Effects of Random Errors on Graph Convolutional Networks

Shinnosuke Ando
University oh Tsukuba
sinnoando@gmail.com

Sho Tsugawa
University of Tsukuba
s-tugawa@cs.tsukuba.ac.jp

Abstract

The use of Graph Convolutional Networks (GCN) has been an emerging trend in the network science research community. While GCN achieves excellent performance in several tasks, there exists an open issue in applying GCN to real-world applications. The issue is the effects of network errors on GCN. Since real-world network data contain several types of noises and errors, GCN is desirable to be less affected by such errors. However, the effects have not been sufficiently evaluated before. In this paper, we analyze the effects of random errors on GCN through extensive experiments. The results show that the node classification accuracy of GCN is decreased only 5% even when 50% of the edges are randomly increased or decreased. Moreover, in terms of false labels, the accuracy of node classification is decreased only 10% even when 20% of the labels are changed.

1. Introduction

Research on network analysis, which is used to analyze large-scale and complex networks such as social networks, transaction networks, and biochemical networks, has been actively pursued [1]. In network analysis, relationships among entities in the real world are represented by a graph. Network analysis is useful for understanding the characteristics of networks, and also useful for enhancing applications such as recommendation systems.

Node classification is one of the most important tasks in network analysis. It is the problem of classifying nodes based on their attributes in a graph. Node classification algorithms predict the labels of unlabeled nodes by using the labels of the small subset of nodes and the graph structure. For instance, suppose that there exists a paper citation network, where the nodes represent papers, and a small subset of the nodes have attribute labels representing the topics of the papers. Then, the node classification task in the paper citation network is to identify the label of each node only from the graph structure and available small number of node labels. In e-commerce networks representing co-purchasing relations among users, the node classification task is to predict the customer types of users [2]. In protein interaction networks representing interaction relations among proteins, the node classification task is to predict the function of each protein [3]. Effective methods for node classification tend to be also effective for other tasks of network analysis [4].

Among several methods for node classification, Graph Convolution Networks (GCN) have achieved excellent classification accuracy in the benchmark experiments [5]. The GCN is a neural network model that enables to obtain vector representations of nodes incorporating both graph structure and node features in a semi-supervised way. In addition, the GCN model is computationally efficient.

Although GCN recorded excellent classification accuracy, the datasets used for the evaluation were clean datasets. GCN have been evaluated on the benchmark datasets that have been used in the evaluation of many papers [5, 6, 7]. These datasets have been cleaned in previous studies by deleting nodes that occur infrequently and by removing nodes that can be noises for the analyses (e.g., nodes representing stemming and stop words in knowledge graphs) [7, 8]. However, real-world networks are different from the clean networks in these datasets. Typically, the graphs used for network analyses will contain multiple errors because it is not easy to accurately and completely identify the entities to be analyzed and the appropriate relationships among them. For instance, graphs used in social network analysis can contain several errors of various types, such as missing links, false links, etc [9]. Moreover, fake users and malicious users who conduct adversarial attacks may introduce errors to the networks [10, 11].

Some previous studies have investigated the effects of network errors on GCN, but all of them limit their

URI: https://hdl.handle.net/10125/79736
978-0-9981331-5-7

focuses on the errors due to adversarial attacks [10, 12]. Here, attacks to a network with an explicit intention to reduce the classification accuracy of GCN are called as adversarial attacks. Zügner et al. [10] show that a malicious user can significantly degrade the classification accuracy of GCN by introducing small amount of adversarial perturbations to features of a small number of nodes. Although such adversarial attacks have significant negative impacts on GCN, conducting such attacks to network data is not easy. Therefore, the vulnerability of the GCN against adversarial attacks does not necessarily imply that GCN are useless in real-world applications. In contrast, as described above, real-world networks typically contain non-adversarial errors. We call errors contained in a network without explicit intentions to reduce the classification accuracy of GCN as non-adversarial errors. For instance, spam users who follow a huge number of unrelated users may introduce noises to the network, but the spam users do not have intentions to reduce the accuracy of node classification. We call such noises as non-adversarial errors. In the literature, while the effects of errors due to adversarial attacks on GCN have been investigated, the effects of non-adversarial errors have not.

The purpose of this paper is to reveal the effects of various noises in the real world, and to clarify how effective the GCN is under such noisy situations. Contrary to the existing studies investigating the effects of adversarial errors on GCN, we focus on non-adversarial ones such as the presence of fake users in social networks and measurement errors in data collection and coding [9]. These noises are represented by randomly adding or deleting links and changing labels. We construct noisy graphs by performing these operations to clean benchmark graphs, and investigate the node classification accuracy of the GCN models trained with the noisy graphs.

The remainder of this paper is organized as follows. Section 2 introduces related work. In Section 3, we present the datasets and research methodology. Section 4 examines the accuracy of node classification against three types of errors (i.e., missing links, false links, and false labels). In Section 5, we discuss the effects of network noises on GCN and their applicability to real-world applications based on the experimental results. Finally, Section 6 contains our conclusions and a discussion of future work.

