Why Do Attributes Propagate in Graph Convolutional Neural Networks?

Liang Yang[1,2,3]**, Chuan Wang**[2*]**, Junhua Gu**[1,3]**, Xiaochun Cao**[2]**, Bingxin Niu**[1,3]

[1]School of Artificial Intelligence, Hebei University of Technology, Tianjin, China
[2]State Key Laboratory of Information Security, Institute of Information Engineering, CAS, Beijing, China
[3]Hebei Province Key Laboratory of Big Data Calculation, Hebei University of Technology, Tianjin, China
yangliang@vip.qq.com, {wangchuan, caoxiaochun}@iie.ac.cn, {jhgu_hebut, niubingxin666}@163.com

Abstract

Many efforts have been paid to enhance Graph Convolutional Network from the perspective of propagation under the philosophy that "Propagation is the essence of the GCNNs". Unfortunately, its adverse effect is over-smoothing, which makes the performance dramatically drop. To prevent the over-smoothing, many variants are presented. However, the perspective of propagation can't provide an intuitive and unified interpretation to their effect on prevent over-smoothing. In this paper, we aim at providing a novel explanation to the question of *"Why do attributes propagate in GCNNs?"*. which not only gives the essence of the oversmoothing, but also illustrates why the GCN extensions, including multi-scale GCN and GCN with initial residual, can improve the performance. To this end, an intuitive Graph Representation Learning (GRL) framework is presented. GRL simply constrains the node representation similar with the original attribute, and encourages the connected nodes possess similar representations (pairwise constraint). Based on the proposed GRL, exiting GCN and its extensions can be proved as different numerical optimization algorithms, such as gradient descent, of our proposed GRL framework. Inspired by the superiority of conjugate gradient descent compared to common gradient descent, a novel Graph Conjugate Convolutional (GCC) network is presented to approximate the solution to GRL with fast convergence. Specifically, GCC adopts the *obtained* information of the last layer, which can be represented as the difference between the input and output of the last layer, as the input to the next layer. Extensive experiments demonstrate the superior performance of GCC.

Introduction

Graph Neural Networks (GNNs) (Wu et al. 2020; Xu et al. 2019) have become a hot topic in deep learning for their potentials in modeling irregular data. GNNs have been widely used and achieved state-of-the-art performance in many fields, such as computer vision, natural language processing (Yang et al. 2020), traffic forecasting, chemistry and medical analysis, etc. Existing GNNs fall into two categories, spectral methods (Defferrard, Bresson, and Vandergheynst 2016) and spatial ones (Hamilton, Ying, and Leskovec 2017; Gilmer et al. 2017; Yang et al. 2019b,a; Jin et al. 2019, 2020, 2021).

Graph Convolutional Network (GCN) (Kipf and Welling 2017), which is a simple, well-behaved and insightful GNN, bridges above two perspectives by proving that the propagation can be motivated from a first-order approximation of spectral graph convolutions. Recently progress also demonstrates the equivalent of spatial and spectral ones (Balcilar et al. 2020). Many efforts have been paid to enhance GCN from the perspective of propagation (Gilmer et al. 2017), such as learnable propagation weights in Graph Attention Network (GAT) (Velickovic et al. 2018), Gated Attention Network (GaAN) (Zhang et al. 2018) and Probabilistic GCN (Yang et al. 2020), structural neighbourhood in Geom-GCN (Pei et al. 2020) and multi-scale (multi-hop) combination in N-GCN (Abu-El-Haija et al. 2019a), MixHop (Abu-El-Haija et al. 2019b), LanczosNet (Liao et al. 2019) and Krylov GCN (Luan et al. 2019). The common philosophy of them is: *"Propagation is the essence of the GCNNs"*. And, the success of GCNs attributes to the Laplacian smoothing induced by the propagation (Li, Han, and Wu 2018).

Unfortunately, the most serious issue of GNNs is the over-smoothing, which makes the performance dramatically drop, caused by the multiple propagations via stacking multiple graph convolution layers. Recently, (Oono and Suzuki 2020) shows the the exponential loss of expressive power of GNNs by generalizing the forward propagation of a GCN as a specific dynamical system. To prevent over-smoothing, two kinds of methods are proposed. On one hand, methods in the first category constrain the propagation. Disentangled GCN (Ma et al. 2019) makes each attribute only be propagated on part of the edges. DropEdge (Rong et al. 2020) randomly removes a certain number of edges from the input graph at each training epoch to reduce the adverse effect of message passing. On the other hand, methods in the second category constrain the propagation result with the original attributes. PageRank-GCN (Klicpera, Bojchevski, and Günnemann 2019) integrates personalized PageRank to GCN to combine the original attribute. JKNet (Xu et al. 2018) employs dense connections for multi-hop message passing, while DeepGCN (Li et al. 2019) and (GCNII) (Chen et al. 2020) incorporates residual layers into GCNs to facilitate the development of deep architectures. However, the perspective of propagation can't provide an intuitive and unified interpretation to their effect on preventing over-smoothing.

