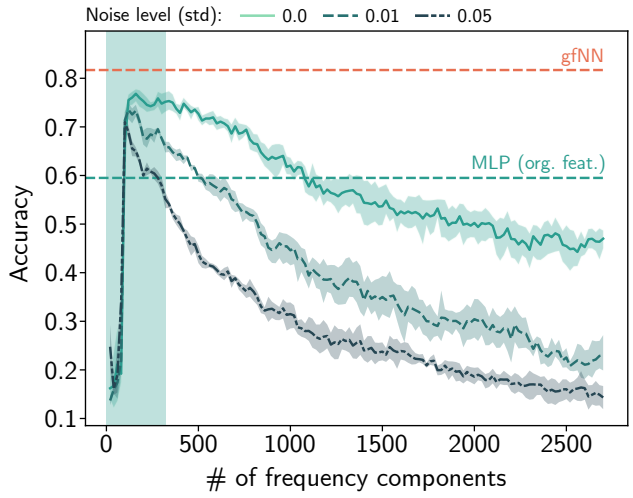
Revisiting GNN: Graph Filtering Perspective



Hoang NT, Takanori Maehara, and Tsuyoshi Murata

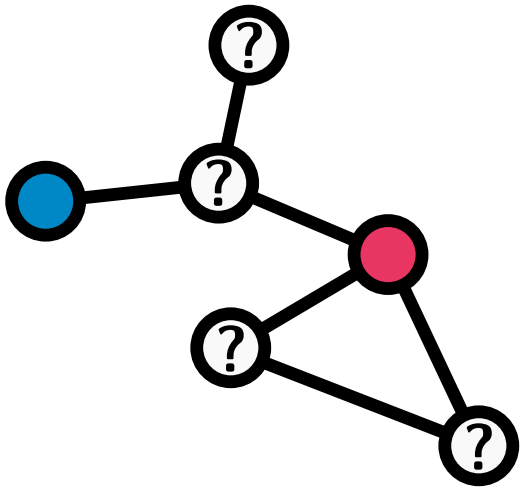
Tokyo Institute of Technology, RIKEN AIP

December 9, 2020

Outline

Vertex Classification Problem

Problem Illustration



Graph: $G = (V, E)$
Vertex features: $\mathcal{X} : V \mapsto \mathbb{R}^d$
Vertex labels: $\mathcal{Y} : V \mapsto \mathcal{C}$

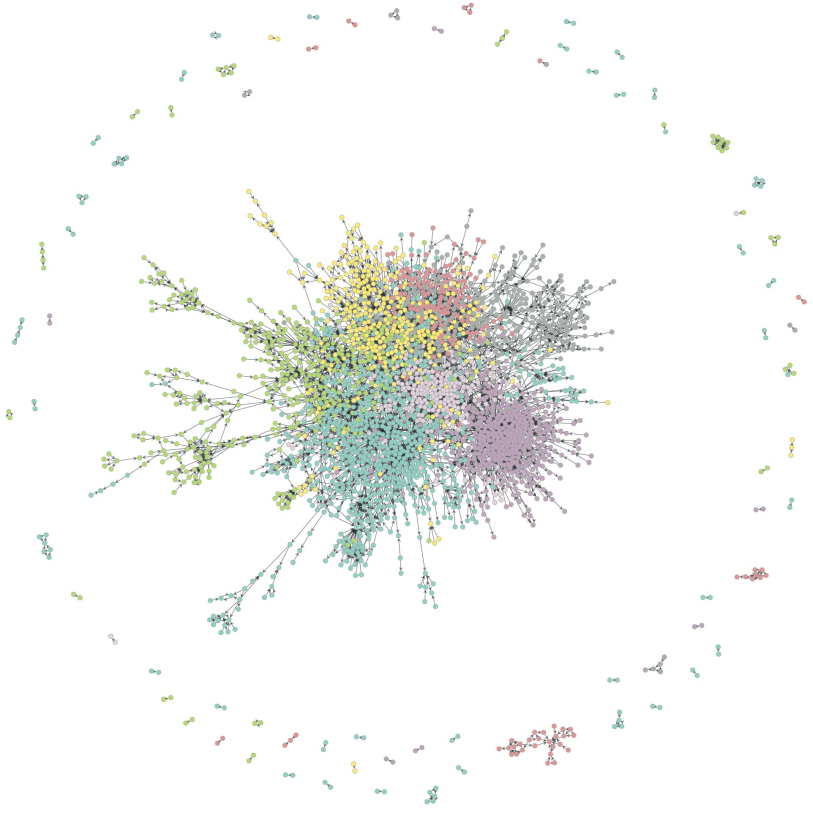
Training set: $V_{\text{train}} \subset V$
$\mathcal{X}_{\text{train}} = \mathcal{X}(V_{\text{train}})$
$\mathcal{Y}_{\text{train}} = \mathcal{Y}(V_{\text{train}})$

Find $\mathcal{Y}(V/V_{\text{train}})$ or $\mathcal{X}(V/V_{\text{train}})$

Community detection [PARS14, YCS16, KW17]; **recommendation systems** [YHC+18]; **molecular discovery/generation** [YLY+18]; **weakly-supervised learning** [KCL+19].

Example: Cora Dataset

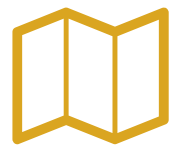
Most recent research have chosen Cora as the benchmark datasets.



What is their assumptions for dataset like Cora?

My answer: Low-frequency assumption!