2. Literature Review

2.1. Node classification

For the node classification problem, early studies have typically used label propagation [13] and graph laplacian regularization such as manifold regularization [14]. After the appearance of skip-grams [15], researchers have focused on models for learning graph embedding, in which the features of nodes are represented by vectors. Graph embedding can be applied to a variety of tasks, and is not limited to node classification problems. DeepWalk [6] treats short series of data obtained from random walks on a graph as input to a skip-gram, and learns graph embedding from predictions of nodes that have close relationships with the nodes. While DeepWalk considers only nodes that are connected by edges, LINE [16] utilizes the information of nodes that are not directly connected by edges. In node2vec [17], breath-first search is used instead of random walk. While these embedding methods use only the graph structure, Planetoid [7] realized an embedding method that uses both the graph structure and label information.

In recent years, with the development of deep learning, methods using graph-based neural networks (GNNs) have been proposed. Although all methods based on deep learning have recorded high classification accuracy, they are computationally expensive [18] and require complex preprocessing [19], which impose significant limitations when considering their use in actual services. GCN [5] was introduced by Bruna et al. [20] and extended by Defferrard et al. [21]. GCN is based on a spectral graph convolutional neural network that performs local convolution at high speed. By introducing many simplifications to the conventional framework, GCN achieves higher classification performance than conventional methods and can be used for large networks. In this research, we verify the effects of network errors on GCN, which is regarded as one of the state-of-the-art node classification methods.

2.2. Network noises

Network analysis has suffered for a long time from the network noises. Early studies that focused on sociometric tests have been affected by the noises due to respondent bias [22], non-response [23], and questionnaire design [24, 25]. In recent years, there has been increasing research analyzing networks in online communities [26, 27, 28, 29]. In online social networks, there exist spy users using spy scripts. A spy user is an account that imitates human activities but has inconsistent behavior because it is not operated by a real person. In Facebook, an independent study [29] revealed that 27% of all accounts are fake users, but given the amount of data in these large online communities, it is difficult to filter them out completely [29]. When using a real-world network, network noise is always included in it. Network noise occurs not only in the network itself, but also when extracting datasets for graph analysis from the network. The noise generated in the extraction of the dataset is caused by coding errors in the extraction. Wang et al. [30] systematically summarized that there are six types of measurement errors in data collection: missing nodes, false nodes, missing edges, false edges, wrongly accumulated nodes, and wrongly failed nodes. In this study, we focused on the increase of spying users by spying scripts and the measurement errors during data collection.

2.3. Effects of network errors on graph neural networks

Existing studies on the effects of network errors on graph neural networks have focused on adversarial attacks [10, 12]. Adversarial attacks introduce small imperceptible changes to the input data of machine learning models so that the accuracy of the models will be significantly degraded. Many studies have investigated the effects of adversarial attacks on deep neural networks, mainly for the task of image classification.[31, 32, 33]. It is also known that graph neural networks and node embedding techniques are highly vulnerable to adversarial attacks. [34, 33].

Zügner et al. [10] propose an attack algorithm called NETTACK that introduces small perturbations to the structure and node features in attributed graphs. The algorithm generates unnoticeable perturbations by preserving the degree distribution of the graph and features co-occurrences. The performance of NETTACK on attacking GCN shows that it can successfully fool GCN and lead to a lot of misclassification of the target node. In another study, Dai et al. [12] proposed a reinforcement-learning-based attack strategy that generates structural perturbations with full or limited information about the target classifier. Their approach has shown to be successful for degrading the accuracy of supervised node classification models. Studies by other groups investigate the effects of adversarial attacks on unsupervised node embeddings [10, 12, 35]. In [10], they transferred their attack model to DeepWalk embedding [6] and observed that the performance of DeepWalk drops on a perturbed graph.

In this study, we focus on non-adversarial attacks. For instance, graphs used in social network analysis can contain several errors of various types, such as missing links, false links, etc [9]. These errors are not adversarial to node classification models. There are also noises due to spamming scripts generating fake user activities in online communities [26, 36, 28]. These *spam-users* mimic human online activity, which is often impossible to filter completely given the amount of data in large online communities [37]. For example, a report [29] suggests that 27% of all Facebook accounts are fake. These noises due to spam and fake users are typically also not adversarial. This study aims to answer the following question: Is GCN vulnerable even against such non-severe errors?

3. Data and Method

3.1. Datasets

Following Kipf et al. [5], we compare the effects of different types of random errors on GCN by simulating them in four empirical datasets. Dataset statistics are summarized in Tab. 1. We used four datasets: Citeseer, Cora, and Pubmed [8] for citation network datasets, and NELL [7, 38] for bipartite graph dataset from a knowledge graphs. In Tab. 1, features denote the number of dimensions of the feature vector of each node, and label rate is the number of labeled nodes used for training divided by the total number of nodes in the dataset. We use these datasets following the experimental settings by Kipf et al. [5], which are common among studies on GCNs to evaluate their performance. Evaluating the effects of errors on GCN using other networks such as social networks obtained from social media remains a future work.

