ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-7605-5/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICASSP39728.2021.9413474

ON THE STABILITY OF GRAPH CONVOLUTIONAL NEURAL NETWORKS UNDER EDGE REWIRING

Henry Kenlay[†] *Dorina Thanou*[‡] *Xiaowen Dong*[†]

[†]University of Oxford [‡]Swiss Data Science Center

ABSTRACT

Graph neural networks are experiencing a surge of popularity within the machine learning community due to their ability to adapt to non-Euclidean domains and instil inductive biases. Despite this, their stability, i.e., their robustness to small perturbations in the input, is not yet well understood. Although there exists some results showing the stability of graph neural networks, most take the form of an upper bound on the magnitude of change due to a perturbation in the graph topology. However, the change in the graph topology captured in existing bounds tend not to be expressed in terms of structural properties, limiting our understanding of the model robustness properties. In this work, we develop an interpretable upper bound elucidating that graph neural networks are stable to rewiring between high degree nodes. This bound and further research in bounds of similar type provide further understanding of the stability properties of graph neural networks.

Index Terms— Graph signal processing, graph convolutional neural networks, spectral graph filters, stability.

1. INTRODUCTION

Recently, there has been an increasing amount of research on the use of machine learning models which operate on graph-structured data [1, 2]. Graphs encode pairwise interactions and can help model non-Euclidean data such as social networks and molecules as well as impart inductive biases [3]. Graph signal processing (GSP), for example, generalises traditional signal processing to network domains allowing us to transfer many of the existing tools such as filtering and sampling [4, 5]. GSP also allows us to generalise convolutions providing a framework to design graph convolutional neural networks (GCNNs) where the convolutional layers are filter banks of spectral graph filters [6, 7, 8]. To date, much of the research has focused on the predictive performance of these models with far fewer studies looking at their theoretical properties.

Like their Euclidean counterparts, GCNNs are susceptible to adversarial attacks [9, 10, 11]. An adversarial attack is a small but targeted perturbation of the input which causes large changes in the output [12]. In the case of GCNNs, the modification of a few edges in the input graph can change significantly the learned node representations so that the prediction in the downstream task, typically node or graph classification, switches from a correct to incorrect prediction. One approach to understanding the vulnerability of these models to adversarial attacks is to consider their robustness against perturbation. Robustness is an important property for reliable deployment of machine learning models in the real world, especially in domains where adversaries are common (e.g., on the web).

Existing literature has shown that large functional classes of spectral graph filters, a key component of GCNNs, are bounded by the magnitude of the change in the input graph. The main drawback of these approaches is that the bounds involved in quantifying stability to specific changes in the topology do not have natural structural interpretations. For example, if we delete just a single edge, it is unclear how loose the bound on the output change will be even if we know the statistical properties about the edge and endpoint nodes.

Our main goal is to extend bounds found in the existing literature so that they have an interpretation in terms of the structural properties of the existing and perturbed graphs. To do this, we focus on polynomial graph filters, and build on our previous work [13], by providing a new bound that is tighter and generalises to the family of normalised augmented adjacency matrices. We then bound the change in normalised augmented adjacency matrix by considering the largest change around each node where the change admits a structural interpretation. As a specific example, we consider perturbations that do not modify the degree distribution of the graph (i.e., double edge rewiring, illustrated in Fig. 1). We then discuss under which scenarios, deleting and adding edges between nodes, guarantee the filter to be robust to perturbations. To demonstrate how these theoretical results can be combined, we bound the change in the representations learned by two well-known GCNN architectures. Specifically, we consider simple graph convolutional networks [14] and multilayered graph convolutional network [15] for scenarios where the graph topology is perturbed using edge rewiring.

2. RELATED WORK

One of the earlier works on stability was by Levie et al [16]. In this work, the authors give an upper bound of change which grows linearly in the distance between the graph shift operators before and after perturbation for a large class of spectral graph filters. The bound is based on analysis in the Cayley smoothness space. Related to this work, [13] also provides a linear bound for polynomial spectral filters, where the bound is based on Taylor expansion for matrix functions. In Sec. 4 we give a tighter bound for polynomial spectral filters via a much simpler proof.

Gama et al. prove that a class of spectral graph filters are stable to changes in the graph topology [17]. Furthermore, GCNNs using filters in this class as filter banks and ReLU nonlinearities are also stable. The main difference between this work and the work presented in this paper is on how the magnitude of perturbation is measured. In [17] the authors posit that simple additive error as a measure does not reflect the fact that the distance between isomorphic graphs can be non-zero. To address this concern they consider a relative measure of perturbation and consider all node permutations for the perturbed graph. We believe that the additive error approach of [13, 16] and this work, which does not consider node permutations, and the approach of [17] are both useful. For example, if the

This work was supported by the EPSRC Centre for Doctoral Training in Autonomous Intelligent Machines and Systems EP/L015897/1.

graph represents a polygon mesh, then the node labelling in the perturbed graph does not have any meaningful interpretation, as they are just used to construct matrix representations. In this case, considering permutations is appropriate. However, in a social network, the node labelling will typically correspond to the identification of a user, in which case it makes sense to consider the labelling as fixed between the original and perturbed graph.