Understanding the SOTA for Vertex Classification



Rayleigh Quotient

Given a symmetric Laplacian matrix $L \in \mathbb{R}^{n \times n}$ of a graph G, the Rayleigh quotient $R(L, f)$ for $f \in \mathbb{R}^n$ is given as:

$$R(L, f) = \frac{f^\top L f}{f^\top f} = \frac{1}{f^\top f} \sum_{u \backsim v} (f(u) - f(v))^2 \tag{1}$$

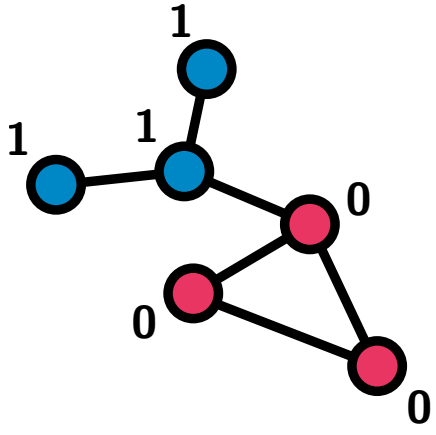


Figure: $R(L, f) = 1/3$. f is called "low-frequency".

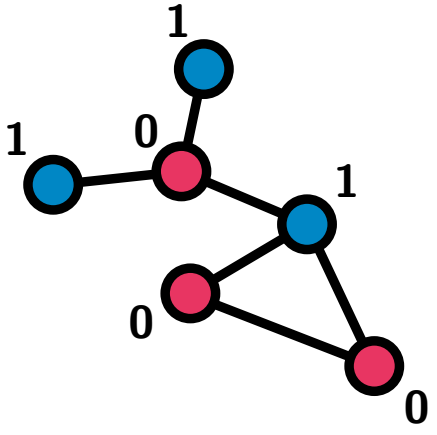


Figure: $R(L, f) = 5/3$. f is called "high-frequency".