Citeseer, Cora, and Pubmed contain sparse bag-of-words feature vectors for each document and a list of citation links between documents. The label represents the field to which the document belongs.

The knowledge graph dataset, NELL [38], is a network that represents the connections between information revealed through the analysis of hundreds of millions of web pages. Information here is a word or phrase, such as "Tokyo" and "Japan". While analyzing a web page, the system determines that "Tokyo" and "Japan" have a "city-country" relationship, and connects these nodes with edges. The edges are given labels such as "city-country". In this way, NELL is constructed as a graph with a set of nodes and labeled edges among them. In this experiment, we conducted pre-processing to the NELL dataset following Yang et al. [7]. In the preprocessing, we extract entities and relations between them from the NELL knowledge base to construct the

Table 1. Dataset statistics, as reported in Kipf et al. [5]

Dataset	Type	Nodes	Links	Classes	Features	Label rate
Citeseer	Citation network	3,327	4,732	6	3,703	0.036
Cora	Citation network	2,708	5,429	7	1,433	0.052
Pubmed	Citation network	19,717	44,338	3	500	0.003
NELL	Knowledge graph	65,755	266,144	210	5,414	0.1

knowledge graph. Bag-of-words features extracted from the ClueWeb09 [1] dataset are used as node features.

3.2. Methodology

Our goal is to investigate the effects of three different error scenarios on GCN. The three different error scenarios are: adding random links, removing random links, and changing random labels. We express the decrease in the activity levels of users by random link deletion, the increase of the inconsistent user behavior by random link addition, and the increase in the number of spy users by label changes. The process of simulating an error scenario is as follows:

1. Obtain a clean graph $G = (V, E)$ from the dataset.
2. Introduce errors to G following one of the three scenarios, and obtain a perturbed network $G' = (V', E')$.
3. Using G' as an input graph, train a model to predict the label of each node using GCN, and evaluate the classification accuracy of the model.

For the GCN setup, we closely follow the experimental setup of Kipf et al. [5]. We train a two-layer GCN and evaluate prediction accuracy on a test set of 1,000 labeled examples. We chose the same dataset splits as Kipf et al. [5] with an additional validation set of 500 labeled examples for hyperparameter optimization. We do not use the validation set labels for training. For the citation network datasets, we optimize hyperparameters on Cora only and use the same set of parameters for Citeseer and Pubmed. The values of the set of hyperparameters are shown in Tab. 2. In contrast to previous experiments [5], we do not set a window size, and train for 200 epochs in all cases. This is because with a window size, it is difficult to compare the case where learning ends early due to large noise and the case where learning continues until the end. We initialize weights using the initialization described in Glorot et al. [39] and accordingly row-normalize input feature vectors. The experiment was conducted 10 times for one error strength of an error scenario, and the average of the classification accuracies was used as the classification accuracy in that condition.

4. Experimental Results

In this Section, we present the experimental results for three different error scenarios based on the methods presented in Section 3. Figure 1 illustrates the scenarios.

4.1. Effects of random link deletion

At first, we investigated the classification accuracy of GCN models under the missing link scenario. We randomly deleted fraction p of links in the original graph G, and obtained perturbed graph G'. Then, classification accuracy of the GCN model constructed from perturbed graph G' was investigated while changing the fraction p of link deletion. Figure 2 shows the classification accuracy against the fraction p of deleted links for each graph. The results for $p = 0$ are equivalent to the results when using the original graphs, whereas the results for $p = 1$ are equivalent to the results when only node features are used for classification and network structures are completely ignored.

From the results, we can find that the classification accuracy still achieves more than 60% even when all links are removed, although the classification accuracy decreases as the link deletion fraction p increases for all datasets. The difference between the classification accuracy at $p = 0$ and the classification accuracy at $p = 1$ indicates the influence of the existence of links in the

Table 2. Sets of hyperparameters

Dataset	Citeseer, Cora, Pubmed	NELL
Dropout rate	0.5	0.1
L2 regularization	$5 \cdot 10^{-4}$	$1 \cdot 10^{-5}$
Number of hidden units	16	64

[1] https://lemurproject.org/clueweb09/

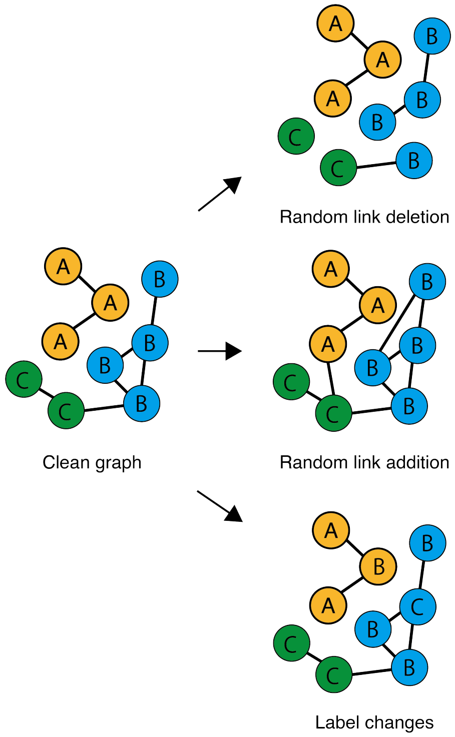

Figure 1. Three different error scenarios.