[*]Corresponding author.

Table 2: Comparison on transductive node classification in terms of AC (%).

Methods	Cora	Citeseer	Pubmed	Texas	Cornell	Wisconsin	Chameleon
GCN	81.5	71.1	79.0	52.1	52.7	45.8	28.2
GAT	83.1	70.8	78.5	58.3	54.3	49.4	42.9
PR-GCN	83.3	71.8	80.1	65.4	73.5	69.0	54.3
JKNet	81.1	69.8	78.1	56.4	57.3	48.8	60.1
DropEdge	83.5	72.7	79.5	57.8	61.6	50.2	61.7
GCNII	85.5	73.4	80.2	69.4	74.8	74.1	60.6
GCC	86.1	**74.3**	**81.1**	71.15	76.44	**75.37**	**61.95**
GCA	**86.3**	73.6	**81.1**	**71.62**	**76.72**	74.87	61.27

Table 3: Results on PPI.

Methods	PPI
GraphSAGE	61.27
VR-GCN	97.80
GAT	97.32
JKNet	97.61
GeniePath	98.52
Cluster-GCN	99.33
GCNII	99.53
GCC	**99.60**
GCA	99.58

nodes and edges represent web pages and hyperlinks, respectively. Node features are the bag-of-words representation of web pages. The web pages were manually classified into the five categories. Chameleon is a page-page network on specific topics in Wikipedia, where nodes, edges and features have the similar meaning as in WebKB. For inductive learning task, 24 Protein-Protein Interaction (PPI) networks are employed (Hamilton, Ying, and Leskovec 2017). Dataset statistics are summarized in Table 1.

Baselines: For transductive learning task, the baselines fall into two categories. GCN (Kipf and Welling 2017) and GAT (Velickovic et al. 2018) are two basic models, which may induce over-smoothing and overfitting. Besides, other 5 recently proposed method to overcome the oversmoothing issue are employed. They are PR-GCN (Klicpera, Bojchevski, and Günnemann 2019), JKNet (Xu et al. 2018), DropEdge (Rong et al. 2020) and GCNII (Chen et al. 2020). Note that although edge dropping strategy in (Rong et al. 2020) can be applied to many other basic model, such as GCN, DropEdge is used to represent to the combination of edge dropping and IncepGCN proposed in (Rong et al. 2020). IncepGCN is the extension of inception network (Szegedy et al. 2016) to GNN by combining 1-hop, 2-hop and 3-hop graph convolutional operations in one IncepGCN layer. For inductive learning, in additional to the GAT (Velickovic et al. 2018), JKNet (Xu et al. 2018) and GCNII (Chen et al. 2020), other 4 state-of-the-art methods, i.e., GeniePath (Liu et al. 2019), Cluster-GCN (Chiang et al. 2019) GraphSAGE (Hamilton, Ying, and Leskovec 2017) and VR-GCN (Chen, Zhu, and Song 2018), are employed.

Parameter Setting: Adam SGD optimizer (Kingma and Ba 2015) is adopted with learning rate as 0.001. Besides, early stopping with a patience of 100 epochs and ℓ_2 regularization (0.0006) is employed to prevent overfitting. $\gamma_t = 0.1$ and $\kappa_t = 0.2$ for transductive learning, while $\gamma_t = 0.45$ and $\kappa_t = 0.32$ in inductive learning. Similar to GCNII (Chen et al. 2020) identity mapping is employed to enhance the learnable mapping $\mathbf{W}$. The number of layers (depth) is selected from 8, 16 and 32. Its impact on performance will be investigated in the last subsection.

Experimental Results Analysis

Transductive Learning: The fixed split for training, validation and testing introduced in (Yang, Cohen, and Salakhutdinov 2016), i.e., 20 nodes per class for training, 500 nodes for validation and 1,000 nodes for testing, are adopted for three citation network Cora, Citeseer, and Pubmed. For each webpage network, i.e., Chameleon, Texas, Cornell and Wisconsin, nodes in each class is randomly split into 60%, 20%, and 20% for training, validation and testing. The results in term of accuracy (AC) are shown in Table 2. It demonstrates that our proposed GCC and GCA consistently outperform the state-of-the-art GNNs. They possess the ability to extract more information from high-order neighbourhoods. The performance improvement mainly due to the fast convergence of the conjugate part in GCC. From propagation perspective, it prevents the over-smoothing caused by inaccurate approximation. The impacts of the depth on performance are given in the Appendix.

