GNN 1: BEGINNING THE JOURNEY

Introduction

Graph neural network models, abbreviated GNN, have gained popularity as a powerful method for the analysis of network data structures, of which the convolutional network model, abbreviated GCN, resembles a simple approach to understand the philosophy and salient features of GNN models. This small study shall mark the commencement of future studies on various aspects of GNN models.

Theoretical considerations

The basic components of GCN model can be understood from numerous publications and online sources, of which we shall iterate the central points here. Let G be a network with the adjacency matrix A, which is a complete description of the connections between nodes. The GCN model states that each layer of the model has a function of the form

$$H_k = \varphi\left(\widetilde{D}^{-\frac{1}{2}}\tilde{A}\widetilde{D}^{-\frac{1}{2}}H_{k-1}W_{k-1}\right).$$

Here, H_{k-1} is the matrix of input features of the kth layer, W_k is a matrix of weights of the kth layer, and H_k is the matrix of output features of the kth layer, which would then become the input features of the $(k+1)$th layer. The operation $\tilde{A} = A + I$ imposes self-loops in the network, which indicates that for each node, the features of the node itself and of its neighbors would be extracted for analysis. The matrix $\widetilde{D}$ is a diagonal matrix which encodes the degrees for all nodes with respect to $\tilde{A}$. Since $\widetilde{D}^{-\frac{1}{2}}\tilde{A}\widetilde{D}^{-\frac{1}{2}}$ is a function of the network only, each layer of the GCN model can be viewed as a linear function of the input features H_k, followed by the nonlinear map φ, where the non-linear map is usually the one that makes the model more expressive.

Simulation protocol

To assess the versatility of the GCN model, we consider a simple hypothetical social network, where individuals are divided into two communities, of which individuals within the same community are more likely to interact with each other than with individuals from the other community. A natural speculation is that individuals within the same community are likely to share similar characteristics, so that we can use these characteristics to infer which community an individual is in. The network structure can be sampled from the stochastic block model, which treats interaction between each pair of individuals as an independent Bernoulli trial. The current simulation assumes that the number of individuals $n = 120$, of which individuals 1 to 60 comes from community 1, and individuals 61 to 120 comes from community 2. The probability of each interaction within a community is 0.8 and the probability of each interaction between two communities is 0.1. Since interactions are reciprocal, the network would be undirected.

Surmise also that each individual is associated with three covariates x_1, x_2 and x_3, which tends to be more similar within the community than between two communities. The simulation scheme for these covariates is described in Table 1:

Table 1: Distributions from which covariates are sampled

	$\boldsymbol{x_1}$	$\boldsymbol{x_2}$	$\boldsymbol{x_3}$
Community 1	$N(20, 4^2)$	$N(40, 4^2)$	$N(22, 3^2)$
Community 2	$N(30, 4^2)$	$N(30, 4^2)$	$N(25, 2^2)$

The tentative GCN to experiment with would be comprised of three layers, where the dimensions of the linear maps as well as the form of the nonlinear map φ are described in Table 2.

Table 2: Architecture of the GCN model

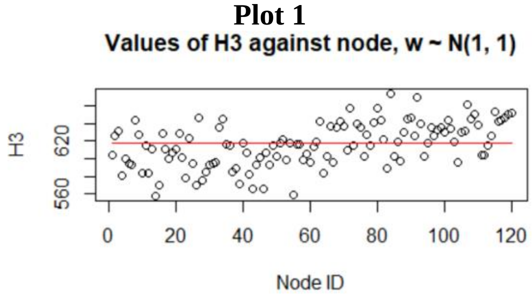
	Dimension of linear map	Nonlinear map
Layer 1	$R^{120\times 3} \mapsto R^{120\times 3}$	$ReLU$
Layer 2	$R^{120\times 3} \mapsto R^{120\times 2}$	$ReLU$
Layer 3	$R^{120\times 2} \mapsto R^{120\times 1}$	$ReLU$

Therefore, the GCN model aims to reduce the dimension of the input from 3 covariates to 1 feature only, and all layers shall use $ReLU$ as the nonlinear map.