original network on the classification accuracy of GCN. The influence of the links on the classification accuracy is the lowest in the Pubmed dataset. The difference in classification accuracy between the original graph and the no-links graph is only approximately 7%. In contrast, for the Cora dataset, that is approximately 10%, which is the highest among the four datasets.

4.2. Effects of random link addition

Next, we investigated the classification accuracy of GCN under the false link scenario. We randomly extracted $|E|p$ pairs from the node pairs without links in the original graph G, and then, created links between them to obtain the perturbed graph G'. Note that p is a parameter for controlling the number of false links. Then, classification accuracy of the GCN model constructed from perturbed graph G' was investigated while changing the fraction p of link addition. Figure 3 shows the classification accuracy against the fraction p of added links for each graph.

From the results, we can find that additional false links degrade the classification accuracy of GCN. The decrease in accuracy when $|E|$ false links are added is approximately 0.1 for all datasets. In addition, it is suggested that the effects of missing links on the classification accuracy of GCN are slightly larger than those of false links when comparing the same number of links are added and removed (see Figs. 2 and 3).

In order to clarify the effects of false links and existing true links, we conducted an experiment of investigating the classification accuracy of GCN when false links were added while all original true links were removed from the networks. By comparing the results with previous results (Fig. 3), we examine the contributions of original true links to the classification accuracy of GCN. Figure 4 shows the relation between the fraction of additional links p and the classification accuracy of GCN when all original links are removed.

From the results, for citation networks, we can find that the effects of false links are larger for the cases where all original links are remove (Fig. 4) than for the cases where all original links exist (Fig. 3). In contrast, for the NELL dataset, the effects of false links are not so different for both cases.

4.3. Effects of label changes

Finally, we investigated the classification accuracy of GCN models under the false label scenario. We randomly changed fraction p of node labels in the original graph G, and obtained perturbed graph G'. Then, classification accuracy of the GCN model constructed from perturbed graph G' was investigated while changing the fraction p of label changes. Figure 5 shows the relation between the fraction p of mislabeled nodes and the classification accuracy of GCN for each dataset. In this experiment, labels were changed to other existing labels. We can see that for all datasets, an increase in the number of mislabeled nodes results in a roughly linear decrease in classification accuracy. However, compared to the results for the NETTACK [10], the effects of random label changes are much smaller than those of the NETTACK.

Figure 6 shows the results of the classification accuracy due to the increase in mislabeling when the chosen label is changed to a newly added class that does not exist in the original network. We can see that when a new label is created, the decrease in classification accuracy is larger than that of changing to other existing labels.

5. Discussion

For random edge deletion, our results show that the classification accuracy of GCN decreases almost linearly as the number of missing links increases, and the decrease in accuracy is small. For the Citeseer and Cora datasets, we recorded higher classification accuracy than non-GCN baseline reported in [7], even when 50% of the total links are deleted. We can