Inductive Learning: Following (Velickovic et al. 2018), The 24 graphs are divided to 20 graph for training, 2 graphs for validation and 2 graphs for testing. The results in terms of F1 score are shown in Table 3. GCC defeats other state-of-the-art with 8 graph conjugate convolutional layers. It illustrates that GCC can effectively combine multi-hop information with efficient graph conjugate convolutional operation. This efficiency may attributes to that graph conjugate convolution integrates the gained information (difference between output and input) in each graph convolutional layer as shown in Figure 1(c).

Conclusions

In this paper, the propagation, which induces oversmoothing issue in Graph Convolutional Network (GCN) and its variant, is investigated. To this end, an intuitive Graph Representation Learning (GRL) framework, which simply constrains the node representation similar with the original attribute, and encourages the connected nodes possess similar representations (pairwise constraint), is presented. Based on GRL, we show that the propagation as well as its weight learning are not the essence of the GCNs, but induced by the numerical optimization of pairwise similarity requirement. Thus, inspired by the superiority of conjugate gradient descent compared to common gradient descent, a novel Graph Conjugate Convolutional (GCC) network, which adopts the obtained information of the last layer as the input to the next layer, is presented. Extensive experiments on transductive and inductive semi-supervised node classification task shows that GCC can enjoy the deep network via effectively and efficiently multi-hop information combination.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61972442, Grant U1936208 and Grant 61802282, in part by the Key Research and Development Project of Hebei Province of China under Grant 20350802D and 20310802D; in part by the Natural Science Foundation of Hebei Province of China under Grant F2020202040, in part by the Hebei Province Innovation Capacity Enhancement Project under Grant 199676146H, in part by the Natural Science Foundation of Tianjin of China under Grant 20JCYBJC00650, and in part by the Key Program of the Chinese Academy of Sciences under Grant QYZDB-SSW-JSC003.

References

Abu-El-Haija, S.; Kapoor, A.; Perozzi, B.; and Lee, J. 2019a. N-GCN: Multi-scale Graph Convolution for Semi-supervised Node Classification. In *UAI*, 310.

Abu-El-Haija, S.; Perozzi, B.; Kapoor, A.; Alipourfard, N.; Lerman, K.; Harutyunyan, H.; Steeg, G. V.; and Galstyan, A. 2019b. MixHop: Higher-Order Graph Convolutional Architectures via Sparsified Neighborhood Mixing. In *ICML*, volume 97, 21–29.

Balcilar, M.; Renton, G.; Héroux, P.; Gauzere, B.; Adam, S.; and Honeine, P. 2020. Bridging the Gap Between Spectral and Spatial Domains in Graph Neural Networks. *arXiv preprint arXiv:2003.11702* .

Chen, J.; Zhu, J.; and Song, L. 2018. Stochastic Training of Graph Convolutional Networks with Variance Reduction. In *ICML*, 941–949.

Chen, M.; Wei, Z.; Huang, Z.; Ding, B.; and Li, Y. 2020. Simple and Deep Graph Convolutional Networks. In *ICML*.

Chiang, W.; Liu, X.; Si, S.; Li, Y.; Bengio, S.; and Hsieh, C. 2019. Cluster-GCN: An Efficient Algorithm for Training Deep and Large Graph Convolutional Networks. In *SIGKDD*, 257–266.

Decell, Jr, H. P. 1965. An application of the Cayley-Hamilton theorem to generalized matrix inversion. *SIAM review* 7(4): 526–528.

Defferrard, M.; Bresson, X.; and Vandergheynst, P. 2016. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In *NIPS*, 3837–3845.

Geman, S.; and McClure, D. 1987. Statistical methods for tomographic image reconstruction. *Bulletin of the International Statistical Institute* 52(4): 5–21.

Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; and Dahl, G. E. 2017. Neural Message Passing for Quantum Chemistry. In *ICML*, 1263–1272.

Golub, G. H.; and Van Loan, C. F. 1996. *Matrix Computations (3rd Ed.)*. USA: Johns Hopkins University Press. ISBN 0801854148.

Hamilton, W. L.; Ying, Z.; and Leskovec, J. 2017. Inductive Representation Learning on Large Graphs. In *NIPS*, 1024–1034.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *CVPR*, 770–778.

Jin, D.; Liu, Z.; Li, W.; He, D.; and Zhang, W. 2019. Graph Convolutional Networks Meet Markov Random Fields: Semi-Supervised Community Detection in Attribute Networks. In *AAAI*, 152–159.

Jin, D.; Song, X.; Yu, Z.; Liu, Z.; Zhang, H.; Cheng, Z.; and Han, J. 2021. BiTe-GCN: A New GCN Architecture via Bidirectional Convolution of Topology and Features on Text-Rich Networks. In *WSDM*.

