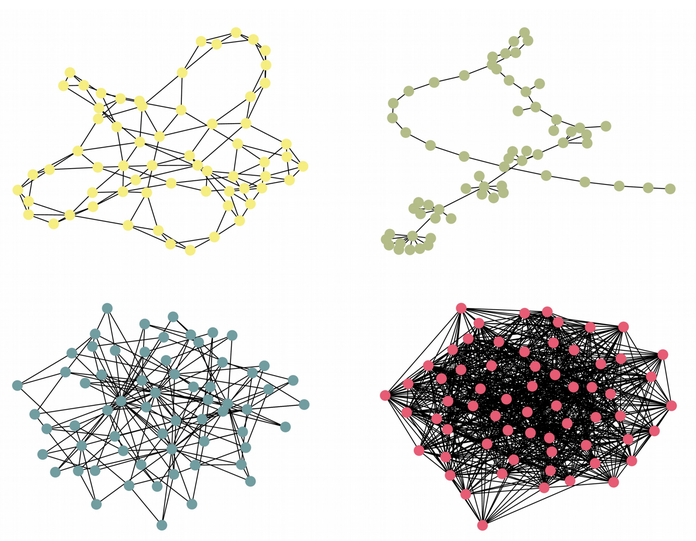
$$\begin{pmatrix} P(v_1 >_c v_1) & P(v_2 >_c v_1) & P(v_3 >_c v_1) \\ P(v_1 >_c v_2) & P(v_2 >_c v_2) & P(v_3 >_c v_2) \\ P(v_1 >_c v_3) & P(v_2 >_c v_3) & P(v_3 >_c v_3) \end{pmatrix} \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

Training and Testing

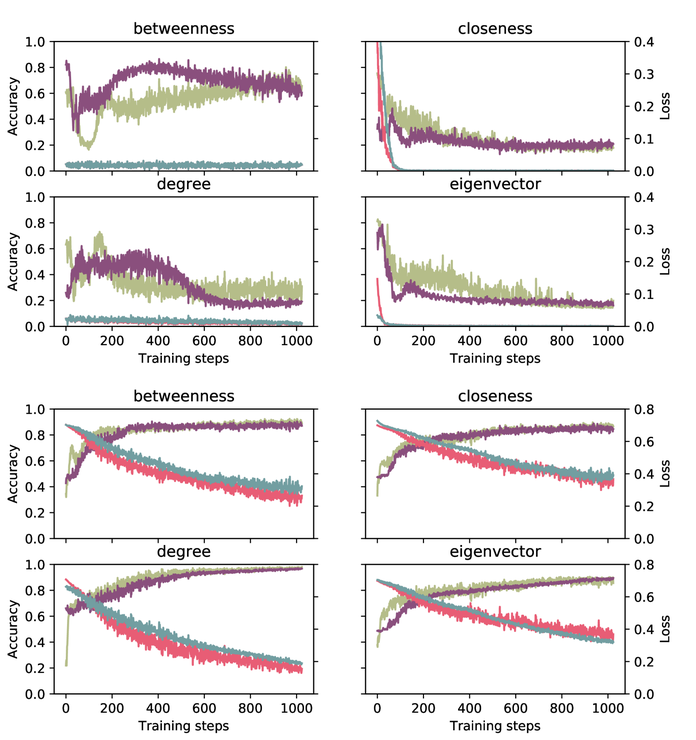
Instead of optimising our target function in the same network we will test it, as some previous work do, we try make our network learn to model centrality measures on arbitrary distributions by making use of the random network models available in the literature for training/testing and using real world networks exclusively for testing.

We train our GNN on a synthetic dataset composed of graphs sampled from the Watts-Strogatz small-world, a power-law tree, the Holme-Kim and the Erdós-Renyi models, with examples shown of these distributions shown below. We test our models on newly generated graphs from these same distributions, with the same amount of nodes as in the raining dataset as well as with more nodes than during training.

To evaluate how our model extends its predictions to different graphs distributions we test in on both synthetic networks from other distributions (such as shell graphs, and graphs sampled from the Barabási-Albert model) as well as real world networks as varied as power-grid, economic, biological, collaboration and social networks.