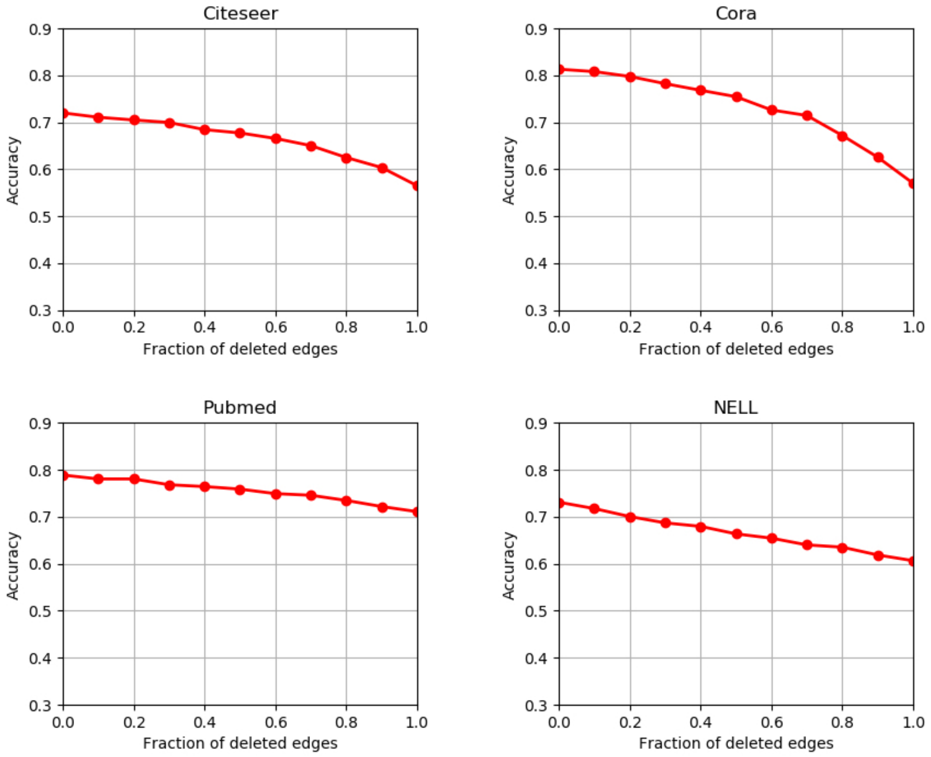

Figure 2. Classification accuracy of GCN under the missing link scenario for each dataset.

see that even with few links, GCN records high classification accuracy if the node features are rich and accurate. The number of links in a social network and e-commerce network indicates the activity level of the users participating in that network. Links in a social network represent users' actions of following other users, and links in an e-commerce network represent users' actions of purchasing products. Therefore, a network with a large number of edges is considered to have a large number of active users whereas a network with a small number of edges is not. From the results of random edge deletion, we can expect that GCN will work even for the networks with small number of edges (i.e., networks of non-active users), as long as the existing edges are reasonable.

As shown in Fig. 3, the impact of random addition of links on GCN models is as small as that of random link deletion. Random additional links in a graph may connect dissimilar nodes, which we consider as false links. A false link in a social network represents an inconsistent action of a user. For instance, a social media user typically follows users with similar interests, but may eventually follow users with dissimilar interests. Such inconsistent actions of users are represented as false links in a social graph. From the results of random link addition, we can expect that in a situation where there is enough user behavior, GCN works for the network even if the users make a certain amount of inconsistent actions.

On the other hand, as shown in Fig. 4, after all the links in the original network have been removed, random edge addition can significantly reduce the classification accuracy of GCN. This indicates that the links in the original network are necessary for preserving the accuracy of GCN. It is also suggested that we need to pay attentions to the occurrence of user's inconsistent actions when the number of links in the original network is small, i.e., when there are few user actions. However, when using the NELL dataset, even after deleting all the edges, there is no particular reduction in classification accuracy due to the addition of random edges. This may be due to the fact that NELL is very different from other datasets as shown in Tab. 1. It is beyond the scope of this paper to determine which specific feature of the NELL affects the results.

As shown in Figs. 5 and 6, the classification accuracy decreased as the number of changed labels increases. A node with a changed label in a network represents a node that has links with other nodes, but most of the links are inconsistent with the label. This is because the characteristics of the node itself are expressed in its label, so changing only the label of a node will make its

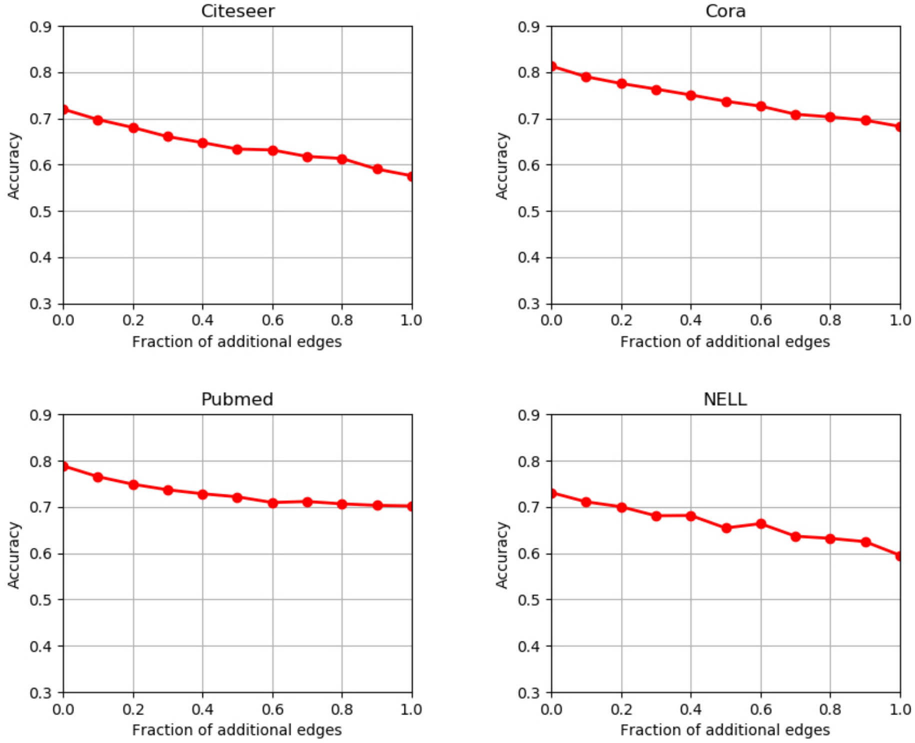

Figure 3. Classification accuracy of GCN under the false link scenario for each dataset.

links inconsistent with the label. Nodes with changes labels are considered to be similar to spam users in the real world. A spam-user is a user who behaves like a real user, but whose behavior differs greatly from his own characteristics. If we assume that the relabeled nodes are spam-users in the real world, it is difficult to assume that 50% of the users in the network are occupied by spam-users. However, as shown in the related research, they may occupy 30% of the network, so it is necessary to take sufficient countermeasures against spam-users when applying GCN.

