Simplifying the Theory on Over-Smoothing

Andreas Roth

TU Dortmund University, 44227 Dortmund, Germany

Abstract

Graph convolutions have gained popularity due to their ability to efficiently operate on data with an irregular geometric structure. However, graph convolutions cause over-smoothing, which refers to representations becoming more similar with increased depth. However, many different definitions and intuitions currently coexist, leading to research efforts focusing on incompatible directions. This paper attempts to align these directions by showing that over-smoothing is merely a special case of power iteration. This greatly simplifies the existing theory on over-smoothing, making it more accessible. Based on the theory, we provide a novel comprehensive definition of rank collapse as a generalized form of over-smoothing and introduce the rank-one distance as a corresponding metric. Our empirical evaluation of 14 commonly used methods shows that more models than were previously known suffer from this issue.

Keywords

graph neural networks, message-passing neural networks, theoretical properties, over-smoothing

1. Introduction

When operating with message-passing neural networks on graph-structured data, over-smoothing describes a phenomenon in which node representations become more similar when the number of convolutional layers increases. Many research efforts provide theoretical insights on over-smoothing and methods to mitigate its effects [1, 2, 3, 4, 5, 6, 7]. However, due to the multitude of different theoretical insights and their complexity, different research efforts often use distinct definitions for over-smoothing, which are partly incompatible. In particular, some works study normalized representations [8, 9, 10] while others consider unnormalized representations [11, 12, 13]. Some define over-smoothing as the convergence to a constant state [5, 7, 11, 13, 12], others claim different limit distributions depending on the spectrum of the aggregation function [14, 15, 16, 17, 8, 9, 10].

To combine these strands, we show that the theory behind over-smoothing can be greatly simplified and reduced by connecting it to the classical power iteration method [18, 19, 20]. While our resulting insights are not novel, our novel proofs aim to make the theory more accessible to a broader part of the community. We first recap power iteration with its in-depth proof. We show that most graph convolutions represent a particular case for which the dominant eigenvector is a Kronecker product. Its properties lead to over-smoothing, for which we provide a novel theoretically founded definition.

2. Preliminaries

Notation Let $G = (\mathcal{V}, \mathcal{E})$ be a graph consisting of a node set $\mathcal{V} = \{v_1, \dots, v_n\}$ and an edge set $\mathcal{E}$. The matrix representations of G is given by its adjacency matrix $\mathbf{A} \in \{0,1\}^{n \times n}$, for which $\mathbf{A}_i j = 1$ only if an edge between nodes v_i and v_j exists. For a given node v_i, the set of neighboring nodes is given by $\mathcal{N}_i = \{v_j \mid (j,i) \in \mathcal{E}\}$. The node degree is given by $d_{ii} = |\mathcal{N}_i|$ and the corresponding degree matrix $\mathbf{D} \in \mathbb{N}^{n \times n}$ with the node degrees along its diagonal. The eigenvalues of a matrix $\mathbf{M}$ are denoted by $\lambda_1^{\mathbf{M}}, \dots, \lambda_n^{\mathbf{M}}$ and sorted with decreasing magnitude. The vectorize operation $\mathrm{vec}(\mathbf{M})$ is defined as stacking the columns of $\mathbf{M}$ into a single vector.

LWDA'24: Lernen, Wissen, Daten, Analysen. September 23–25, 2024, Würzburg, Germany
andreas.roth@tu-dortmund.de (A. Roth)
0000-0002-0515-7635 (A. Roth)

Graph neural networks Given a graph G and a d-dimensional node signal $\mathbf{X} \in \mathbb{R}^{n \times d}$, we want to obtain expressive node embeddings capturing information given by both the data structure and the signals. These embeddings are utilized in downstream tasks like node classification and graph classification. Most graph convolutions apply a node mixing function $\tilde{\mathbf{A}} \in \mathbb{R}^{n \times n}$ that represents the graph structure and its edge weights, and a feature transformation $\mathbf{W} \in \mathbb{R}^{d \times d}$. In matrix notation, these can be expressed as iterative transformations of the form

$$\mathbf{X}^{(k+1)} = \tilde{\mathbf{A}}\mathbf{X}^{(k)}\mathbf{W}^{(k)}\,, \tag{1}$$

Popular instantiations covered by this notation include the graph convolutional network (GCN) [21], the graph isomorphism network (GIN) [22], and the graph attention network (GAT) [23].

Over-smoothing in GNNs Over-smoothing describes a phenomenon in which the node representations become excessively similar as the number of layers k increases. As many definitions and intuitions for over-smoothing coexist and in some cases contradict each other, we want to familiarize the reader with the different studies.

Li et al. [14] has shown that the normalized adjacency matrix employed by the GCN performs a special form of Laplacian smoothing, leading to over-smoothing when many iterations are performed.However, their study did not consider the role of the feature transformation. Oono and Suzuki [16] studied the distance of $\mathbf{X}^{(k)}$ to a subspace $\mathbf{M}$ that is spanned by the dominating eigenvector $\mathbf{v}_1$ across all columns. They bound this distance using the second largest eigenvalue $\lambda_2^{\tilde{\mathbf{A}}}$ and the largest singular value $\sigma_1^{\mathbf{W}}$ of $\mathbf{W}$. Intuitively, each aggregation step $\mathbf{A}$ reduces this distance, while $\mathbf{W}$ can increase the distance arbitrarily. Thus, they consequently claim potential over-separation of node representations when $\sigma_1^{\mathbf{W}} > \frac{1}{\lambda_2^{\mathbf{A}}}$. [15] introduced the Dirichlet energy

$$E(\mathbf{X}) = \operatorname{tr}(\mathbf{X}^T\boldsymbol{\Delta}\mathbf{X}) = \frac{1}{2}\sum_{(i,j)\in\mathcal{E}}\left\|\frac{\mathbf{x}_i}{\sqrt{d_i}} - \frac{\mathbf{x}_j}{\sqrt{d_j}}\right\|_2^2 \tag{2}$$

as an efficient and interpretable metric to quantify over-smoothing. Their study considers the GCN with aggregation matrix $\tilde{\mathbf{A}} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ and the symmetrically normalized graph Laplacian $\boldsymbol{\Delta} = \mathbf{I} - \tilde{\mathbf{A}}$, as the dominating eigenvector corresponds to its nullspace. A low energy value corresponds to similar neighboring node states. Similarly to Oono and Suzuki [16], they provided the bound

$$E(\mathbf{A}\mathbf{X}\mathbf{W}) \leq \left(\lambda_2^{\mathbf{A}}\right)^2 \left(\sigma_1^{\mathbf{W}}\right)^2 E(\mathbf{X}) \tag{3}$$

for each convolution and prove an exponential convergence to zero for the GCN. As their proof again only holds in case $\sigma_1^{\mathbf{W}} \leq \frac{1}{\lambda_2^{\mathbf{A}}}$, they similarly claim potential over-separation. Zhou et al. [17] provide a lower bound on the energy to show that the Dirichlet energy can go to infinite.