Jin, D.; Yu, Z.; He, D.; Yang, C.; Yu, P. S.; and Han, J. 2020. GCN for HIN via Implicit Utilization of Attention and Meta-paths. *arXiv preprint arXiv:2007.02643* .

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.

Klicpera, J.; Bojchevski, A.; and Günnemann, S. 2019. Predict then Propagate: Graph Neural Networks meet Personalized PageRank. In *ICLR*.

Li, G.; Müller, M.; Thabet, A. K.; and Ghanem, B. 2019. DeepGCNs: Can GCNs Go As Deep As CNNs? In *ICCV*, 9266–9275. doi:10.1109/ICCV.2019.00936.

Li, Q.; Han, Z.; and Wu, X. 2018. Deeper Insights Into Graph Convolutional Networks for Semi-Supervised Learning. In *AAAI*, 3538–3545.

Liao, R.; Zhao, Z.; Urtasun, R.; and Zemel, R. S. 2019. LanczosNet: Multi-Scale Deep Graph Convolutional Networks. In *ICLR*.

Liu, Z.; Chen, C.; Li, L.; Zhou, J.; Li, X.; Song, L.; and Qi, Y. 2019. GeniePath: Graph Neural Networks with Adaptive Receptive Paths. In *AAAI*, 4424–4431.

Luan, S.; Zhao, M.; Chang, X.; and Precup, D. 2019. Break the Ceiling: Stronger Multi-scale Deep Graph Convolutional Networks. In *NeurIPS*, 10943–10953.

Ma, J.; Cui, P.; Kuang, K.; Wang, X.; and Zhu, W. 2019. Disentangled Graph Convolutional Networks. In *ICML*, 4212–4221.

Oono, K.; and Suzuki, T. 2020. Graph Neural Networks Exponentially Lose Expressive Power for Node Classification. In *ICLR*.

Pei, H.; Wei, B.; Chang, K. C.-C.; Lei, Y.; and Yang, B. 2020. Geom-GCN: Geometric Graph Convolutional Networks. In *ICLR*.

Rong, Y.; Huang, W.; Xu, T.; and Huang, J. 2020. DropEdge: Towards Deep Graph Convolutional Networks on Node Classification. In *ICLR*.

Sen, P.; Namata, G.; Bilgic, M.; Getoor, L.; Gallagher, B.; and Eliassi-Rad, T. 2008. Collective Classification in Network Data. *AI Magazine* 29(3): 93–106.

Shah, S. A.; and Koltun, V. 2017. Robust continuous clustering. *PNAS* 114(37): 9814–9819.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the Inception Architecture for Computer Vision. In *CVPR*, 2818–2826.

Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *ICLR*.

Wu, F.; Jr., A. H. S.; Zhang, T.; Fifty, C.; Yu, T.; and Weinberger, K. Q. 2019. Simplifying Graph Convolutional Networks. In *ICML*, 6861–6871.

Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; and Yu, P. S. 2020. A comprehensive survey on graph neural networks. *IEEE TNNLS* 1–14.

Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2019. How Powerful are Graph Neural Networks? In *ICLR*.

Xu, K.; Li, C.; Tian, Y.; Sonobe, T.; Kawarabayashi, K.; and Jegelka, S. 2018. Representation Learning on Graphs with Jumping Knowledge Networks. In *ICML*, 5449–5458.

Yang, L.; Guo, Y.; Gu, J.; Jin, D.; Yang, B.; and Cao, X. 2020. Probabilistic Graph Convolutional Network via Topology-Constrained Latent Space Model. *IEEE Transactions on Cybernetics* 1–14.

Yang, L.; Kang, Z.; Cao, X.; Jin, D.; Yang, B.; and Guo, Y. 2019a. Topology Optimization based Graph Convolutional Network. In *IJCAI*, 4054–4061.

Yang, L.; Wu, F.; Gu, J.; Wang, C.; Cao, X.; Jin, D.; and Guo, Y. 2020. Graph Attention Topic Modeling Network. In *WWW*, 144–154.

Yang, L.; Wu, F.; Wang, Y.; Gu, J.; and Guo, Y. 2019b. Masked Graph Convolutional Network. In *IJCAI*, 4070–4077.

Yang, Z.; Cohen, W. W.; and Salakhutdinov, R. 2016. Revisiting Semi-Supervised Learning with Graph Embeddings. In *ICML*, 40–48.

Zhang, J.; Shi, X.; Xie, J.; Ma, H.; King, I.; and Yeung, D. 2018. GaAN: Gated Attention Networks for Learning on Large and Spatiotemporal Graphs. In *UAI*, 339–349.