6. Conclusion and Future Works

We analyzed the effects of random noises on GCN, missing links, false links, and false labels. We extended the scope of the noised from adversarial attacks studied in existing studies to more diverse types of noises, which aimed to clarify the applicability of GCN to real world applications. Experiments on four different datasets showed the classification accuracy of the GCN under the three error scenarios.

In this paper, the effects of some noises on GCN are clarified. Our results suggest that the application of GCN may be effective in the real-world applications. On the other hand, there still remain some open issues that should be addressed in future research. First, the relation between the effects of errors on GCN in networks and their structural characteristics should be clarified. Second, experiments on actual applications of GCN are also necessary in the future work. Third, improving the classification accuracy of GCN models in noisy networks is also an important future work. This study focuses to understand the effects of noises on GCN, but improving GCN for noisy networks is also important when using GCN in the real-world applications.

Acknowledgments

This work was partly supported by Support Center for Advanced Telecommunications Technology Research, Foundation (SCAT).

References

[1] J. Davidson, B. Liebald, J. Liu, P. Nandy, T. Van Vleet, U. Gargi, S. Gupta, Y. He, M. Lambert, B. Livingston, and D. Sampath, “The youtube video recommendation system,” in *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys ’10, p. 293–296, 2010.

[2] E. W. Ngai, L. Xiu, and D. C. Chau, “Application of data mining techniques in customer relationship management: A literature review and classification,” *Expert Systems with Applications*, vol. 36, no. 2, pp. 2592–2602, 2009.

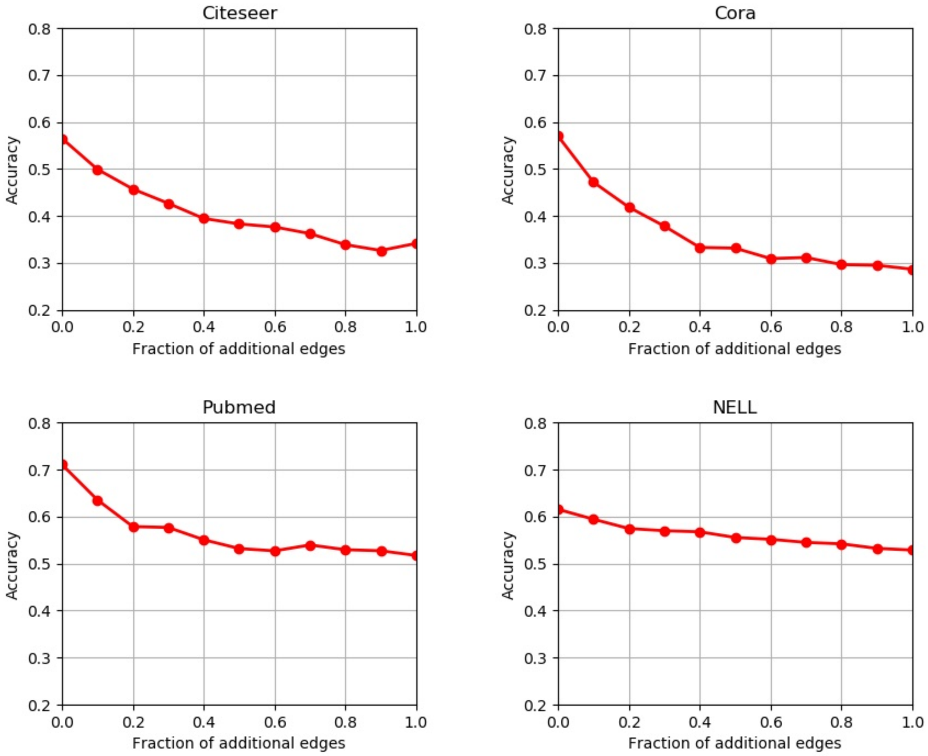

Figure 4. Classification accuracy of GCN when all existing edges in the network are removed and then edges are are randomly added in each dataset.

[3] E. Alm and A. P. Arkin, "Biological networks," *Current Opinion in Structural Biology*, vol. 13, no. 2, pp. 193–202, 2003.

[4] P. Goyal and E. Ferrara, "Graph embedding techniques, applications, and performance: A survey," *Knowledge-Based Systems*, vol. 151, pp. 78–94, 2018.

[5] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proceedings of the 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[6] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'14)*, pp. 701–710, 2014.