Beyond the notion of stability adopted in the aforementioned studies, there are efforts to quantify alternative notions of stability for graph-based models. An input to a classification model is certifiably robust if for any perturbation, under a given perturbation model, the predicted label will not change. Methods for generating robustness certificates for nodes in semi-supervised learning tasks have recently been proposed for graph neural networks [18, 19]. One can also measure the stability of graph-based models by considering the graph topology and signal as random variables and considering the statistical properties of the model. For example, [20] proves that in stochastic time-evolving graphs, the output of the filter behaves the same as the deterministic filter in expectation. In [21], the authors show how to deal with uncertainties in the graph topology to approximate the original filtering process. In [22] stochastic graph neural networks are proposed to account for training on a stochastic graph. Unlike the existing approaches outlined in this section, to the best of our knowledge, our work is one of the first to provide sufficient conditions for stability that come with a structural interpretation.

3. PROBLEM FORMULATION

We consider unweighted and undirected graphs $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is a finite set of nodes and $\mathcal{E}$ is the edge set. The adjacency matrix $\mathbf{A}$ encodes connections between nodes with $\mathbf{A}_{uv} = 1$ if there is an edge between nodes u and v and zero otherwise. The degree d_u of a node u is the number of nodes that u is connected to. A graph signal is a function $x : \mathcal{V} \to \mathbb{R}$ that assigns a scalar value to each node. By fixing a labelling of the nodes we can represent this as a vector $\mathbf{x} \in \mathbb{R}^n$ where $\mathbf{x}_i$ is the function value of node i and $n = |\mathcal{V}|$ is the number of nodes in the graph. We will be concerned with the normalised augmented adjacency matrix $\mathbf{\Delta}_\gamma = \mathbf{D}_\gamma^{-1/2}\mathbf{A}_\gamma\mathbf{D}_\gamma^{-1/2}$ where $\mathbf{A}_\gamma = \mathbf{A} + \gamma\mathbf{I}$ and $\mathbf{D}_\gamma = \mathbf{D} + \gamma\mathbf{I}$ with $\gamma \geq 0$. When $\gamma = 0$ this matrix is the normalised adjacency matrix. For simplicity, we drop the γ subscript when the context is clear or the specific value of γ is unimportant. We can write $\mathbf{\Delta}$ in terms of its entries as

$$\mathbf{\Delta}_{uv} = \begin{cases} \frac{\gamma}{d_u+\gamma} & \text{if } u = v \\ \frac{1}{\sqrt{(d_u+\gamma)(d_v+\gamma)}} & \text{if } \mathbf{A}_{uv} = 1 \text{ and } u \neq v \\ 0 & \text{otherwise} \end{cases}. \tag{1}$$

Normalised augmented adjacency matrices admit an eigendecomposition $\mathbf{\Delta} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\intercal$ where the columns of $\mathbf{U}$ are the orthonormal eigenvectors, and $\mathbf{\Lambda} = \operatorname{diag}(\lambda_1, \dots \lambda_n)$ where $\lambda_1 \leq, \dots, \leq \lambda_n$ is the diagonal matrix of eigenvalues. The graph Fourier transform of a signal $\mathbf{x}$ is given by $\hat{\mathbf{x}} = \mathbf{U}^\intercal\mathbf{x}$, with the inverse graph Fourier transform defined as $\mathbf{x} = \mathbf{U}\hat{\mathbf{x}}$. With a notion of Fourier basis, a spectral graph filter is defined as a function $g : \mathbb{R} \to \mathbb{R}$ which amplifies and attenuates specific frequencies. We can filter a signal by applying the filter to the graph shift operator (in this case $\mathbf{\Delta}$) directly: $\mathbf{U}\operatorname{diag}(g(\lambda_1), \dots g(\lambda_n))\mathbf{U}^\intercal\mathbf{x} = \mathbf{U}g(\mathbf{\Lambda})\mathbf{U}^\intercal\mathbf{x} = g(\mathbf{\Delta})\mathbf{x}$.