Experimental Results

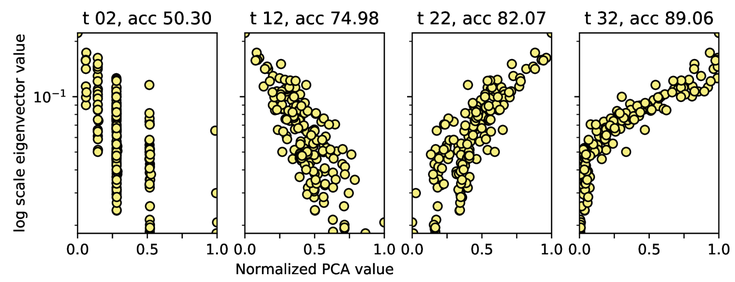


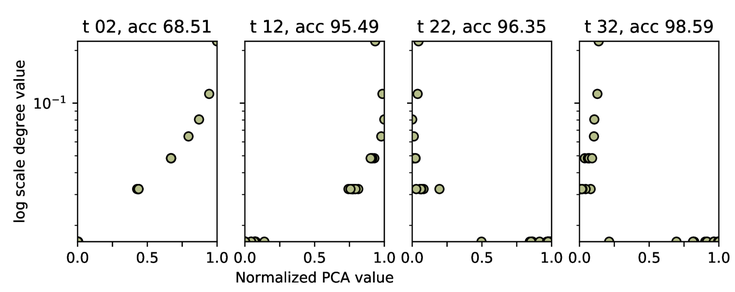
"Test" Centrality	AN P	AN R	AN TN	AN Acc	AM P	AM R	AM TN	AM Acc	RN P	RN R	RN TN	RN Acc
Without multitasking												
Betweenness	82.4	**88.6**	84.6	86.0	39.9	45.6	43.5	44.1	**90.3**	88.5	**91.0**	**89.8**
Closeness	83.1	**85.4**	84.3	84.8	83.1	**85.4**	84.4	84.9	**88.4**	84.5	**89.8**	**87.3**
Degree	75.3	**98.4**	81.6	87.5	63.3	85.8	70.4	75.6	**99.3**	94.9	**99.4**	**97.6**
Eigenvector	87.0	87.0	87.6	87.3	86.5	86.5	87.0	86.8	86.2	**90.2**	82.3	86.3
Average	82.0	**89.8**	84.5	86.4	68.2	75.8	71.3	72.8	**91.0**	89.5	90.6	**90.3**
With multitasking												
Betweenness	73.6	79.6	76.1	77.3	41.9	46.7	46.3	46.2	87.2	87.2	88.9	87.9
Closeness	71.4	73.0	73.6	73.3	80.2	82.5	81.7	82.1	86.9	82.0	88.7	85.5
Degree	73.7	96.0	80.4	85.9	72.4	94.3	79.3	84.7	98.3	92.4	99.0	96.4
Eigenvector	83.5	83.5	84.2	83.9	85.5	85.5	86.2	85.9	**89.8**	88.3	**90.4**	**89.4**
Average	75.5	83.0	78.6	80.1	70.0	77.3	73.4	74.7	90.5	87.5	**91.8**	89.8

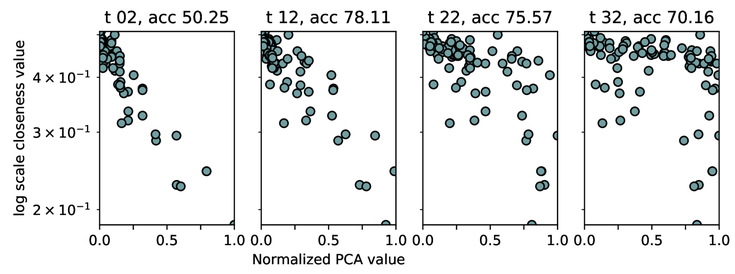
"Different" Centrality	AN P	AN R	AN TN	AN Acc	AM P	AM R	AM TN	AM Acc	RN P	RN R	RN TN	RN Acc
Without multitasking												
Betweenness	80.2	**80.3**	80.8	**80.5**	43.1	43.2	44.8	44.0	**81.2**	77.5	**81.8**	79.7
Closeness	80.8	82.4	81.6	82.0	80.8	82.4	81.6	82.0	81.7	75.3	**84.2**	79.9
Degree	82.7	94.6	85.2	89.0	75.3	86.8	78.3	81.7	86.4	72.5	89.0	82.1
Eigenvector	70.4	70.4	71.2	70.8	69.9	69.9	70.7	70.3	**84.9**	**87.9**	**83.8**	**85.8**
Average	78.5	**81.9**	79.7	80.6	67.3	70.6	68.9	69.5	**83.6**	78.3	**84.7**	**81.9**
With multitasking												
Betweenness	73.4	73.6	74.3	73.9	46.6	46.8	48.2	47.5	77.9	77.0	78.5	77.8
Closeness	81.3	82.9	82.2	82.5	**81.9**	**83.5**	82.7	**83.1**	79.6	77.5	81.4	79.5
Degree	85.0	**97.0**	87.4	**91.3**	84.1	96.1	86.5	90.3	**87.4**	74.9	**91.0**	84.0
Eigenvector	67.5	67.5	68.4	68.0	70.4	70.4	71.1	70.7	79.6	80.5	79.9	80.2
Average	76.8	80.3	78.1	78.9	70.8	74.2	72.1	72.9	81.1	77.5	82.7	80.4