Results and discussion

To commence, the weight matrices of all layers are initialized with independent normal distributions of $\mu = 1$ and $\sigma = 1$, to reduce the chance that too many entries in the output become zero after application of $ReLU$. In the sample produced, the GCN model calculates the output feature H_3 as shown in Plot 1.

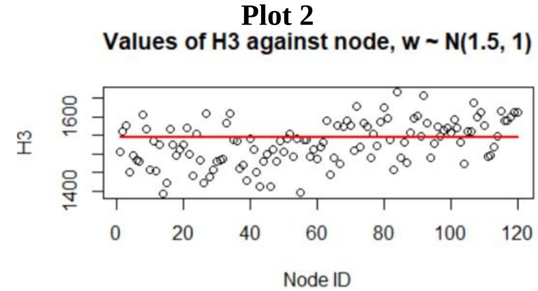
Plot 1



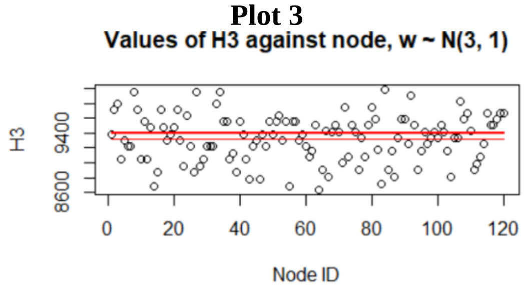
Obviously, the output feature H_3 centers around two different means for nodes 1 to 60 in community 1, and nodes 61 to 120 in community 2. To classify the individuals, a simple search over possible values of H_3 as threshold shows that a cutoff at 618 offers a maximum accuracy of 75.8%. This demonstrates that even without training, the aforementioned GCN model could reduce dimensionality of the covariates and create features useful for classification.

Nonetheless, attempts to use different initialization schemes for weights yield some interesting observations. For the same network, when $\mu = 1.5$ and $\mu = 3$ are chosen to initialize the weights, utility of H_3 is reduced, as shown in Plot 2 and Plot 3.

Plot 2



Plot 3



As μ increases, the distinction of the cutoff between community 1 and community 2 becomes less clear, and with $\mu = 3$, the cutoff is entirely absent. The increase in μ chosen to initialize the weights appears to have a negative impact on the utility of the output feature. This result is not entirely unexpected, if we consider the mathematical form of the GCN model. Since all entries in $\tilde{D}$ and $\tilde{A}$ must be nonnegative, and since $ReLU$ necessarily returns nonnegative values as well, from layer 2, all sources of negative values from the linear map must come from the weights themselves. When $\mu \to \infty$, $P(w < 0) = P\left(z < -\frac{\mu}{\sigma}\right) \to 0$, we have $P(x < 0) \to 0$ and $P(ReLU(x) \neq x) \to 0$, such that $ReLU$ behaves almost the same as an identity function with a loss of its nonlinear characteristics. This loss of nonlinear characteristics could explain why the output feature becomes less useful as μ increases. Nonetheless, this would unlikely be a problem in subsequent studies when the models are trained.

Conclusion

This small study has constructed a simple GCN model from basic mathematical principles, and demonstrated the potential for GCN to achieve dimensional reduction and offer useful features. This is only the start of the journey, and subsequent studies would be more inclined to use established statistical packages instead of manual creation of GCN.

References

[1] Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C. and Sun, M., 2020. Graph neural networks: A review of methods and applications. AI Open, 1, pp.57-81.

[2] Understanding Graph Convolutional Networks for Node Classification. [online] Available at: <https://towardsdatascience.com/understanding-graph-convolutional-networks-for-node-classification-a2bfdb7aba7b> [Accessed 22 May 2021].

[3] How to do Deep Learning on Graphs with Graph Convolutional Networks. [online] Available at: <https://towardsdatascience.com/how-to-do-deep-learning-on-graphs-with-graph-convolutional-networks-62acf5b143d0> [Accessed 22 May 2021].

[4] How to do Deep Learning on Graphs with Graph Convolutional Networks. [online] Available at: <https://towardsdatascience.com/how-to-do-deep-learning-on-graphs-with-graph-convolutional-networks-7d2250723780> [Accessed 22 May 2021].