In this work $\|\cdot\|_2$ represents the Euclidean norm when applied to vectors and the operator norm when applied to matrices. We will also consider the Frobenius norm $\|\mathbf{A}\|_F^2 = \sum_{i,j}\mathbf{A}_{i,j}^2$, the matrix one norm $\|\mathbf{A}\|_1 = \max_i \sum_j |\mathbf{A}_{ij}|$ and the matrix infinity norm $\|\mathbf{A}\|_\infty = \max_j \sum_i |\mathbf{A}_{ij}|$. The eigenvalues of the normalised augmented adjacency matrix lie in the interval $[-1, 1]$. Furthermore, 1 is always an eigenvalue so $\|\mathbf{\Delta}\|_2 = 1$.

Given a graph $\mathcal{G}$ and a perturbed graph $\mathcal{G}_p$ with normalised augmented adjacency matrices $\mathbf{\Delta}$ and $\mathbf{\Delta}_p$ respectively, our goal is twofold. First, we want to understand the stability of spectral graph filters in terms of structural perturbation, by providing bounds that quantify the change in the output. Second, we want to find sufficient conditions for this change to be small, by using interpretable structural properties of the graphs and the perturbation. We address the first goal in Sec. 4 and the second in Sec. 5. Furthermore, we demonstrate how these theoretical contributions can be combined to give insight into the stability of GCNNs. In Sec. 6, we combine the results of Sec. 4 and Sec. 5 to show how we can provide sufficient conditions for the stability of two popular GCNN architectures.

4. STABILITY OF POLYNOMIAL FILTERS

Our notion of stability is based on relative output distance which is bounded by the filter distance

$$\frac{\|g(\mathbf{\Delta})\mathbf{x} - g(\mathbf{\Delta}_p)\mathbf{x}\|_2}{\|\mathbf{x}\|_2} \leq \max_{\mathbf{x}\neq 0} \frac{\|g(\mathbf{\Delta})\mathbf{x} - g(\mathbf{\Delta}_p)\mathbf{x}\|_2}{\|\mathbf{x}\|_2} \stackrel{\text{def}}{=} \|g(\mathbf{\Delta}) - g(\mathbf{\Delta}_p)\|_2. \tag{2}$$

In [13] we bounded the filter distance of polynomial filters by some constant times the error $\|\mathbf{E}\|_2$, where the constant depends on the filter and the error is the magnitude of the difference between normalised Laplacian matrices. We say that by satisfying this condition the filters are stable. Although the focus of this paper is not on tight bounds, we improve this bound by finding a smaller constant for polynomial filters. To do so we will use the following Lemma.

Lemma 1 (Lemma 3, [16]). *Suppose $\mathbf{B}, \mathbf{D}, \mathbf{E} \in \mathbb{C}^{N\times N}$ are Hermitian matrices satisfying $\mathbf{B} = \mathbf{D} + \mathbf{E}$, and $\|\mathbf{B}\|_2, \|\mathbf{D}\|_2 \leq C$ for some $C > 0$. Then for every $l \geq 0$*

$$\left\|\mathbf{B}^l - \mathbf{D}^l\right\|_2 \leq lC^{l-1}\|\mathbf{E}\|_2.$$

From here on in we will write $\mathbf{E} = \mathbf{\Delta_p} - \mathbf{\Delta}$ to be the difference of normalised augmented adjacency matrices between a graph $\mathcal{G}$ and a perturbed version of the graph $\mathcal{G}_p$. The smaller constant is given by the following proposition.

Proposition 1. *Let $\mathbf{\Delta}$ and $\mathbf{\Delta}_p$ be the normalised augmented adjacency matrix for $\mathcal{G}$ and $\mathcal{G}_p$. Consider a polynomial graph filter $g_\theta(\lambda) = \sum_{k=0}^K \theta_k\lambda^k$. Then*

$$\|g_\theta(\mathbf{\Delta}) - g_\theta(\mathbf{\Delta}_p)\|_2 \leq \sum_{k=1}^K k|\theta_k|\|\mathbf{E}\|_2.$$

Proof. Using the triangle inequality followed by an application of Lemma 1 with constant $C = 1$ we get

$$\|g_\theta(\mathbf{\Delta}) - g_\theta(\mathbf{\Delta}_p)\|_2 = \left\|\sum_{k=1}^K \theta_k(\mathbf{\Delta}^k - \mathbf{\Delta}_p^k)\right\|_2 \leq \sum_{k=1}^K |\theta_k| \left\|\mathbf{\Delta}^k - \mathbf{\Delta}_p^k\right\|_2 \leq \sum_{k=1}^K k|\theta_k|\|\mathbf{E}\|_2. \square$$

In this work, we assume that the parameters of the model are fixed before and after perturbation. In the adversarial learning literature, an attack that modifies the input to cause large changes in the output whilst the model parameters are fixed is known as an evasion attack [11]. Robust models in the context of our work are those that are robust to evasion attacks with respect to the graph structure.