Rayleigh quotient for $\mathcal{Y}$

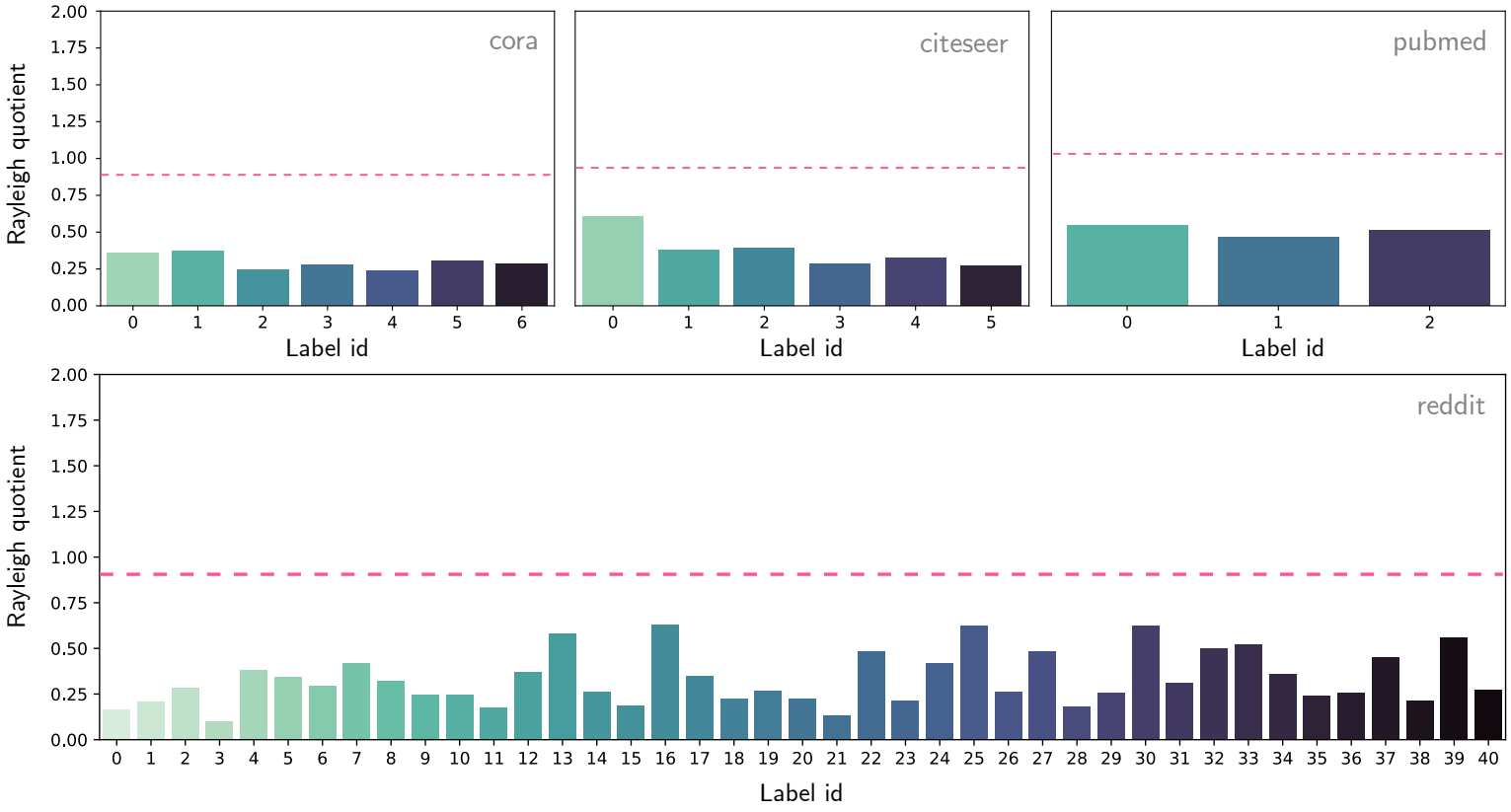


Figure: Rayleigh quotient of $\mathcal{Y}$ in benchmark datasets

Graph Low-pass Filters: Hard filter

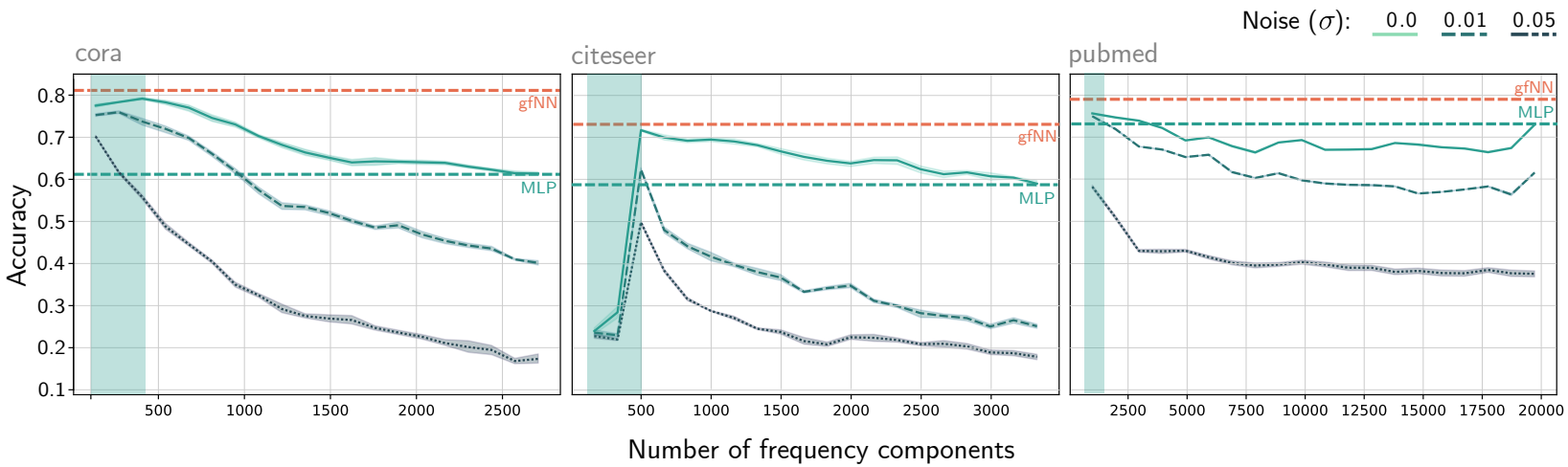


Figure: Classification accuracy by number of frequency components

The classification accuracy *increases* in the low-frequency regions for the benchmark datasets. In addition, this low-frequency regions (green boxes) are relatively noise tolerant.

Forgotten(?) Assumption

Two previous experiments show:

- Information is concentrated in the low-frequency regions.
- Rayleigh quotient can be used to predict the useful frequency regions.

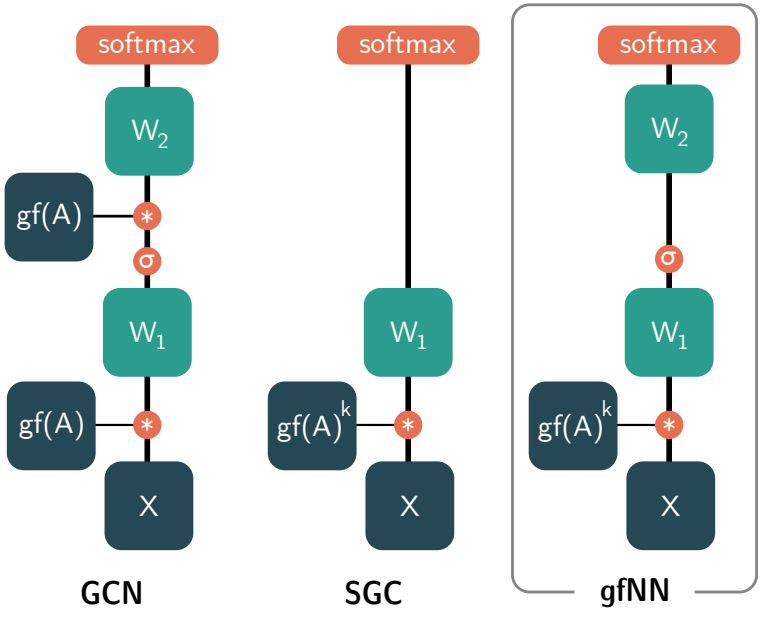
Assumption

In the vertex classification problem, we assume $R(L, y)$ to be sufficiently small. If $R(L, y)$ is large, the performances of most SOTA models are not guaranteed.

This is the "low-frequency" assumption. This assumption is also made for the feature $\mathcal{X}$.