Another branch of recent research [7, 5, 13, 24] defines over-smoothing as the exponential convergence of $\mathbf{X}^{(k)}$ to a state with a constant value in each column. This contradicts the aforementioned studies as the dominant eigenvector of the aggregation matrix [16, 15, 17] is not always a constant vector, e.g., for the GCN [21].

These inconsistencies were recently clarified as the necessity to consider the normalized state was shown [8, 25, 9]. When not considering the normalized state, the norm of $\mathbf{X}^{(k)}$ going to zero can be wrongly interpreted as a convergence to a constant state [8, 9]. In the other direction, even when $\mathbf{X}^{(k)}$ is dominated by the dominant eigenvector, the Dirichlet energy of the unnormalized state wrongly indicates over-separation.

As the theory and the corresponding proofs are lengthy and complex, several recent studies still claim over-smoothing as convergence to a constant state or do not consider the normalized state. This work greatly simplifies the theory behind over-smoothing to make the theoretical backgrounds more accessible.

3. Power Iteration

As the proof for over-smoothing of graph convolutions will be a special case, we first provide the detailed proof for the well-known power iteration [18, 19, 20]. Power iteration refers to the process where a vector, when repeatedly multiplied by a matrix, gets dominated by an eigenvector of the matrix that corresponds to the eigenvalue with the largest magnitude. The proof we provide mostly follows [26], but similar proofs are available in many textbooks. For any square matrix $\mathbf{M}$, its eigenvalues are denoted by $\lambda_1^{\mathbf{M}}, \ldots, \lambda_n^{\mathbf{M}}$ and are sorted descending by their magnitude, i.e., $|\lambda_i^{\mathbf{M}}| \geq |\lambda_{i+1}^{\mathbf{M}}|$.

Proposition 3.1. (Power Iteration [26]) Let $\mathbf{S} \in \mathbb{R}^{p \times p}$ be a matrix with $|\lambda_1^{\mathbf{S}}| > |\lambda_2^{\mathbf{S}}|$ and $\mathbf{v}_1^{\mathbf{S}} \in \mathbb{R}^p$ be an eigenvector corresponding to $\lambda_1^{\mathbf{S}}$. Further, let $\mathbf{x}_0 \in \mathbb{R}^q$ be a vector that has a non-zero component c_1 in direction $\mathbf{v}_1^{\mathbf{S}}$. Then,

$$\frac{\mathbf{S}^k \mathbf{x}_0}{\|\mathbf{S}^k \mathbf{x}_0\|} = \beta_k \mathbf{v}_1^{\mathbf{S}} + \mathbf{r}_k \tag{4}$$

for some $\mathbf{r}_k \in \mathbb{R}^p$ with $\lim_{k\to\infty} \|\mathbf{r}_k\| = 0$ and $\beta_k = \frac{c_1}{|c_1|} \left(\frac{\lambda_1^{\mathbf{S}}}{|\lambda_1^{\mathbf{S}}|}\right)^k \frac{1}{\|\mathbf{v}_1^{\mathbf{S}}\|} \in \mathbb{R}$.

Proof. Let $\mathbf{S} = \mathbf{V}\mathbf{J}\mathbf{V}^{-1}$ be its Jordan decomposition, where $\mathbf{J} \in \mathbb{C}^{p \times p}$ is a block diagonal matrix containing the eigenvalues on its diagonal and $\mathbf{V} \in \mathbb{C}^{p \times p}$ contains the generalized eigenvectors as columns. As the generalized eigenvectors form a basis of $\mathbf{R}^p$, $\mathbf{x}_0$ can be decomposed as $\mathbf{x}_0 = c_1 \mathbf{v}_1^{\mathbf{S}} + \cdots + c_p \mathbf{v}_p^{\mathbf{S}}$ into a linear combination. This allows the following equalities:

$$\begin{aligned}
\frac{\mathbf{S}^k \mathbf{x}_0}{\|\mathbf{S}^k \mathbf{x}_0\|} &= \frac{(\mathbf{V}\mathbf{J}\mathbf{V}^{-1})^k (c_1 \mathbf{v}_1^{\mathbf{S}} + \ldots c_n \mathbf{v}_n^{\mathbf{S}})}{\|(\mathbf{V}\mathbf{J}\mathbf{V}^{-1})^k (c_1 \mathbf{v}_1^{\mathbf{S}} + \ldots c_n \mathbf{v}_n^{\mathbf{S}})\|} \\
&= \frac{\mathbf{V}\mathbf{J}^k (c_1 \mathbf{e}_1 + \ldots c_n \mathbf{e}_n)}{\|\mathbf{V}\mathbf{J}^k (c_1 \mathbf{e}_1 + \ldots c_n \mathbf{e}_n)\|} \\
&= \frac{c_1}{|c_1|} \left(\frac{\lambda_1^{\mathbf{S}}}{|\lambda_1^{\mathbf{S}}|}\right)^k \frac{\mathbf{V}(\frac{1}{\lambda_1^{\mathbf{S}}}\mathbf{J})^k \frac{1}{c_1}(c_1 \mathbf{e}_1 + \ldots c_n \mathbf{e}_n)}{\|\mathbf{V}(\frac{1}{\lambda_1^{\mathbf{S}}}\mathbf{J})^k \frac{1}{c_1}(c_1 \mathbf{e}_1 + \ldots c_n \mathbf{e}_n)\|}
\end{aligned} \tag{5}$$

The second equation uses the fact $\mathbf{V}^{-1}\mathbf{v}_k^{\mathbf{S}} = \mathbf{e}_k$, i.e., the natural basis vector pointing in direction k. As $\mathbf{J}$ is normalized by its unique largest entry $\lambda_1^{\mathbf{S}}$, it converges to

$$\lim_{k\to\infty} \left(\frac{1}{\lambda_1^{\mathbf{S}}}\mathbf{J}\right)^k = \begin{bmatrix} 1 & & & \\ & 0 & & \\ & & \ddots & \\ & & & 0 \end{bmatrix}. \tag{6}$$