5. ROBUSTNESS TO EDGE REWIRING PERTURBATIONS

In this section we bound the error term $\|\mathbf{E}\|_2$ by interpretable properties relating to the structural change. Consider a graph $\mathcal{G}$ which we perturb to arrive at $\mathcal{G}_p$. Our approach to upper bounding $\|\mathbf{E}\|_2$ relies on the inequality $\|\mathbf{E}\|_2^2 \leq \|\mathbf{E}\|_1\|\mathbf{E}\|_\infty$ [23, Section 6.3]. As $\mathbf{E}$ is symmetric $\|\mathbf{E}\|_1 = \|\mathbf{E}\|_\infty$ giving $\|\mathbf{E}\|_2 \leq \|\mathbf{E}\|_1$. There may exist strategies which give tighter bounds, but the benefit of this approach to bounding the error term is that $\|\mathbf{E}\|_1$ leads to an interpretation in the structural domain. For a matrix $\mathbf{E}$ we write $\mathbf{E}_u$ as the uth column of $\mathbf{E}$ so $\mathbf{E}_u^\intercal$ is the uth row. The row $\mathbf{E}_u^\intercal$ corresponds to the node u in the graph. By definition $\|\mathbf{E}\|_1 = \max_{u\in\mathcal{V}}\|\mathbf{E}_u^\intercal\|_1$ where $\|\mathbf{E}_u^\intercal\|_1 = \sum_v|\mathbf{E}_{uv}|$ is the Manhattan norm of the row. Perturbations which cause small changes to $\|\mathbf{E}_u^\intercal\|_1$ over all nodes u guarantee small change in terms of $\|\mathbf{E}\|_2$.

The focus of this section is on developing an interpretable upper bound on $\|\mathbf{E}\|_2$. Before establishing an upper bound, we briefly mention the following lower bound to the error term

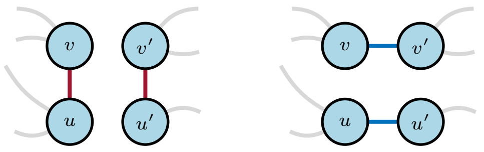
$$\max_{i,j}|\mathbf{E}_{ij}| \overset{\text{def}}{=} \|\mathbf{E}\|_{\max} \leq \|\mathbf{E}\|_2.$$

This lower bound gives us sufficient conditions for large values of the error term. For example, deleting or adding an edge such that $\sqrt{(d_u+\gamma)(d_v+\gamma)}$ is small (e.g., if both degrees are small) will cause $\|\mathbf{E}\|_2$ to be large. In these cases, our bound will be loose and cannot provide guarantees about the filter robustness.

We aim to understand what type of perturbations spectral graph filters may be robust to. As a specific example consider perturbations that preserve the degree of the nodes. For this scenario we can consider how the entries change from $\mathbf{\Delta}$ to $\mathbf{\Delta}_p$ by considering Eq. (1) to write a closed-form for $\|\mathbf{E}_u^\intercal\|_1$. We will write $\mathcal{A}_u$ to be the set of edges added around a node u and $\mathcal{D}_u$ to be the set of edges deleted around a node u. The diagonal entries of $\mathbf{\Delta}$ and $\mathbf{\Delta}_p$ remain unchanged, and each edge deletion flips the entry from zero to the second case of Eq. (1), whilst edge addition flips the entry the other way. This insight lets us write $\|\mathbf{E}_u^\intercal\|_1$ in closed-form as

$$\begin{aligned}\|\mathbf{E}_u^\intercal\|_1 &= \sum_{v\in\mathcal{D}_u}\frac{1}{\sqrt{(d_u+\gamma)(d_v+\gamma)}} + \sum_{v\in\mathcal{A}_u}\frac{1}{\sqrt{(d_u+\gamma)(d_v+\gamma)}}\\ &= \frac{1}{\sqrt{d_u+\gamma}}\left(\sum_{v\in\mathcal{D}_u}\frac{1}{\sqrt{d_v+\gamma}} + \sum_{v\in\mathcal{A}_u}\frac{1}{\sqrt{d_v+\gamma}}\right) \qquad (3)\end{aligned}$$

One such perturbation is edge rewiring that preserves degree distribution. We define double edge rewiring as a function of two edges (u,v) and (u',v') such that u is not connected to u' or v' and similarly v is not connected to u' and v'. The operation consists of deleting edges (u,v) and (u',v') and adding edges (u,u') and (v,v'). This operation is depicted graphically in Fig. 1. Although the precise definition of rewiring in this work is slightly different, the idea of rewiring has been proposed as a strategy to make modifications imperceptible in the context of topological adversarial attacks [24]. Approximately preserving the degree distribution has also been a criterion used to define imperceptibility [9]. Beyond the adversarial attack literature, double edge rewiring has been used to model changes in a network where the capacity of a node is fixed and remains at full load such as in communication networks [25].