Filter-then-classify

Most recent models can be generalized to "filter-then-classify" approach. The proposal of SGC [WZSJ+19] and the work by [LWL+19] support this observation.



$$h_{\text{GCN}} = W_2 \times \underbrace{\text{gf}(A)}_{\text{filter}} \times \sigma[W_1 \times \underbrace{\text{gf}(A)}_{\text{filter}} \times X]$$

$$h_{\text{SGC}} = W_1 \times \underbrace{\text{gf}(A)^k}_{\text{filter}} \times X$$

$$h_{\text{gfNN}} = W_2 \times \sigma[W_1 \times \text{gf}(A)^k \times X]$$

We will see that "filter-then-classify" has a few advantages to the feature propagation understanding.

Figure: Toy models.

Advantage in noisy settings

GCN and other multi-layers model might overfit to noisy data.

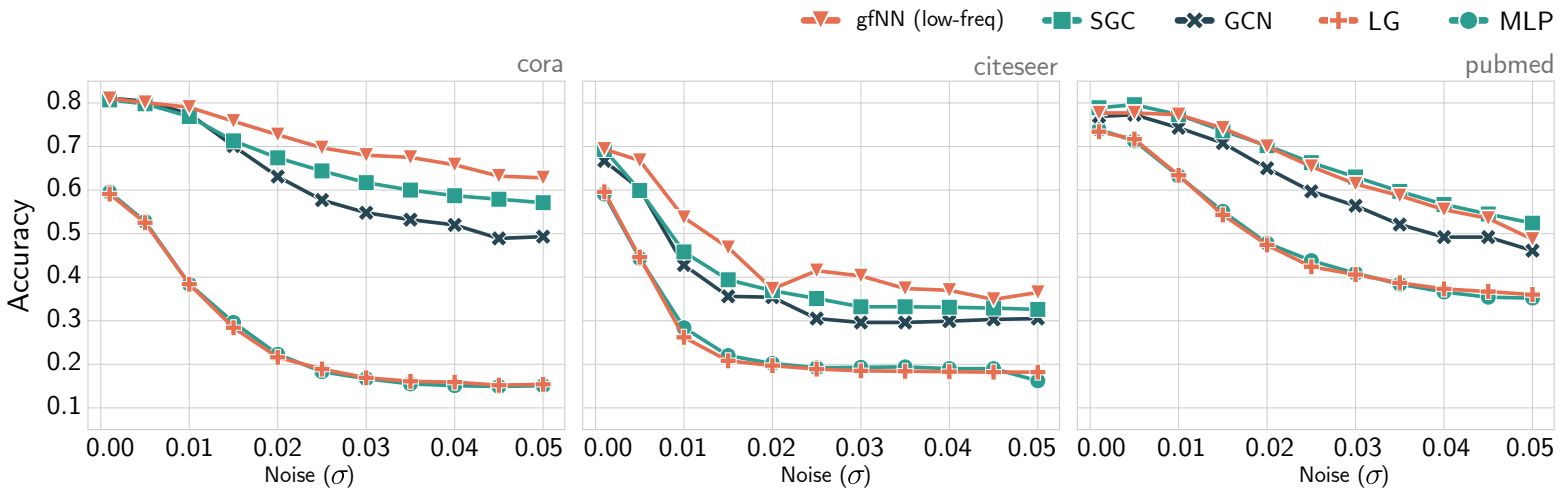


Figure: Add gaussian noise to features.

Advantage in decoupling functional parts (1)

Claim: Graph filters cannot "learn" manifolds!

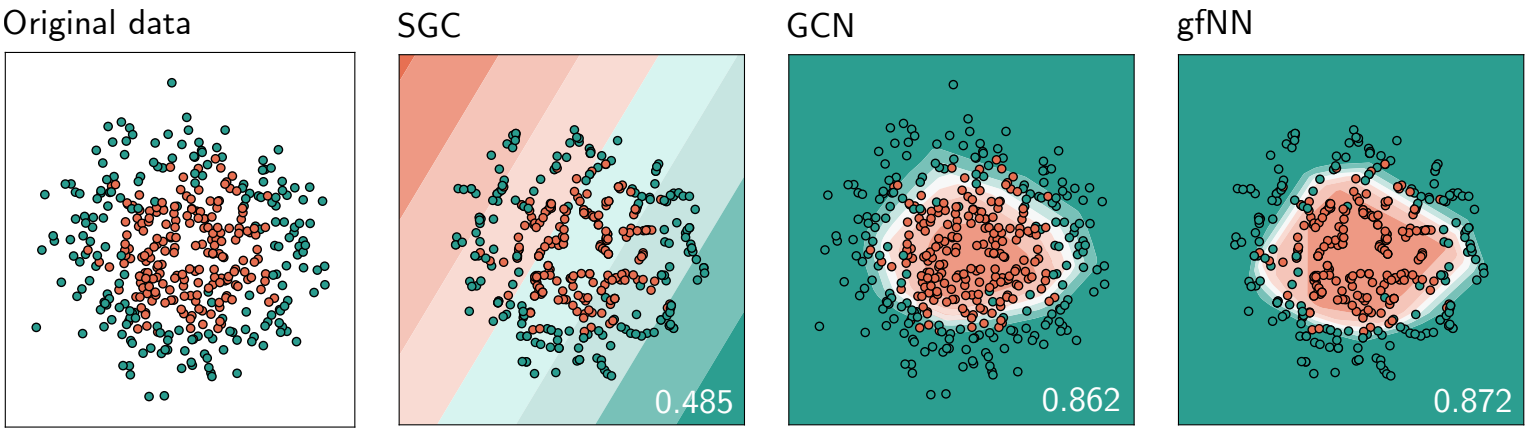


Figure: Results for donuts case.

Advantage in decoupling functional parts (2)

We have to learn the filters, not only the neural network's weights!