[7] Z. Yang, W. Cohen, and R. Salakhudinov, "Revisiting semi-supervised learning with graph embeddings," in *Proceedings of International Conference on Machine Learning*, pp. 40–48, PMLR, 2016.

[8] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad, "Collective classification in network data," *AI Magazine*, vol. 29, no. 3, pp. 93–93, 2008.

[9] D. J. Wang, X. Shi, D. A. McFarland, and J. Leskovec, "Measurement error in network data: A re-classification," *Social Networks*, vol. 34, no. 4, pp. 396–409, 2012.

[10] D. Zügner, A. Akbarnejad, and S. Günnemann, "Adversarial attacks on neural networks for graph data," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD'18)*, pp. 2847–2856, 2018.

[11] A. Bojchevski and S. Günnemann, "Adversarial attacks on node embeddings via graph poisoning," in *International Conference on Machine Learning*, pp. 695–704, PMLR, 2019.

[12] H. Dai, H. Li, T. Tian, X. Huang, L. Wang, J. Zhu, and L. Song, "Adversarial attack on graph structured data," in *Proceedings of the 35th International Conference on Machine Learning* (J. Dy and A. Krause, eds.), vol. 80 of *Proceedings of Machine Learning Research*, pp. 1115–1124, 10–15 Jul 2018.

[13] X. Zhu, Z. Ghahramani, and J. D. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pp. 912–919, 2003.

[14] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *Journal of Machine Learning Research*, vol. 7, no. 85, pp. 2399–2434, 2006.

[15] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing systems*, vol. 26, pp. 3111–3119, 2013.

[16] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *Proceedings of the 24th International Conference on World Wide Web*, pp. 1067–1077, 2015.

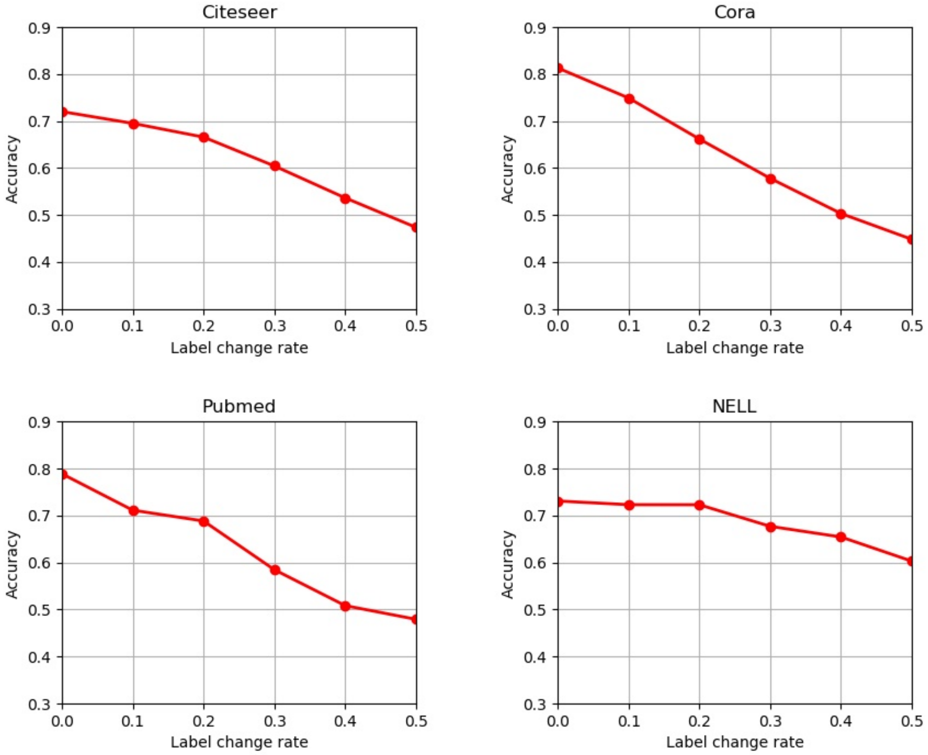

Figure 5. Classification accuracy of GCN under the false label scenario for each dataset.

[17] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'16)*, pp. 855–864, 2016.

[18] J. Atwood and D. Towsley, "Diffusion-convolutional neural networks," in *Proceedings of Advances in Neural Information Processing Systems*, pp. 1993–2001, 2016.

[19] M. Niepert, M. Ahmed, and K. Kutzkov, "Learning convolutional neural networks for graphs," in *International Conference on Machine Learning*, pp. 2014–2023, 2016.

[20] J. Bruna, W. Zaremba, A. Szlam, and Y. Lecun, "Spectral networks and locally connected networks on graphs," in *International Conference on Learning Representations (ICLR2014), CBLS, April 2014*, 2014.

[21] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proceedings of Advances in Neural Information Processing Systems*, pp. 3844–3852, 2016.

[22] I. de Sola Pool and M. Kochen, "Contacts and influence," *Social Networks*, vol. 1, no. 1, pp. 5 – 51, 1978.