Equation 5 then simplifies to

$$\frac{c_1}{|c_1|} \left(\frac{\lambda_1}{|\lambda_1|}\right)^k \frac{\mathbf{V}(\frac{1}{\lambda_1}\mathbf{J})^k \frac{1}{c_1}(c_1 \mathbf{e}_1 + \ldots c_n \mathbf{e}_n)}{\|\mathbf{V}(\frac{1}{\lambda_1}\mathbf{J})^k \frac{1}{c_1}(c_1 \mathbf{e}_1 + \ldots c_n \mathbf{e}_n)\|} = \frac{c_1}{|c_1|} \left(\frac{\lambda_1^{\mathbf{S}}}{|\lambda_1^{\mathbf{S}}|}\right)^k \frac{\mathbf{v}_1^{\mathbf{S}}}{\|\mathbf{v}_1^{\mathbf{S}}\|} + \mathbf{r}_k \tag{7}$$

with $\lim_{k\to\infty} \|\mathbf{r}_k\| = 0$. It converges to $\frac{\mathbf{v}_1^{\mathbf{S}}}{\|\mathbf{v}_1^{\mathbf{S}}\|}$ iff $\lambda_1^{\mathbf{S}} > 0$. □

Note that $|\lambda_1^{\mathbf{S}}| > |\lambda_2^{\mathbf{S}}|$ holds for almost every matrix $\mathbf{S}$ with respect to the Lebesgue measure [27].

4. Graph Convolutions as Power Iteration

We now show that this proof can be applied to all graph convolutions of the form given by Eq. 1. We express these graph convolutions in vector notation [28]

$$\mathrm{vec}(\mathbf{A}\mathbf{X}\mathbf{W}) = (\mathbf{W}^T \otimes \mathbf{A})\mathrm{vec}(\mathbf{X}) = \mathbf{S}\mathbf{x}_0 \tag{8}$$

using the Kronecker product $\otimes$ that is defined as $\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & \dots & a_{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & \dots & a_{mn}\mathbf{B} \end{bmatrix}$. This formulation is commonly used to study over-smoothing [8, 10, 9] and other properties of graph convolutions [29, 6, 30]. The Kronecker product has a key spectral property affecting power iteration: All eigenvectors $\mathbf{v}_{ij}^{\mathbf{S}} = \mathbf{v}_i^{(\mathbf{W}^\mathbf{T})} \otimes \mathbf{v}_j^{\mathbf{A}}$ of $\mathbf{W}^T \otimes \mathbf{A}$ are Kronecker products of the eigenvectors of $\mathbf{A}$ and $\mathbf{W}^T$ with corresponding eigenvalue $\lambda_i^{\mathbf{A}}\lambda_j^{\mathbf{W}}$ [28]. This lets us state the reason behind over-smoothing in a clearer way than in previous works by substituting $\mathbf{v}_1^{\mathbf{S}}$:

Proposition 4.1. (Power Iteration with a Kronecker Product) Let $\mathbf{S} = \mathbf{W} \otimes \mathbf{A} \in \mathbb{R}^{(n\cdot d)\times(n\cdot d)}$ for any $\mathbf{W} \in \mathbb{R}^{d\times d}$ and $\mathbf{A} \in \mathbb{R}^{d\times d}$ with $|\lambda_1^{\mathbf{S}}| > |\lambda_2^{\mathbf{S}}|$. Let $\mathbf{v}_1^{\mathbf{A}}, \mathbf{v}_1^{\mathbf{W}}$ be two eigenvectors corresponding to $\lambda_1^{\mathbf{A}}$ and $\lambda_1^{\mathbf{W}}$, respectively. Further, let $\mathbf{x}_0 \in \mathbb{R}^{n\cdot d}$ be a vector that has a non-zero component c_1 in direction $\mathbf{v}_1^{\mathbf{S}} = \mathbf{v}_1^{\mathbf{W}} \otimes \mathbf{v}_1^{\mathbf{A}}$. Then,

$$\frac{(\mathbf{W}\otimes\mathbf{A})^k\mathbf{x}_0}{\|(\mathbf{W}\otimes\mathbf{A})^k\mathbf{x}_0\|} = \beta_k \cdot \mathbf{v}_1^{\mathbf{W}} \otimes \mathbf{v}_1^{\mathbf{A}} + \mathbf{r}_k \tag{9}$$

for some $\mathbf{r}_k \in \mathbb{R}^{n\cdot d}$ with $\lim_{k\to\infty} \|\mathbf{r}_k\| = 0$ and $\beta_k = \frac{c_1}{|c_1|}\frac{\left(\frac{\lambda_1^{\mathbf{A}}\lambda_1^{\mathbf{W}}}{|\lambda_1^{\mathbf{A}}\lambda_1^{\mathbf{W}}|}\right)^k}{\|\mathbf{v}_1^{\mathbf{W}}\otimes\mathbf{v}_1^{\mathbf{A}}\|} \in \mathbb{R}$.

Proof. Given that $|\lambda_1^{\mathbf{S}}| > |\lambda_2^{\mathbf{S}}|$, and $\lambda_{i\cdot j}^{\mathbf{S}} = \lambda_i^{\mathbf{A}} \cdot \lambda_j^{\mathbf{W}}$ for all $0 < i < n$ and $0 < j < d$, we have $|\lambda_1^{\mathbf{A}}| > |\lambda_2^{\mathbf{A}}|$ and $|\lambda_1^{\mathbf{W}}| > |\lambda_2^{\mathbf{W}}|$. The corresponding eigenvector $\mathbf{v}_1^{\mathbf{S}} = \mathbf{v}_1^{\mathbf{A}} \otimes \mathbf{v}_1^{\mathbf{W}}$ is the Kronecker product of the corresponding eigenvectors of $\mathbf{A}$ and $\mathbf{W}$. Substituting these in Proposition 3.1 results in our Proposition 4.1. □

Extending Proposition 4.1 for any $\mathbf{W}$ and possibly repeated $\lambda_1^{\mathbf{W}}$ is similar, as all generalized eigenvectors of $\mathbf{W} \otimes \mathbf{A}$ corresponding to $\lambda_1^{\mathbf{S}}$ are of the form $\mathbf{u} \otimes \mathbf{v}_1^{\mathbf{A}}$ for different $\mathbf{u}$. To simplify this work, we provide this proof as Proposition A.1 in Appendix A. The implications of Proposition 4.1 become clearer when looking into its matrix form:

Remark 4.2. (Power Iteration with a Kronecker Product in Matrix Notation) Stating Proposition 4.1 in matrix notation leads to

$$\frac{\mathbf{A}^k\mathbf{X}\mathbf{W}^k}{\|\mathbf{A}^k\mathbf{X}\mathbf{W}^k\|} = \beta_k \mathbf{v}_1^{\mathbf{A}} \left(\mathbf{v}_1^{\mathbf{W}}\right)^T + \mathbf{R}_k \tag{10}$$

for $\mathrm{vec}(\mathbf{X}) = \mathbf{x}_0$ and some $\mathbf{R}_k$ with $\lim_{k\to\infty} \|\mathbf{R}_k\| = 0$ and $\beta_k = \frac{c_1}{|c_1|}\frac{\left(\frac{\lambda_1^{\mathbf{A}}\lambda_1^{\mathbf{W}}}{|\lambda_1^{\mathbf{A}}\lambda_1^{\mathbf{W}}|}\right)^k}{\|\mathbf{v}_1^{\mathbf{W}}\otimes\mathbf{v}_1^{\mathbf{A}}\|} \in \mathbb{R}$.

Any graph convolution of this form amplifies the same signal across all feature columns, and the state gets closer to a rank one matrix, with each column becoming a multiple of $\mathbf{v}_1^{\mathbf{A}}$. This phenomenon was termed rank collapse, but a definition is still missing [9]. A comprehensive definition must consider the normalized representations and be independent of one specific vector. We introduce the following definition:

Definition 4.1. (Rank Collapse) A sequence of matrices $\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(k)} \in \mathbb{R}^{n\times d}$ is said to suffer from **rank collapse** if there exists a sequence of rank-one matrices $\mathbf{Y}^{(1)}, \dots, \mathbf{Y}^{(k)} \in \mathbb{R}^{n\times d}$ such that

$$\lim_{k\to\infty} \left\| \frac{\mathbf{X}^{(k)}}{\|\mathbf{X}^{(k)}\|} - \mathbf{Y}^{(k)} \right\| = 0 \tag{11}$$

It is commonly referred to as over-smoothing as they only consider stochastic aggregation functions or symmetrically normalized adjacency matrices. Their eigenvector $\mathbf{v}_1^{\mathbf{A}}$ is a smooth vector for typical choices of $\mathbf{A}$, e.g., it is the vector of all ones $\mathbf{v}_1^{\mathbf{A}} = \mathbb{1}$ for the (weighted) mean aggregation, and $\mathbf{v}_1^{\mathbf{A}} = \mathbf{D}^{\frac{1}{2}}\mathbb{1}$ for the symmetrically normalized adjacency matrix [31]. We thus define over-smoothing as a special case of rank collapse:

Definition 4.2. (Over-Smoothing) A sequence of matrices $\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(k)} \in \mathbb{R}^{n \times d}$ is said to suffer from **over-smoothing** if it suffers from rank collapse and the rank-one matrices are of the form $\mathbf{Y}^{(l)} = \mathbf{1}\mathbf{c}_{(l)}^T$ or $\mathbf{Y} = \mathbf{D}^{\frac{1}{2}}\mathbf{1}\mathbf{c}_{(l)}^T$ for any $\mathbf{c}_{(l)} \in \mathbb{R}^d$.

To quantify the degree of over-smoothing, a frequently used metric is the Dirichlet energy [15]

$$E\left(\frac{\mathbf{X}}{\|\mathbf{X}\|}\right) = \operatorname{tr}\left(\frac{\mathbf{X}}{\|\mathbf{X}\|}\boldsymbol{\Delta}\frac{\mathbf{X}}{\|\mathbf{X}\|}\right), \tag{12}$$

with $\boldsymbol{\Delta} = \mathbf{D} - \mathbf{A}$ or $\boldsymbol{\Delta} = \mathbf{I}_n - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$, depending on case of over-smoothing. The Dirichlet energy converges to zero for methods utilizing the corresponding aggregation function as $\mathbf{v}_1$ is in the nullspace of $\boldsymbol{\Delta}$, i.e., $\boldsymbol{\Delta}\mathbf{v}_1 = \mathbf{0}$. However, this requires different $\boldsymbol{\Delta}$ for different aggregation functions, as $\mathbf{v}_1^{\mathbf{A}}$ may be different.

In order to quantify the more general phenomenon of rank collapse, we need to find a suitable sequence of rank-one matrices. The singular vectors corresponding to the largest singular value give the closest rank-one approximation of a given matrix. However, finding singular vectors is computationally expensive. As we are mainly interested in whether $\mathbf{X}$ converges to zero and not the exact value, we will instead utilize a row and column vector of the given $\mathbf{X}$. Any row and column will be sufficient if $\mathbf{X}$ is close to a rank one matrix. We consider the row and column vectors with the largest norm for numerical stability. This leads to our newly proposed rank-one distance metric:

Definition 4.3. (Rank-one Distance (ROD)) Let $\mathbf{X} \in \mathbb{R}^{p \times q}$ be any matrix. We define the row with the largest norm as $\mathbf{v} = \max_i \|\mathbf{x}_i\| \in \mathbb{R}^p$ and the column index corresponding the largest norm as $j = \arg\max_i \|\mathbf{x}_{:,i}\| \in \mathbb{R}^q$. To account for the correct signs, we utilize the j-th column vector $\mathbf{u} = \mathbf{x}_{:,j}$ if $v_j > 0$ or its negated version $\mathbf{u} = -\mathbf{x}_{:,j}$ otherwise. The rank-one distance of $\mathbf{X}$ is then defined as

$$\operatorname{ROD}(\mathbf{X}) = \left\| \frac{\mathbf{X}}{\|\mathbf{X}\|} - \frac{\mathbf{u}\mathbf{v}^T}{\|\mathbf{u}\mathbf{v}^T\|} \right\|. \tag{13}$$

As a generalization of over-smoothing, a Dirichlet energy of zero implies a ROD of zero, i.e., $E(\mathbf{X}/\|\mathbf{X}\|) = 0 \Rightarrow \operatorname{ROD}(\mathbf{X}) = 0$. Similarly to Roth and Liebig [9], our theory also explains how to prevent over-smoothing and rank collapse. It needs to be ensured that the graph convolution is not a Kronecker product, i.e., a single aggregation and transformation matrix. One direction is to operate on multiple computational graphs $\mathbf{A}_1, \ldots, \mathbf{A}_l$ with distinct feature transformations $\mathbf{W}_1, \ldots, \mathbf{W}_l$:

$$\begin{aligned} \mathbf{S}\operatorname{vec}(\mathbf{X}) &= (\mathbf{W}_1 \otimes \mathbf{A}_1 + \cdots + \mathbf{W}_l \otimes \mathbf{A}_l)\operatorname{vec}(\mathbf{X}) \\ &= \operatorname{vec}(\mathbf{A}_1\mathbf{X}\mathbf{W}_1^T + \cdots + \mathbf{A}_l\mathbf{X}\mathbf{W}_l^T). \end{aligned} \tag{14}$$

As the eigenvectors of sums of Kronecker products can be linearly independent across the corresponding columns, different signals can get amplified for each feature column.