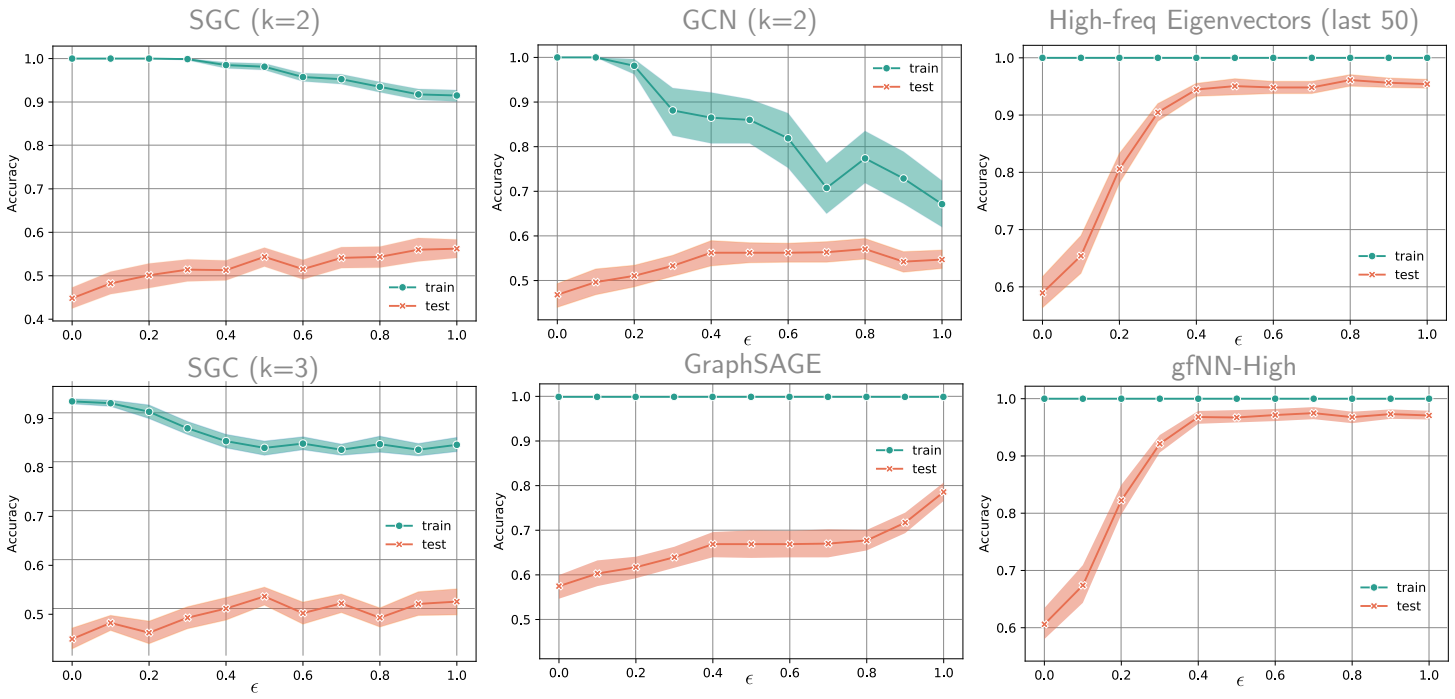
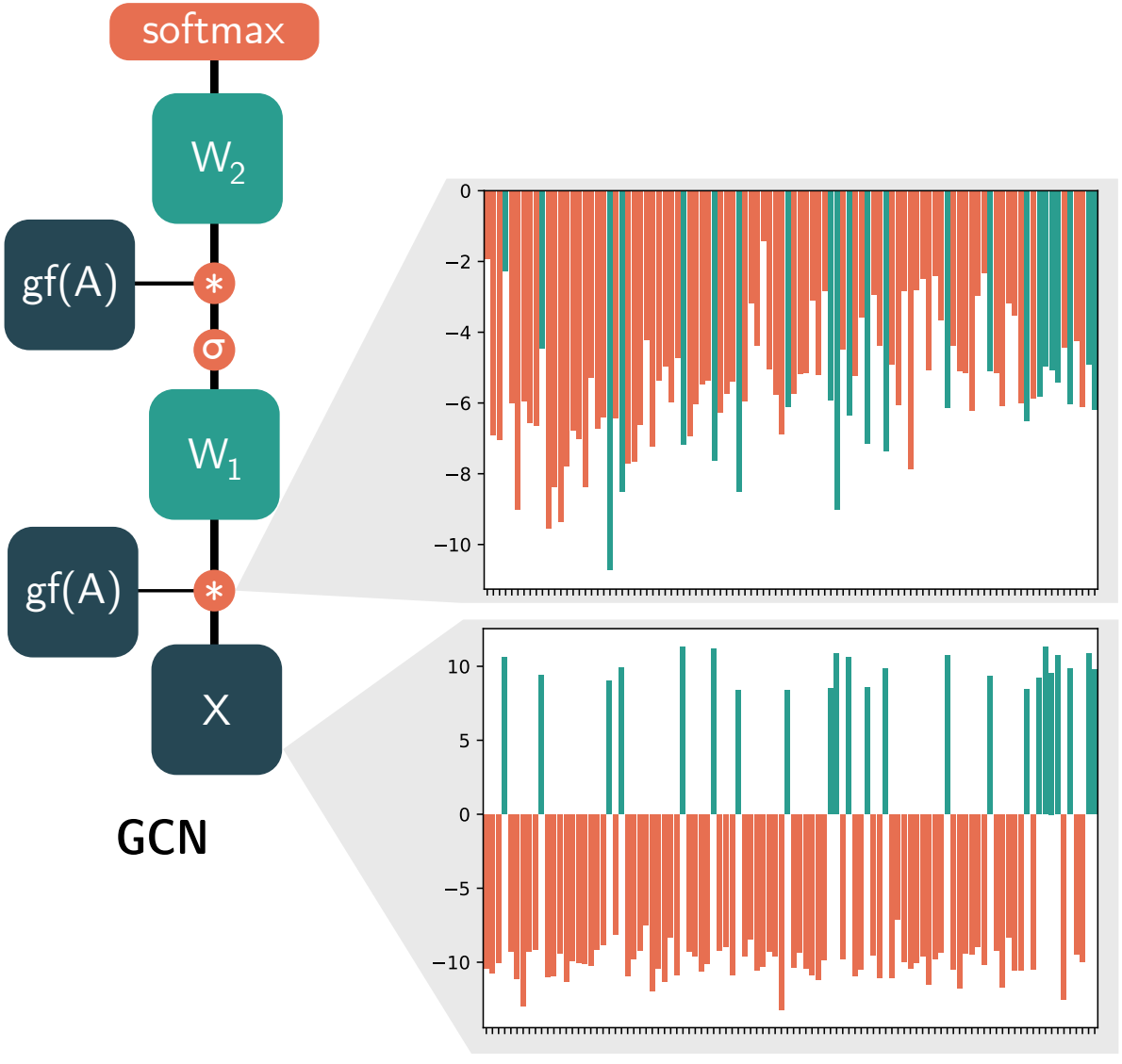