(a) Before rewiring (b) After rewiring

Fig. 1: In the rewiring operation the red edges are deleted and the blue edges are added. The degree of each node remains the same.

We will write R_u to be the number of rewiring operations involving u and write δ_u to be the smallest degree amongst either the nodes u disconnects with or is now connected to. Each rewiring causes a single edge deletion and edge addition for each node involved so that the number of terms in each sum, the cardinality of sets $\mathcal{D}_u$ and $\mathcal{A}_u$, is R_u. Using this we can bound Eq. (3) to get that

$$\begin{aligned}\|\mathbf{E}_u^\intercal\|_1 &\leq \frac{1}{\sqrt{d_u+\gamma}}\left(\sum_{v\in\mathcal{D}_u}\frac{1}{\sqrt{\delta_u+\gamma}} + \sum_{v\in\mathcal{A}_u}\frac{1}{\sqrt{\delta_u+\gamma}}\right)\\ &= \frac{2R_u}{\sqrt{(d_u+\gamma)(\delta_u+\gamma)}}.\end{aligned}$$

The largest possible value for the right hand side of the above equation over all nodes u provides an upper bound for $\|\mathbf{E}\|_2$. From this, we can draw some conclusions as to when the filter will be robust to rewiring operations. The first is that one should not rewire around one node significantly as this will increase R_u. To keep $\|\mathbf{E}\|_1$ small, and thus $\|\mathbf{E}\|_2$ small, we must keep $\|\mathbf{E}_u^\intercal\|_1$ small for all nodes u suggesting the perturbation should be distributed across the graph. Related to this observation, in [13] it was numerically demonstrated that the locality of perturbations can play a role in the magnitude of the error. The second strategy to ensure robustness is to rewire between high degree nodes. This will cause d_u to be large and therefore $\|\mathbf{E}_u^\intercal\|_1$ will be small. Finally, using a normalised augmented adjacency matrix with larger values of γ can cause smaller changes.

6. STABILITY OF GCNN MODELS

We make use of results in previous sections to analyse the stability of two popular GCNN models, i.e., the simple graph convolutional networks (SGCN) [14] and the multilayered graph convolutional network (GCN) [15].

6.1. SGCN

The SGCN model is motivated by considering a multilayered GCN with the activation functions removed. By removing the activation functions the model boils down to a fixed monomial filter of order K followed by applying a fully connected layer and a softmax layer to the node features.

Let the input data be given by the matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$ where d is the dimension of the features associated with each node. We can consider $\mathbf{X}$ as d stacked graph signals which we will call feature maps. We will assume that each column (feature map) of $\mathbf{X}$ has unit norm. The output is a matrix $\mathbf{Y} \in \mathbb{R}^{n\times c}$, representing class probabilities for each node. The SGCN model is defined as $\mathbf{Y} = \text{softmax}(\tilde{\mathbf{\Delta}}^K\mathbf{X}\mathbf{\Theta})$ where $\text{softmax}(\mathbf{x})_i = \exp(\mathbf{x}_i)/\sum_i\exp(\mathbf{x}_i)$

normalises the rows to be probability distributions. We write $\tilde{\boldsymbol{\Delta}} \overset{\text{def}}{=} \boldsymbol{\Delta}_1$ as the normalised augmented adjacency matrix with $\gamma = 1$. In this subsection, we will analyse how the logits, the node representations before the softmax is applied, change. The softmax function has a Lipschitz constant of 1 so any bound on the logits can trivially be applied to the model outputs [26][Proposition 4]. We begin by stating the following Lemma.

Lemma 2. *Let* $\mathbf{B} \in \mathbb{R}^{m \times r}$ *and* $\mathbf{C} \in \mathbb{R}^{r \times n}$ *then*

$$\|\mathbf{BC}\|_F \leq \|\mathbf{B}\|_F \|\mathbf{C}\|_2.$$

Proof. By decomposing $\mathbf{BC}$ in terms of the rows $\mathbf{b}_k^\intercal$ of $\mathbf{B}$ we get

$$\|\mathbf{BC}\|_F^2 = \sum_{k=1}^{m} \|\mathbf{b}_k^\intercal \mathbf{C}\|_2^2 \leq \sum_{k=1}^{m} \|\mathbf{b}_k^\intercal\|_2^2 \|\mathbf{C}\|_2^2 = \|\mathbf{B}\|_F^2 \|\mathbf{C}\|_2^2. \quad (4)$$

The inequality follows from the observation that $\|\mathbf{b}^\intercal \mathbf{C}\|_2 = \|(\mathbf{b}^\intercal \mathbf{C})^\intercal\|_2 = \|\mathbf{C}^\intercal \mathbf{b}\|_2 \leq \|\mathbf{C}^\intercal\|_2 \|\mathbf{b}\|_2 = \|\mathbf{b}\|_2 \|\mathbf{C}\|_2$. Taking the square root of both sides of Eq. (4) gives the result. □

Using this we can provide a bound on how much the logits can change under the Frobenius norm. To motivate why we are interested in the Frobenius norm here, consider taking the Euclidean norm of each node output to measure the amount of change in node representation. Taking the mean squared error of these distances amounts to taking the Frobenius norm of the logits matrix.