5. Experimental Validation

Given our novel definition and corresponding metric, we evaluate various well-established graph convolutions in terms of their ability to avoid rank collapse. Our implementation based on PyTorchGeometric [32] is available online [1].

Methods As base message-passing methods, we evaluate the dynamics of the graph convolutional network [21] and the graph attention network (GAT) [23]. These are known to suffer from over-smoothing, so they also suffer rank collapse. The other methods we consider are not generally known to cause over-smoothing or are specifically designed to prevent it. While previous theory shows

[1] https://github.com/roth-andreas/simplifying-over-smoothing

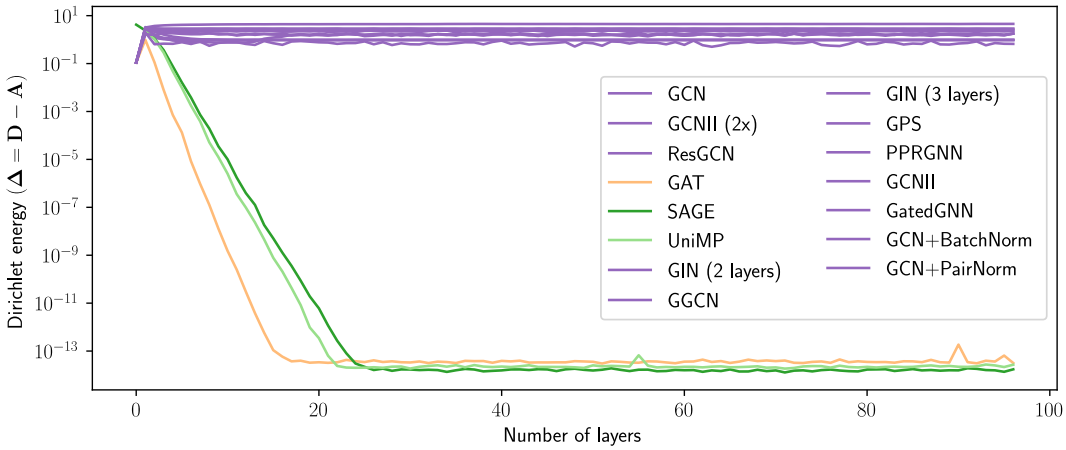

Figure 1: Change in Dirichlet energy using the unnormalized graph Laplacian when increasing the number of layers for randomly initialized models. Mean values over 50 random seeds.

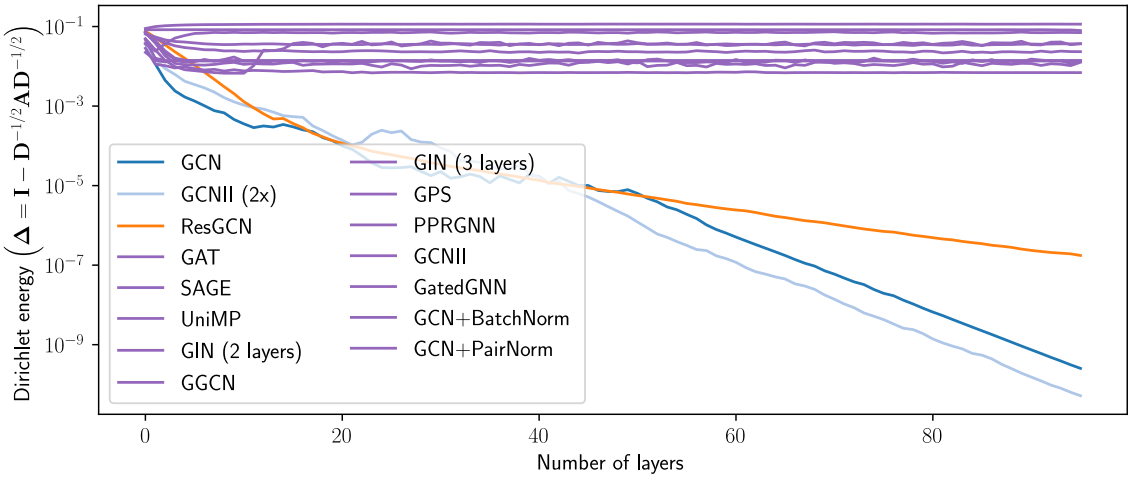

Figure 2: Change in Dirichlet energy using the symmetrically normalized graph Laplacian when increasing the number of layers for randomly initialized models. Mean values over 50 random seeds.

that negative edge weights and similarly residual connections can avoid over-smoothing [33, 34, 8], the connection to power iteration implies that these cannot prevent rank collapse. We evaluate the combination of the GCN and a residual connection (ResGCN), which adds the previous state to the output of each convolution. Similarly, we evaluate the SAGE convolution [35], which additionally applies a linear transformation to the previous state. We also consider a commonly employed method that allows negative attention weights, namely the generalized GCN (GGCN) [34]. Another direction that aims to prevent over-smoothing is based on combining the output of each iteration with the initial state, referred to as an initial residual connection [36] or a restart term [6]. We evaluate the GCNII [36]. While previous work argued that this prevents over-smoothing for the unnormalized state. Given the critical importance of considering the normalized state, we want to evaluate whether this property still holds. As the magnitude of the representations may grow in each iteration, the constant influence of the initial state may become negligible. Towards this end, we consider two versions: one with the regular parameter initialization (GCNII) and one for which all parameters are scaled by a factor of two (GCNII 2x). As an alternative approach, we also consider the personalized page rank graph neural network (PPRGNN) [6], a formulation that provably converges to a steady state with increased depth. Gating mechanisms allow a node to retain its representation by limiting the mixing with neighboring states using learnable gating functions. Here, we evaluate the gated graph neural network (GatedGNN) [37]. Normalization layers were also shown to prevent over-smoothing. Here, we evaluate PairNorm [2]