Figure: Results for high frequency case

In this setting, $R(\mathcal{L}, y) \approx 2$ (maximum value).

High Frequency: Feature Shift



Conclusions

1. Most benchmark datasets are community detection in nature, hence the designs for SOTA Graph Neural Networks are biased toward the low-frequency characteristics of these datasets.
2. A tool like Rayleight quotient and more flexible models like gfNN are needed in solving real-world vertex classifications.
3. High frequency cases (or different frequency case) are interesting because they can be used for constructing adversarial examples for graphs.
4. Disadvantage of filter-then-classify is that it doesn't provide an immediate intuition for the spacial domain. Also, currently selecting the appropriate graph filter for a problem beyond Decision Trees and Random Forest remains an open problem.

Next steps

Learning Theory for Vertex Classification

Viewing graph simply as a filter allows several directions:

- Statistical learning analysis: Quantify model complexity, number of samples for optimal training (somewhat similar to the Nyquist rate in SP and CS).
- Practical models: Adaptive filters with trade-off of data efficiency[1].

[1] https://arxiv.org/pdf/2011.10988.pdf

[Bha13] Rajendra Bhatia.
Matrix analysis, volume 169.
Springer Science & Business Media, 2013.

[DBV16] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst.
Convolutional neural networks on graphs with fast localized spectral filtering.
In *Advances in neural information processing systems*, pages 3844–3852, 2016.

[HYL17] Will Hamilton, Zhitao Ying, and Jure Leskovec.
Inductive representation learning on large graphs.
In *Advances in Neural Information Processing Systems*, pages 1024–1034, 2017.

[KCL+19] Michael Kampffmeyer, Yinbo Chen, Xiaodan Liang, Hao Wang, Yujia Zhang, and Eric P Xing.
Rethinking knowledge graph propagation for zero-shot learning.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11487–11496, 2019.

[KW17] Thomas N. Kipf and Max Welling.
Semi-supervised classification with graph convolutional networks.
In *International Conference on Learning Representations*, 2017.

[LWL+19] Qimai Li, Xiao-Ming Wu, Han Liu, Xiaotong Zhang, and Zhichao Guan.
Label efficient semi-supervised learning via graph filtering.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9582–9591, 2019.

[NJW02] Andrew Y Ng, Michael I Jordan, and Yair Weiss.
On spectral clustering: Analysis and an algorithm.
In *Advances in neural information processing systems*, pages 849–856, 2002.

[PARS14] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena.
Deepwalk: Online learning of social representations.
In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, 2014.

[WZSJ+19] Felix Wu, Tianyi Zhang, Amauri Holanda de Souza Jr., Christopher Fifty, Tao Yu, and Kilian Q. Weinberger.
Simplifying graph convolutional networks.
In *Proceedings of the 36th International Conference on International Conference on Machine Learning*, volume 97. JMLR, 2019.

[YCS16] Zhilin Yang, William W Cohen, and Ruslan Salakhutdinov.
Revisiting semi-supervised learning with graph embeddings.
In *Proceedings of the 33rd International Conference on International Conference on Machine Learning*, volume 48. JMLR, 2016.

[YHC+18] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec.
Graph convolutional neural networks for web-scale recommender systems.
In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 974–983, 2018.

[YLY+18] Jiaxuan You, Bowen Liu, Zhitao Ying, Vijay Pande, and Jure Leskovec.
Graph convolutional policy network for goal-directed molecular graph generation.
In *Advances in neural information processing systems*, pages 6410–6421, 2018.

Observation of models designs

Baseline vertex classification models such as spectral clustering [NJW02] often use first few eigenvectors to make feature vectors for vertices. More recent models such as Deepwalk [PARS14] or Planetoid [YCS16] relies on embedding neighbors "close" together.