[23] D. Stork and W. D. Richards, "Nonrespondents in communication network studies: Problems and possibilities," *Group & Organization Management*, vol. 17, no. 2, pp. 193–209, 1992.

[24] R. S. Burt, "Network items and the general social survey," *Social Networks*, vol. 6, no. 4, pp. 293–339, 1984.

[25] P. W. Holland and S. Leinhardt, "The structural implications of measurement error in sociometry," *Journal of Mathematical Sociology*, vol. 3, no. 1, pp. 85–111, 1973.

[26] A. Narayanan and V. Shmatikov, "De-anonymizing social networks," in *Proceedings of 30th IEEE Symposium on Security and Privacy*, pp. 173–187, IEEE, 2009.

[27] B. Nardi and J. Harris, "Strangers and friends: Collaborative play in World of Warcraft," in *Proceedings of the 20th Anniversary Conference on Computer supported Cooperative Work*, pp. 149–158, 2006.

[28] K. Lewis, J. Kaufman, M. Gonzalez, A. Wimmer, and N. Christakis, "Tastes, ties, and time: A new social network dataset using facebook. com," *Social Networks*, vol. 30, no. 4, pp. 330–342, 2008.

[29] R. Richmond, "Stolen facebook accounts for sale," *The New York Times*, vol. 2, 2010.

[30] D. J. Wang, X. Shi, D. A. McFarland, and J. Leskovec, "Measurement error in network data: A re-classification," *Social Networks*, vol. 34, no. 4, pp. 396–409, 2012.

[31] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *arXiv preprint arXiv:1607.02533*, 2016.

[32] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 1765–1773, 2017.

[33] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing

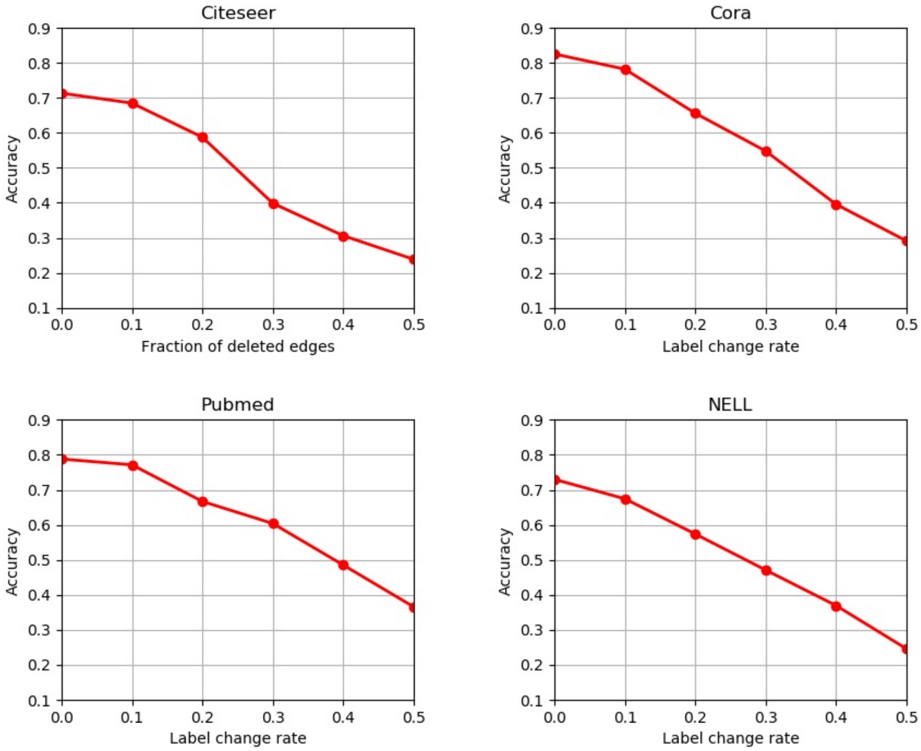

Figure 6. Classification accuracy of GCN when node labels are changed to a new class for each dataset.

properties of neural networks," in *International Conference on Learning Representations*, 2014.

[34] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *International Conference on Learning Representations*, 2015.

[35] M. Sun, J. Tang, H. Li, B. Li, C. Xiao, Y. Chen, and D. Song, "Data poisoning attack against unsupervised node embedding methods," *arXiv preprint arXiv:1810.12881*, 2018.

[36] R. Ackland, "Social network services as data sources and platforms for e-researching social networks," *Social Science Computer Review*, vol. 27, no. 4, pp. 481–492, 2009.

[37] M. Conti, R. Poovendran, and M. Secchiero, "Fakebook: Detecting fake profiles in on-line social networks," in *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pp. 1071–1078, 2012.

[38] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. Hruschka, and T. Mitchell, "Toward an architecture for never-ending language learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2010.

[39] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings 13th International Conference on Artificial Intelligence and Statistics*, pp. 249–256, 2010.