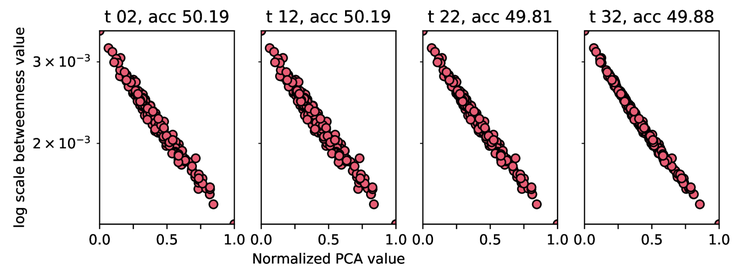
Interpretability

Inspired by [9] we applied PCA on the embeddings at the last layer of the GNN block to interpret our results, leading to mixed results. Sometimes the model would behave in an interpretable fashion, with its PCA showing a high visual correlation with the predicted centralities an sometimes it showed erratic behaviour. But in the cases where the PCA behaved well the model seemed to be storing information related to the log of the predicted centralities inside its embeddings.









Discussion

This work presents, to the best of our knowledge, the first application of Graph Neural Networks to multiple centrality measures and the proposal of a comparison framework for processing node embeddings. We yield an effective model and provide ways to have such a model work with various centralities at once, in a more memory-efficient way than having a different model for every centrality – with minimal loss in its performance.

Although our model is presented here with four sample centralities, some of which are easier to compute precisely rather than approximating with our method, we believe that the joint prediction of multiple centralities is still useful, since the time complexity of our model, which is dependant mostly on matrix multiplications, still remains polynomial even if one increases the amount of centralities being predicted.

References

[1] Grando, F., Granville, L.Z., Lamb, L.C.: Machine learning in network centrality measures: Tutorial and outlook. ACM Comput. Surv. 51(5), 102:1–102:32 (2019). DOI 10.1145/3237192

[2] Grando, F., Lamb, L.C.: Estimating complex networks centrality via neural networks and machine learning. In: IJCNN, pp. 1–8. IEEE (2015). DOI 10.1109/IJCNN.2015.7280334

[3] Grando, F., Lamb, L.C.: On approximating networks centrality measures via neural learning algorithms. In: IJCNN, pp. 551–557. IEEE (2016). DOI 10.1109/IJCNN. 2016.7727248

[4] Grando, F., Lamb, L.C.: Computing vertex centrality measures in massive real networks with a neural learning model. In: IJCNN, pp. 1–8. IEEE (2018). DOI 10.1109/IJCNN.2018.8489690

[5] Kumar, A., Mehrotra, K.G., Mohan, C.K.: Neural networks for fast estimation of social network centrality measures. In: FANCCO, pp. 175–184. Springer (2015). DOI 10.1007/978-3-319-27212-2_14

[6] Lee, C.Y.: Correlations among centrality measures in complex networks. arXiv preprint physics/0605220 (2006)

[7] Scarselli, F., Yong, S.L., Gori, M., Hagenbuchner, M., Tsoi, A.C., Maggini, M.: Graph neural networks for ranking web pages. In: Web Intelligence, pp. 666–672. IEEE Computer Society (2005). DOI 10.1109/WI.2005.67

[8] Schoch, D., Valente, T.W., Brandes, U.: Correlations among centrality indices and a class of uniquely ranked graphs. Social Networks 50, 46–54 (2017). DOI 10. 1016/j.socnet.2017.03.010

[9] Selsam, D., Lamm, M., Bünz, B., Liang, P., de Moura, L., Dill, D.L.: Learning a SAT solver from single-bit supervision. arXiv preprint abs/1802.03685 (2018)

Acknowledgements

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001, and by the Brazilian Research Council CNPq. We gratefully acknowledge the support of NVIDIA® Corporation with the donation of the Quadro® P6000 GPU used in this research.

Other contact Information:

Whatsapp: +5534992079305

Github: github.com/phcavelar