Recent neural network based models such as ChebNet [DBV16], GCN [KW17], and GraphSAGE [HYL17] combine vertex features with graph structure by averaging neighbors (similar to feature propagation). **Common theme:** Low-frequency design!

Multiplying the feature vectors to $\Delta_{\text{sym}} = I - D^{-1/2}LD^{-1/2}$ is similar to applying the $1 - \lambda$ filter. Furthermore, adding loops to the graph truncates the largest eigenvector.

Theorem 1 ([WZSJ+19])

Let A be the adjacency matrix of an undirected, weighted, simple graph G without isolated nodes and with corresponding degree matrix D. Let $\tilde{A} = A + \gamma I$, such that $\gamma > 0$, be the augmented adjacency matrix with corresponding degree matrix D. Also, let λ_1 and λ_n denote the smallest and largest eigenvalues of $\Delta_{sym} = I - D^{-1/2}AD^{-1/2}$; similarly, let $\tilde{\lambda}_1$ and $\tilde{\lambda}_n$ be the smallest and largest eigenvalues of $\tilde{\Delta}_{sym}$. We have that

$$0 = \lambda_1 = \tilde{\lambda}_1 < \tilde{\lambda}_n < \lambda_n$$

Improving Theorem 3

Since [WZSJ+19] only proved for the largest eigenvalues, we do not know the relation between λ_i and $\tilde{\lambda}_i$ for $0 < i < n$.

Solution: Use the Courant–Fisher–Weyl's min-max principle to argue about other pairs of eigenvalues!

Theorem 2 (**NT**MM, 2019)

Let $\lambda_i(\gamma)$ be the i-th smallest generalized eigenvalue of $(\tilde{D}, L) = (D + \gamma I)$. Then, $\lambda_i(\gamma)$ is a non-negative number, and monotonically non-increasing in $\gamma \geq 0$. Moreover, $\lambda_i(\gamma)$ is strictly monotonically decreasing if $\lambda_i(0) \neq 0$.

Proof by [WZSJ+19] I

It is trivial to see that $\lambda_1 = \tilde{\lambda}_1 = 0$:

$$x^\top \tilde{\Delta}_{\text{sym}} x = \sum_i x_i^2 - \sum_i \sum_j \frac{\tilde{a}_{ij} x_i x_j}{\sqrt{(d_i + \gamma)(d_j + \gamma)}} \leq 0 \qquad (2)$$

Let $\beta_1 \leq \beta_2 \leq \ldots \leq \beta_n$ be the eigenvalues of $D^{-1/2} A D^{-1/2}$ and $\alpha_1 \leq \alpha_2 \leq \ldots \leq \alpha_n$ be the eigenvalues of $\tilde{D}^{-1/2} A \tilde{D}^{-1/2}$. We see that $\beta_1 < 0$. Choose x such that $||x|| = 1$ and $y = D^{1/2} \tilde{D}^{-1/2} x$,

Proof by [WZSJ+19] II

see that $||y||^2 = \sum_i \frac{d_i}{d_i+\gamma} x_i^2$ and $\frac{\min_i d_i}{\gamma+\min_i d_i} \leq ||y||^2 \leq \frac{\max_i d_i}{\gamma+\max_i d_i}$.
Using Rayleigh quotient to look at α_1:

$$\alpha_1 = \min_x (x^\top \tilde{D}^{-1/2} A \tilde{D}^{-1/2} x) \tag{3}$$

$$= \min_x (\frac{y^\top D^{-1/2} A D^{-1/2} y}{||y||^2} ||y||^2) \tag{4}$$

$$\geq \min_x (\frac{y^\top D^{-1/2} A D^{-1/2} y}{||y||^2}) \max_x (||y||^2) \tag{5}$$

$$= \beta_1 \max_x ||y||^2 \tag{6}$$

$$\geq \frac{\max_i d_i}{\gamma + \max_i d_i} \tag{7}$$

Proof by [WZSJ+19] III

Note that $\tilde{\Delta}_{\text{sym}} = I - \gamma\tilde{D}^{-1} - \tilde{D}^{-1/2}A\tilde{D}^{-1/2}$. Using the result above we have:

$$\begin{aligned}
\tilde{\lambda}_n &= \max_x x^\top (I - \gamma\tilde{D}^{-1} - \tilde{D}^{-1/2}A\tilde{D}^{-1/2})x && (8)\\
&\leq 1 - \min_x \gamma x^\top \tilde{D}^{-1}x - \min_x x^\top \tilde{D}^{-1/2}A\tilde{D}^{-1/2}x && (9)\\
&= 1 - \frac{\gamma}{\gamma + \max_i d_i} - \alpha_1 && (10)\\
&< 1 - \beta_1 = \lambda_n && (11)
\end{aligned}$$

My proof

Since the generalized eigenvalues of $(D + \gamma I, L)$ are the eigenvalues of a positive semidefinite matrix $(D + \gamma I)^{1/2} L (D + \gamma I)^{1/2}$, these are non-negative real numbers. To obtain the shrinking result, we use the Courant–Fisher–Weyl's min-max principle [Bha13, Corollary III. 1.2]: For any $0 \leq \gamma_1 < \gamma_2$,

$$\lambda_i(\gamma_2) = \min_{H:\text{subspace},\dim(H)=i} \max_{x \in H, x \neq 0} \frac{x^\top L x}{x^\top (D + \gamma_2 I) x} \tag{12}$$

$$\leq \min_{H:\text{subspace},\dim(H)=i} \max_{x \in H, x \neq 0} \frac{x^\top L x}{x^\top (D + \gamma_1 I) x} \tag{13}$$