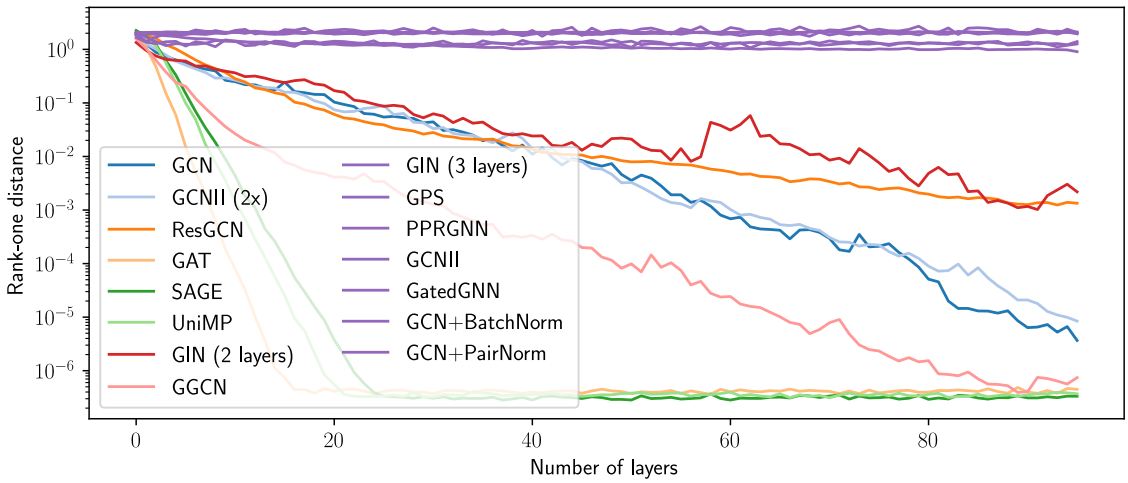

Figure 3: Change of the rank-one distance when increasing the number of layers for randomly initialized models. Mean values over 50 random seeds.

(GCN+PairNorm) and BatchNorm [38] (GCN+BatchNorm), combined with the GCN by their ability to mitigate rank collapse. The graph isomorphism network [22] applies a multi-layer non-linear feature transformation to each state. While this was designed to allow for injective mappings of multisets to achieve maximal expressivity, this should allow resulting representations to be linearly independent. We evaluate a version using a multi-layer perceptron (MLP) with two layers (GIN 2 layers) and a version using an MLP with three layers (GIN 3 layers). As global methods, we consider the unified message passaging model (UniMP) [39] and the general, powerful, scalable (GPS) graph transformer [40] as two graph transformer variations.

Setting We track the rank-one distance for all methods on the KarateClub dataset [41], which is a small and undirected graph consisting of 34 nodes and 156 edges. We initialize each node with randomly assigned features following a normal distribution. For each method and for each iteration, the corresponding graph convolution followed by a ReLU-activation is applied. The rank-one distance is computed after each iteration. For direct comparison, we also compute the Dirichlet energy based on the unnormalized graph Laplacian $\mathbf{\Delta} = \mathbf{D} - \mathbf{A}$ and the symmetrically normalized graph Laplacian $\mathbf{\Delta} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$. We repeat this process for 96 iterations as several methods become too unstable at further depth, i.e., feature magnitudes explode or diminish. Runs for each method are repeated for 50 random seeds. Mean values are reported.

Results The changes in the Dirichlet energy using the unnormalized graph Laplacian are shown in Figure 1. This Dirichlet energy converges to zero only for GAT, SAGE, and UniMP, which all use weighted mean aggregation. In Figure 2, we visualize the changes in Dirichlet energy using the symmetrically normalized graph Laplacian. For GCN, GCNII (2x), and ResGCN, this metric converges to zero as these methods utilize the symmetrically normalized adjacency matrix for aggregation. In Figure 3, we present the rank-one distance (ROD) for all methods. The results confirm that ROD converges to zero for all methods for which the Dirichlet energy with either graph Laplacian converges to zero. In addition, we find GIN (2 layers) and GGCN to suffer from rank collapse. While these methods use different aggregation functions, ROD captures the underlying issue of rank collapse. Different rates of convergence are notable, particularly between methods using the mean aggregation (GAT, SAGE, UniMP) and methods using other aggregation functions (GCN, GGCN, GCNII, ResGCN, and GIN). However, all these methods cause the rank to converge to one across all 50 random initializations. Methods for which we do not observe rank collapse use normalization techniques (GCN+PairNorm and GCN+BatchNorm), prevent unlimited depth (PPRGNN, GCNII, and GatedGNN), and use non-linear feature transformations (GPS and GIN (3-layer)).

6. Conclusion

We have shown that over-smoothing is a special case of power iteration, with the dominant eigenvector of graph convolutions $\mathbf{W} \otimes \mathbf{A}$ taking the form $\mathbf{v}_1^{\mathbf{W}} \otimes \mathbf{v}_1^{\mathbf{A}}$. As given in power iteration, normalization is required, and the limit distribution is not always the constant vector, as it depends on the dominant eigenvector of $\mathbf{A}$. Based on our novel definition of rank collapse and the corresponding rank-one distance, we identified several methods suffering from rank collapse that were previously considered to prevent over-smoothing. We empirically found three general directions to prevent rank collapse: Normalization layers, limiting the effective depth of a model using a restart term or a gating mechanism, and complex non-linear feature transformations. These are promising avenues for further consideration. In addition, to solve rank collapse in the message-passing steps itself, the theory indicates that it needs to be ensured that the dominant eigenvector $\mathbf{v}_1^{\mathbf{S}}$ is not a simple Kronecker product so that it can amplify different signals across feature columns.

Acknowledgments

This research has been funded by the Federal Ministry of Education and Research of Germany under grant no. 01IS22094E WEST-AI.

References

[1] K. Zhou, X. Huang, Y. Li, D. Zha, R. Chen, X. Hu, Towards deeper graph neural networks with differentiable group normalization, Advances in neural information processing systems 33 (2020) 4917–4928.

[2] L. Zhao, L. Akoglu, Pairnorm: Tackling oversmoothing in gnns, in: International Conference on Learning Representations, 2020.