Proposition 2. *The distance of the logits is bounded like so*

$$\left\| \tilde{\boldsymbol{\Delta}}^K \mathbf{X} \boldsymbol{\Theta} - \tilde{\boldsymbol{\Delta}}_p^K \mathbf{X} \boldsymbol{\Theta} \right\|_F \leq \sqrt{d} K \|\mathbf{E}\|_2 \|\boldsymbol{\Theta}\|_2$$

Proof. We first note that $\|\mathbf{X}\|_F = \sqrt{d}$. Using this, two applications of Lemma 2, and an application of Proposition 1 we get

$$\begin{aligned} \left\| \tilde{\boldsymbol{\Delta}}^K \mathbf{X} \boldsymbol{\Theta} - \tilde{\boldsymbol{\Delta}}_p^K \mathbf{X} \boldsymbol{\Theta} \right\|_F &\leq \left\| \tilde{\boldsymbol{\Delta}}^K \mathbf{X} - \tilde{\boldsymbol{\Delta}}_p^K \mathbf{X} \right\|_F \|\boldsymbol{\Theta}\|_2 \\ &\leq \left\| \tilde{\boldsymbol{\Delta}}^K - \tilde{\boldsymbol{\Delta}}_p^K \right\|_2 \|\mathbf{X}\|_F \|\boldsymbol{\Theta}\|_2. \\ &\leq \sqrt{d} K \|\mathbf{E}\|_2 \|\boldsymbol{\Theta}\|_2. \end{aligned}$$

□

6.2. Multilayer GCN

We now consider a multilayered GCN model. We again consider the logits of a model which this time consists of multiple GCN layers with pointwise non-linearities giving the lth layer representation as $\mathbf{X}^{(l)} = \sigma(\tilde{\boldsymbol{\Delta}} \mathbf{X}^{(l-1)} \boldsymbol{\Theta}^{(l)})$, where σ is the ReLU activation function and $\boldsymbol{\Theta}^{(l)}$ are the layer parameters. We will consider the number of feature maps to be the same throughout the model.

Proposition 3. *Let* $\mathbf{X}^{(l)} = \sigma(\tilde{\boldsymbol{\Delta}} \mathbf{X}^{(l-1)} \boldsymbol{\Theta}^{(l)})$ *where* $\mathbf{X}^{(0)} \in \mathbb{R}^{n \times d}$ *is the input feature maps,* $\boldsymbol{\Theta}^{(l)} \in \mathbb{R}^{d \times d}$ *are the weight matrices and* L *is the number of layers so* $\mathbf{X}^L$ *is the output features. Then*

$$\left\| \mathbf{X}^{(L)} - \mathbf{X}_p^{(L)} \right\|_F \leq \sqrt{d} L \|\mathbf{E}\|_2 \prod_{l=1}^{L} \left\| \boldsymbol{\Theta}^{(l)} \right\|_2.$$

Proof. We prove this by induction. The base case $L = 1$ follows immediately from Proposition 2. Consider a more general L. Then

$$\begin{aligned} \left\| \mathbf{X}^{(L)} - \mathbf{X}_p^{(L)} \right\|_F &= \left\| \sigma(\tilde{\boldsymbol{\Delta}} \mathbf{X}^{(l-1)} \boldsymbol{\Theta}^{(l)}) - \sigma(\tilde{\boldsymbol{\Delta}}_p \mathbf{X}_p^{(l-1)} \boldsymbol{\Theta}^{(l)}) \right\|_F \\ &\leq \left\| \tilde{\boldsymbol{\Delta}} \mathbf{X}^{(l-1)} \boldsymbol{\Theta}^{(l)} - \tilde{\boldsymbol{\Delta}}_p \mathbf{X}_p^{(l-1)} \boldsymbol{\Theta}^{(l)} \right\|_F \\ &\leq \left\| \tilde{\boldsymbol{\Delta}} \mathbf{X}^{(l-1)} - \tilde{\boldsymbol{\Delta}}_p \mathbf{X}_p^{(l-1)} \right\|_F \left\| \boldsymbol{\Theta}^{(l)} \right\|_2, \end{aligned}$$