$$= \lambda_i(\gamma_1). \tag{14}$$

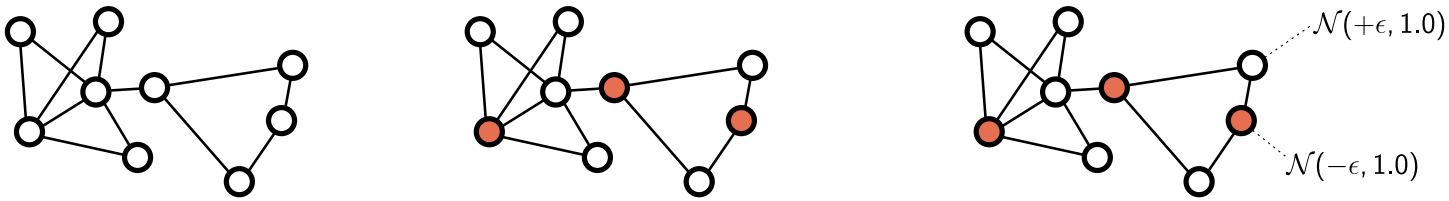
Here, the second inequality follows because $x^\top (D + \gamma_1) x < x^\top (D + \gamma_2) x$ for all $x \neq 0$ Hence, the inequality is strict if $x^\top L x \neq 0$, i.e., $\lambda_i(\gamma_1) \neq 0$.

Dataset details

Table: Real-world benchmark datasets and synthetic datasets for vertex classification

Dataset	**Nodes**	**Edges**	**Features (X)**	(μ_X, σ_X)	**Classes**	**Train/Val/Test**
Cora	2,708	5,278	1,433	(0.0007, 0.0071)	7	140/500/1,000
Citeseer	3,327	4,732	3,703	(0.0003, 0.0029)	6	120/500/1,000
Pubmed	19,717	44,338	500	(0.0019, 0.0087)	3	60/500/1,000
Reddit	231,443	11,606,919	602	-	41	151,708/23,699/55,334
PPI	56,944	818,716	50	-	121	44,906/6,514/5,524
Two Circles	4,000	10,000	2	-	2	80/80/3,840
BA-High	200	2000	50	(0,1)	2	10/10/180

High Frequency Artificial Data



Generate G | Find maximal indep. set | Assign random signal

Figure: Artificial BA with high-freq labels.

In this setting, $R(\mathcal{L}, y) \approx 2$ (maximum value).

Classification results

Table: Average test accuracy on original train/val/test splits (50 times)

	Cora	**Citeseer**	**Pubmed**	**Reddit**	**PPI**	**2Circles**	**BA-High**
DGI	83.1 ± 0.2	72.1 ± 0.1	80.1 ± 0.2	94.5 ± 0.3	99.2 ± 0.1	85.2 ± 0.6	54.6 ± 1.8
GCN	80.0 ± 1.8	69.6 ± 1.1	79.3 ± 1.3	-	-	84.9 ± 0.8	58.9 ± 2.2
SGC	77.6 ± 2.2	65.6 ± 0.1	78.4 ± 1.1	94.9 ± 0.2	89.0 ± 0.1	53.5 ± 1.4	55.5 ± 1.3
gfNN-low	82.3 ± 0.2	71.8 ± 0.1	79.2 ± 0.2	94.8 ± 0.2	89.3 ± 0.5	85.6 ± 0.8	55.4 ± 2.3
gfNN-high	24.2 ± 1.9	22.5 ± 2.2	43.6 ± 1.3	10.5 ± 2.6	86.6 ± 0.1	48.3 ± 3.5	96.2 ± 1.0
gf-Ensemble	82.9 ± 1.2	72.3 ± 1.2	81.5 ± 1.3	94.8 ± 0.2	88.2 ± 0.4	83.5 ± 0.3	95.7 ± 1.2

Compute Rayleigh quotient for $\mathcal{Y}$

We use the symmetric normalized Laplacian $\mathcal{L} = D^{-1/2}(D - A)D^{-1/2}$ and create a one-hot vector to indicate the label on each vertex.
For example, suppose we have a simple graph $G = (V, E)$, $|V| = n$, $|E| = m$, $\mathcal{Y} : V \mapsto \mathcal{C}$, and $|C| = 3$. We construct the one-hot matrix: $Y \in 0, 1^{n \times |C|}$. We denote $Y_i \in 0, 1^n$ as the column of the binary matrix Y. The Rayleight quotient for label i is given by:

$$R(\mathcal{L}, Y_i) = \frac{{Y_i}^\top \mathcal{L} Y_i}{{Y_i}^\top Y_i} = \frac{1}{{Y_i}^\top Y_i} \sum_{u \backsim v} (f(u) - f(v))^2$$

Graph Low-pass Filters: Hard filter

1. Compute the graph Fourier basis U from $\mathcal{L}$
2. Add Gaussian noise to the input features: $\mathcal{X} \leftarrow \mathcal{X} + \mathcal{N}(0, \sigma^2)$ for $\sigma = \{0, 0.01, 0.05\}$
3. Compute the first k-frequency component: $\hat{\mathcal{X}}_k = U[:k]^\top D^{1/2} \mathcal{X}$
4. Reconstruct the features: $\tilde{\mathcal{X}}_k = D^{-1/2} U[:k] \hat{\mathcal{X}}_k$
5. Train and report test accuracy of a 2-layers neural net on the reconstructed features $\tilde{\mathcal{X}}_k$