[3] Y. Rong, W. Huang, T. Xu, J. Huang, Dropedge: Towards deep graph convolutional networks on node classification, in: International Conference on Learning Representations, 2020.

[4] D. Chen, Y. Lin, W. Li, P. Li, J. Zhou, X. Sun, Measuring and relieving the over-smoothing problem for graph neural networks from the topological view, in: Proceedings of the AAAI conference on artificial intelligence, volume 34, 2020, pp. 3438–3445.

[5] T. K. Rusch, B. Chamberlain, J. Rowbottom, S. Mishra, M. Bronstein, Graph-coupled oscillator networks, in: International Conference on Machine Learning, PMLR, 2022, pp. 18888–18909.

[6] A. Roth, T. Liebig, Transforming pagerank into an infinite-depth graph neural network, in: Joint European conference on machine learning and knowledge discovery in databases, Springer, 2022, pp. 469–484.

[7] T. K. Rusch, B. P. Chamberlain, M. W. Mahoney, M. M. Bronstein, S. Mishra, Gradient gating for deep multi-rate learning on graphs, in: The Eleventh International Conference on Learning Representations, 2023.

[8] F. D. Giovanni, J. Rowbottom, B. P. Chamberlain, T. Markovich, M. M. Bronstein, Understanding convolution on graphs via energies, Transactions on Machine Learning Research (2023).

[9] A. Roth, T. Liebig, Rank collapse causes over-smoothing and over-correlation in graph neural networks, in: The Second Learning on Graphs Conference, 2023.

[10] S. Maskey, R. Paolino, A. Bacho, G. Kutyniok, A fractional graph laplacian approach to oversmoothing, in: A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, S. Levine (Eds.), Advances in Neural Information Processing Systems, volume 36, Curran Associates, Inc., 2023, pp. 13022–13063.

[11] X. Wu, A. Ajorlou, Z. Wu, A. Jadbabaie, Demystifying oversmoothing in attention-based graph neural networks, in: A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, S. Levine (Eds.), Advances in Neural Information Processing Systems, volume 36, Curran Associates, Inc., 2023, pp. 35084–35106.

[12] M. Scholkemper, X. Wu, A. Jadbabaie, M. Schaub, Residual connections and normalization can provably prevent oversmoothing in gnns, arXiv preprint arXiv:2406.02997 (2024).
[13] T. K. Rusch, M. M. Bronstein, S. Mishra, A survey on oversmoothing in graph neural networks, arXiv preprint arXiv:2303.10993 (2023).
[14] Q. Li, Z. Han, X.-M. Wu, Deeper insights into graph convolutional networks for semi-supervised learning, in: Proceedings of the AAAI conference on artificial intelligence, volume 32, 2018.
[15] C. Cai, Y. Wang, A note on over-smoothing for graph neural networks, arXiv preprint arXiv:2006.13318 (2020).
[16] K. Oono, T. Suzuki, Graph neural networks exponentially lose expressive power for node classification, in: International Conference on Learning Representations, 2020.
[17] K. Zhou, X. Huang, D. Zha, R. Chen, L. Li, S.-H. Choi, X. Hu, Dirichlet energy constrained learning for deep graph neural networks, Advances in Neural Information Processing Systems 34 (2021) 21834–21846.
[18] G. Kowalewski, Einführung in die Determinantentheorie einschließlich der unendlichen und der Fredholmschen Determinanten, Veit & comp., 1909.
[19] L. Miintz, Solution direct de 1'equation seculaire et de quelques problemes analogues transcendents, Comptes Rendus de I'Academie des Sciences, Paris 156 (1913) 43–6.
[20] R. Mises, H. Pollaczek-Geiringer, Praktische verfahren der gleichungsauflösung., ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik 9 (1929) 58–77.
[21] T. N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: International Conference on Learning Representations, 2017.
[22] K. Xu, W. Hu, J. Leskovec, S. Jegelka, How powerful are graph neural networks?, in: International Conference on Learning Representations, 2019. URL: https://openreview.net/forum?id=ryGs6iA5Km.
[23] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, Graph attention networks, in: International Conference on Learning Representations, 2018. URL: https://openreview.net/forum?id=rJXMpikCZ.
[24] X. Wu, A. Ajorlou, Z. Wu, A. Jadbabaie, Demystifying oversmoothing in attention-based graph neural networks, Advances in Neural Information Processing Systems 36 (2023).
[25] S. Maskey, R. Paolino, A. Bacho, G. Kutyniok, A fractional graph laplacian approach to oversmoothing, Advances in Neural Information Processing Systems 36 (2023).
[26] P. Knabner, W. Barth, Lineare Algebra, Springer, 2017.
[27] T. Tao, Topics in random matrix theory, volume 132, American Mathematical Soc., 2012.
[28] K. Schacke, On the kronecker product, Master's thesis, University of Waterloo (2004).
[29] F. Gu, H. Chang, W. Zhu, S. Sojoudi, L. El Ghaoui, Implicit graph neural networks, Advances in Neural Information Processing Systems 33 (2020) 11984–11995.
[30] F. Di Giovanni, T. K. Rusch, M. M. Bronstein, A. Deac, M. Lackenby, S. Mishra, P. Veličković, How does over-squashing affect the power of gnns?, arXiv preprint arXiv:2306.03589 (2023).
[31] U. Von Luxburg, A tutorial on spectral clustering, Statistics and computing 17 (2007) 395–416.
[32] M. Fey, J. E. Lenssen, Fast graph representation learning with pytorch geometric, arXiv preprint arXiv:1903.02428 (2019).
[33] D. Bo, X. Wang, C. Shi, H. Shen, Beyond low-frequency information in graph convolutional networks, in: Proceedings of the AAAI conference on artificial intelligence, volume 35, 2021, pp. 3950–3957.
[34] Y. Yan, M. Hashemi, K. Swersky, Y. Yang, D. Koutra, Two sides of the same coin: Heterophily and oversmoothing in graph convolutional neural networks, in: 2022 IEEE International Conference on Data Mining (ICDM), IEEE, 2022, pp. 1287–1292.
[35] W. Hamilton, Z. Ying, J. Leskovec, Inductive representation learning on large graphs, Advances in neural information processing systems 30 (2017).
[36] M. Chen, Z. Wei, Z. Huang, B. Ding, Y. Li, Simple and deep graph convolutional networks, in: International conference on machine learning, PMLR, 2020, pp. 1725–1735.