where the first inequality comes from ReLU having unit Lipschitz constant and the second being an application of Lemma 2. By using triangle inequality we bound the following term

$$\begin{aligned} &\left\| \tilde{\boldsymbol{\Delta}} \mathbf{X}^{(l-1)} - \tilde{\boldsymbol{\Delta}}_p \mathbf{X}_p^{(l-1)} \right\|_F \\ &= \left\| \tilde{\boldsymbol{\Delta}} \mathbf{X}^{(l-1)} - \tilde{\boldsymbol{\Delta}}_p \mathbf{X}^{(l-1)} + \tilde{\boldsymbol{\Delta}}_p \mathbf{X}^{(l-1)} - \tilde{\boldsymbol{\Delta}}_p \mathbf{X}_p^{(l-1)} \right\|_F \\ &\leq \left\| \tilde{\boldsymbol{\Delta}} \mathbf{X}^{(l-1)} - \tilde{\boldsymbol{\Delta}}_p \mathbf{X}^{(l-1)} \right\|_F + \left\| \tilde{\boldsymbol{\Delta}}_p \mathbf{X}^{(l-1)} - \tilde{\boldsymbol{\Delta}}_p \mathbf{X}_p^{(l-1)} \right\|_F \\ &\leq \|\mathbf{E}\|_2 \left\| \mathbf{X}^{(l-1)} \right\|_F + \left\| \mathbf{X}^{(l-1)} - \mathbf{X}_p^{(l-1)} \right\|_F. \end{aligned}$$

Note that $\|\mathbf{X}^{(l)}\|_F = \|\sigma(\tilde{\boldsymbol{\Delta}} \mathbf{X}^{(l-1)} \boldsymbol{\Theta}^{(l)})\|_F \leq \|\mathbf{X}^{(l-1)}\|_F \|\boldsymbol{\Theta}^{(l)}\|_2$ so by recursivity we get that $\|\mathbf{X}^{(l)}\|_F \leq \|\mathbf{X}^{(0)}\|_F \|\boldsymbol{\Theta}^{(1)}\|_2 \ldots \|\boldsymbol{\Theta}^{(l)}\|_2$. Recall that $\|\mathbf{X}^{(0)}\|_F = \sqrt{d}$. Using this observation and the inductive assumption we get that

$$\begin{aligned} &\left(\|\mathbf{E}\|_2 \left\| \mathbf{X}^{(l-1)} \right\|_F + \left\| \mathbf{X}^{(l-1)} - \mathbf{X}_p^{(l-1)} \right\|_F \right) \left\| \boldsymbol{\Theta}^{(l)} \right\|_2 \\ &\leq \sqrt{d} \|\mathbf{E}\|_2 \prod_{l=1}^{L} \left\| \boldsymbol{\Theta}^{(l)} \right\|_2 + \sqrt{d}(L-1) \|\mathbf{E}\|_2 \prod_{l=1}^{L} \left\| \boldsymbol{\Theta}^{(l)} \right\|_2 \\ &= \sqrt{d} L \|\mathbf{E}\|_2 \prod_{l=1}^{L} \left\| \boldsymbol{\Theta}^{(l)} \right\|_2. \end{aligned}$$

□

We finish this section by combining Proposition 3 with the results from Sec. 5 to give the following.

Corollary 1. *Consider the GCN outputs* $\mathbf{X}^{(L)}$ *and* $\mathbf{X}_p^{(L)}$ *for a graph* $\mathcal{G}$ *and a perturbed graph* $\mathcal{G}_p$ *where the perturbed graph is a result of double edge rewiring. Let* $\tilde{\boldsymbol{\Delta}}$ *and* $\tilde{\boldsymbol{\Delta}}_p$ *be the corresponding normalised augmented adjacency matrices. Define* d_u*,* δ_u *and* R_u *as in Sec. 5, then the following holds*

$$\left\| \mathbf{X}^{(L)} - \mathbf{X}_p^{(L)} \right\|_F \leq \sqrt{d} L \prod_{l=1}^{L} \left\| \boldsymbol{\Theta}^{(l)} \right\|_2 \max_{u \in \mathcal{V}} \frac{2R_u}{\sqrt{(d_u + 1)(\delta_u + 1)}}.$$

A similar result holds for SGCN by combining Proposition 2 with the results from Sec. 5. We suspect these bounds will likely be loose in practice; nevertheless, they provide conceptional insights into the factors that may be related to the robustness of the GCN and SGCN models. In particular, we can reason when this bound will be small as in the final paragraph of Sec. 5, which relates interpretable structural perturbation to the robustness of these models.