[37] Y. Li, D. Tarlow, M. Brockschmidt, R. S. Zemel, Gated graph sequence neural networks, in: Y. Bengio, Y. LeCun (Eds.), 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings, 2016.

[38] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: International conference on machine learning, pmlr, 2015, pp. 448–456.

[39] Y. Shi, H. Zhengjie, S. Feng, H. Zhong, W. Wang, Y. Sun, Masked label prediction: Unified message passing model for semi-supervised classification, 2021, pp. 1548–1554. doi:10.24963/ijcai.2021/214.

[40] L. Rampášek, M. Galkin, V. P. Dwivedi, A. T. Luu, G. Wolf, D. Beaini, Recipe for a general, powerful, scalable graph transformer, Advances in Neural Information Processing Systems 35 (2022) 14501–14515.

[41] W. W. Zachary, An information flow model for conflict and fission in small groups, Journal of Anthropological Research 33 (1977) 452–473. doi:10.1086/jar.33.4.3629752.

A. Appendix

Proposition A.1. (Power Iteration with a Kronecker Product) Let $\mathbf{S} = \mathbf{W} \otimes \mathbf{A}$ for $\mathbf{W} \in \mathbb{R}^{d \times d}$ and $\mathbf{A} \in \mathbb{R}^{n \times n}$ with $|\lambda_1^{\mathbf{A}}| > |\lambda_2^{\mathbf{A}}|$. Let $\mathbf{v}_1^{\mathbf{A}}$ be an eigenvector corresponding to $\lambda_1^{\mathbf{A}}$. Further, let $\mathbf{x}_0 \in \mathbb{R}^{n \cdot d}$ be any vector that has a non-zero component in the direction of a generalized eigenvector $\mathbf{v}_1^{\mathbf{S}}$ corresponding to $\lambda_1^{\mathbf{S}}$. Then,

$$\frac{(\mathbf{W} \otimes \mathbf{A})^k \mathbf{x}_0}{\|(\mathbf{W} \otimes \mathbf{A})^k \mathbf{x}_0\|} = \beta_k \cdot \mathbf{u} \otimes \mathbf{v}_1^{\mathbf{A}} + \mathbf{r}_k \tag{15}$$

for some $\mathbf{r}_k \in \mathbb{R}^{n \cdot d}$ with $\lim_{k \to \infty} \|\mathbf{r}_k\| = 0$, bounded β_k, and some $\mathbf{u} \in \mathbb{R}^d$.

Proof. This proof is similar to the proof of Proposition 3.1. However, $\pm\lambda_1^{\mathbf{S}}$ may occur multiple times, so there can be multiple Jordan blocks corresponding to $\pm\lambda_1^{\mathbf{S}}$, and they can have a size larger than one. Let p be the size of the largest Jordan block corresponding to $\lambda_1^{\mathbf{S}}$. Then, $\mathbf{J}^k$ will be dominated by $q_k = \binom{k}{p-1} \lambda_1^{S^{k-(p-1)}}$:

$$\lim_{k \to \infty} \left(\frac{1}{q_k}\mathbf{J}\right)^k = \begin{bmatrix} 0 & \dots & 0 & 1 & & & & & & \\ & \ddots & & 0 & & & & & & \\ \vdots & & & \vdots & & & & & & \\ 0 & \dots & & 0 & & & & & & \\ & & & & \ddots & & & & & \\ & & & & & 0 & \dots & 0 & 1 & & \\ & & & & & & \ddots & & 0 & & \\ & & & & & \vdots & & & \vdots & & \\ & & & & & 0 & \dots & & 0 & & \\ & & & & & & & & & 0 & \\ & & & & & & & & & & \ddots \\ & & & & & & & & & & & 0 \end{bmatrix}. \tag{16}$$

The number of blocks containing a 1 is determined by the number of Jordan blocks corresponding to $\pm\lambda_1^{\mathbf{S}}$ with size p. Let there be i such blocks. We further know that all corresponding generalized eigenvectors are of the form $\mathbf{v}_{i \cdot p}^{\mathbf{S}} = \mathbf{v}_{i \cdot p}^{\mathbf{W}} \otimes \mathbf{v}_1^{\mathbf{W}}$. For eigenvalues constructed with $\lambda_2^{\mathbf{A}}$ it holds that $\lambda_2^{\mathbf{A}} \lambda_{i \cdot p} < \lambda_1^{\mathbf{A}} \lambda_{i \cdot p}$. This lets us simplify the statement:

$$\begin{aligned} \left(\frac{q_k}{|q_k|}\right)^k \frac{\mathbf{V}(\frac{1}{q_k}\mathbf{J})^k (c_1 \mathbf{e}_1 + \dots c_n \mathbf{e}_n)}{\|\mathbf{V}(\frac{1}{q_k}\mathbf{J})^k (c_1 \mathbf{e}_1 + \dots c_n \mathbf{e}_n)\|} &= \left(\frac{q_k}{|q_k|}\right)^k \frac{c_{1 \cdot p} \mathbf{v}_{1 \cdot p}^{\mathbf{S}} + \dots + c_{i \cdot p} \mathbf{v}_{i \cdot p}^{\mathbf{S}}}{\|c_{1 \cdot p} \mathbf{v}_{1 \cdot p}^{\mathbf{S}} + \dots + c_{i \cdot p} \mathbf{v}_{i \cdot p}^{\mathbf{S}}\|} + \mathbf{r}_k \\ &= \left(\frac{q_k}{|q_k|}\right)^k \frac{b\mathbf{u} \otimes \mathbf{v_1}^{\mathbf{A}}}{\|b\mathbf{u} \otimes \mathbf{v_1}^{\mathbf{A}}\|} + \mathbf{r}_k \end{aligned} \tag{17}$$

for $b = c_{1 \cdot p} \cdot \dots \cdot c_{i \cdot p}$, $\mathbf{u} = \mathbf{v}_{1 \cdot p} + \dots + \mathbf{v}_{i \cdot p}$, and $\lim_{k \to \infty} \|\mathbf{r}_k\| = 0$ which converges to $\frac{\mathbf{v}_1}{\|\mathbf{v}_1\|}$ iff $\lambda_1 > 0$. Setting $\beta_k = \left(\frac{q_k}{|q_k|}\right)^k \frac{1}{\|b\mathbf{u} \otimes \mathbf{v_1}^{\mathbf{A}}\|}$ leads to Proposition 4.1. □