7. DISCUSSION

In this work, we bound the change in output of spectral graph filters under a specific form of topological perturbation, i.e., edge rewiring, where the bound involves terms which have a structural interpretation. We then demonstrate a practical application of this bound by applying it to the SGCN and GCN models. Future directions include proving that the change in output of other models scales proportional to $\|\mathbf{E}\|_2$ and providing further interpretable bounds to $\|\mathbf{E}\|_2$ to provide an understanding of stability for graph-based models. Furthermore, the SGCN and GCN models we consider both apply low-pass filters, and exploring models with other filtering characteristics is an interesting future direction. Finally, extensions of the framework presented in this work such as considering a more general perturbation model or extending it to weighted graphs will benefit its utility for practical applications.

8. REFERENCES

[1] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: Going beyond euclidean data," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, 2017.

[2] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–21, 2020.

[3] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, C. Gulcehre, F. Song, A. Ballard, J. Gilmer, G. Dahl, A. Vaswani, K. Allen, C. Nash, V. Langston, C. Dyer, N. Heess, D. Wierstra, P. Kohli, M. Botvinick, O. Vinyals, Y. Li, and R. Pascanu, "Relational inductive biases, deep learning, and graph networks," *arXiv:1806.01261*, 2018.

[4] D. I Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, May 2013.

[5] X. Dong, D. Thanou, L. Toni, M. Bronstein, and P. Frossard, "Graph signal processing for machine learning: A review and new perspectives," *arXiv:2007.16061*, 2020.

[6] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Advances in Neural Information Processing Systems 29*, 2016, pp. 3844–3852.

[7] R. Levie, F. Monti, X. Bresson, and M. M. Bronstein, "Cayleynets: Graph convolutional neural networks with complex rational spectral filters," *IEEE Transactions on Signal Processing*, vol. 67, no. 1, pp. 97–109, 2019.

[8] F. M. Bianchi, D. Grattarola, C. Alippi, and L. Livi, "Graph neural networks with convolutional arma filters," *arXiv:1901.01343*, 2020.

[9] 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*, July 2018, KDD '18, pp. 2847–2856.

[10] H. Dai, H. Li, T. Tian, X. Huang, L. Wang, J. Zhu, and L. Song, "Adversarial attack on graph structured data," in *International Conference on Machine Learning*. PMLR, 2018, pp. 1115–1124.

[11] W. Jin, X. Li, H. Xu, Y. Wang, and J. Tang, "Adversarial attacks and defenses on graphs: A review and empirical study," *arXiv:2003.00653*, 2020.

[12] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv:1312.6199*, 2013.

[13] H. Kenlay, D. Thanou, and X. Dong, "On the stability of polynomial spectral graph filters," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[14] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, and K. Weinberger, "Simplifying graph convolutional networks," in *Proceedings of the 36th International Conference on Machine Learning*, June 2019, vol. 97 of *Proceedings of Machine Learning Research*, pp. 6861–6871.

[15] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations (ICLR)*, 2017.

[16] R. Levie, E. Isufi, and G. Kutyniok, "On the transferability of spectral graph filters," *arXiv:1901.10524*, 2019.

[17] F. Gama, J. Bruna, and A. Ribeiro, "Stability properties of graph neural networks," *IEEE Transactions on Signal Processing*, vol. 68, pp. 5680–5695, 2020.

[18] A. Bojchevski and S. Günnemann, "Certifiable robustness to graph perturbations," in *Advances in Neural Information Processing Systems*, 2019, vol. 32.

[19] D. Zügner and S. Günnemann, "Certifiable robustness and robust training for graph convolutional networks," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 246–256.

[20] E. Isufi, A. Loukas, A. Simonetto, and G. Leus, "Filtering random graph processes over random time-varying graphs," *IEEE Transactions on Signal Processing*, vol. 65, no. 16, pp. 4406–4421, 2017.

[21] E. Ceci and S. Barbarossa, "Robust graph signal processing in the presence of uncertainties on graph topology," in *2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2018, pp. 1–5.

[22] Z. Gao, E. Isufi, and A. Ribeiro, "Stochastic graph neural networks," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 9080–9084.

[23] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, Society for Industrial and Applied Mathematics, 2002.

[24] Yao Ma, Suhang Wang, Tyler Derr, Lingfei Wu, and Jiliang Tang, "Attacking graph convolutional networks via rewiring," *arXiv:1906.03750*, 2019.

[25] D. Bienstock and O. Günlük, "A degree sequence problem related to network design," *Networks*, vol. 24, no. 4, pp. 195–205, 1994.

[26] B. Gao and L. Pavel, "On the properties of the softmax function with application in game theory and reinforcement learning," *arXiv:1704.00805*, 2017.