Learning to Approximate Adaptive Kernel Convolution on Graphs

Jaeyoon Sim[1] Sooyeon Jeon[1] InJun Choi[1] Guorong Wu[2] Won Hwa Kim[1]

[1]Pohang University of Science and Technology, Pohang, South Korea
[2]University of North Carolina at Chapel Hill, Chapel Hill, USA

Introduction

- **Key Idea: Training on the node-wise range of neighborhood efficiently instead of excessively stacking convolutional layer.**
- **Problem 1: Oversmoothed node representations.**
  - Most of existing GNNs have limitations such as oversmoothing, i.e., information becomes excessively averaged as the number of hidden layers increases.
- **Problem 2: Inefficient computation to construct diffusion kernel.**
  - Recent works leverage a diffusion kernel to redefine the graph structure and incorporate information from farther nodes. However, such methods suffer from heavy diagonlization of a graph Laplacian or learning a large transform matrix.
- **Contribution:**

  1. We propose a GNN with an adaptive diffusion kernel whose approximations are trainable in an end-to-end manner at each node.
  2. We derive closed-form derivatives of various polynomial coefficients with respect to the range (i.e., scale) so that graph convolution can be efficiently trained.
  3. Learning on scales provides interpretable results on the semantics of each node, validated on two independent datasets with different tasks.

Preliminary: Approximation of Convolution with Heat Kernel

- An undirected graph $G = \{V, E\}$ comprises a node set V with $|V| = N$ and an edge set E, and a graph Laplacian is defined as $L = D - A$ where D is a degree matrix and A is a symmetric adjacency matrix. An normalized Laplacian is defined as $\hat{L} = I_N - D^{-1/2}AD^{-1/2}$ where I_N is an identity matrix.
- Approximation of heat kernel convolution was introduced using several orthogonal polynomials (such as Chebyshev, Hermite, and Laguerre), and the analytic solutions to the polynomial coefficient $c_{s,n}$ for scale s were derived for corresponding polynomial P_n where n denotes the degree of each polynomial.
- Using convolution theorem, graph Fourier transform offers a way to define the graph convolution $*$ of a signal $x(p)$ with a filter h_s. The heat kernel $e^{-s\lambda}$ can be defined with P_n and $c_{s,n}$ as $e^{-s\lambda} = \sum_{n=0}^{\infty} c_{s,n}P_n(\lambda)$, and now the solution to the heat diffusion can be expressed in terms of P_n and $c_{s,n}$ as

$$h_s * x(p) = \sum_{n=0}^{\infty} c_{s,n}P_n(\hat{L})x(p). \tag{1}$$

Learning to Approximate Kernel Convolution

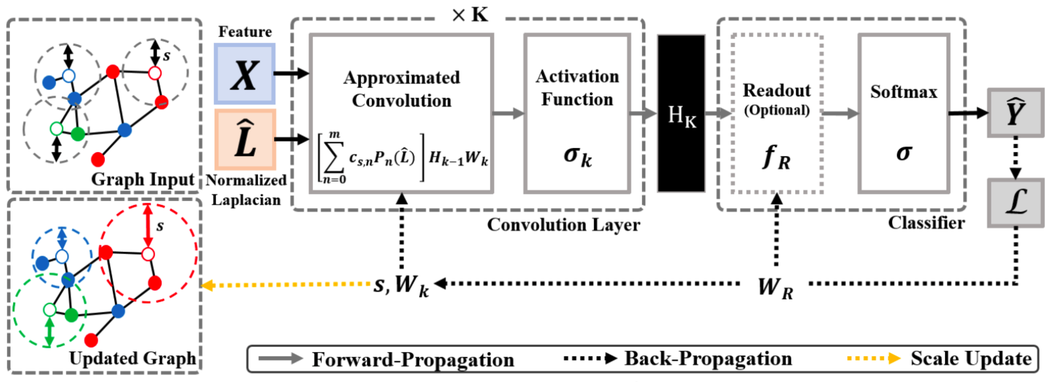

Figure: Illustration of LSAP. A graph (as normalized Laplacian $\hat{L}$) and node feature X are inputted to the convolution layer. The output H_K is inputted to a downstream classifier which yields a prediction $\hat{Y}$. The loss from $\hat{Y}$ is backpropagated to update the classifier and convolution approximation with $\mathbf{s} = [s_1, \ldots, s_N]$ to adaptively adjust the scale of each node.

- We introduce our model, i.e., LSAP, that *trains on the approximations* for the optimal range of neighborhood at individual nodes.
- **Convolution Layer:** We can reformulate convolution operation by replacing the adjacency matrix from most GCN frameworks with the heat kernel with polynomial approximation as

$$H_k = \sigma_k([\sum_{n=0}^{m} c_{s,n}P_n(\hat{L})]H_{k-1}W_k). \tag{2}$$

  While A let the model combine information from nodes within 1-hop distance, Eq.(2) let it aggregate information within a "range" of each node defined by s within $c_{s,n}$.
- **Gradients of Polynomial Coefficients with Scale:** In order to design a gradient-based "learning" framework of node-wise range (i.e., scale) based on coefficients, we derived gradients of loss with respect to the scale in closed-forms. The gradient can be achieved using the chain rule in a traditional way, and to obtain this gradient in terms of the H_k, we compute $\frac{\partial c_{s,n}}{\mathbf{s}}$ for each coefficient.
- **Model Update:** The loss is backpropagted to update the model parameters, and a multi-variate $\mathbf{s}$ across all nodes can be also trained as $\mathbf{s} \leftarrow \mathbf{s} - \beta_s \frac{\partial \mathcal{L}}{\partial \mathbf{s}}$.
  - The gradient on scale for semi-supervised node classification can be computed as

$$\begin{aligned}\frac{\partial \mathcal{L}_{\mathbf{err}}}{\partial \mathbf{s}} =& (\hat{Y} - Y) \times \sigma_k'([\sum_{n=0}^{m} c_{\mathbf{s},n}P_n(\hat{L})]H_{k-1}W_k)W_k^{\mathsf{T}} \\ & \times (\sum_{n=0}^{m} P_n(\hat{L})H_{k-1}^{\mathsf{T}} + [\sum_{n=0}^{m} c_{\mathbf{s},n}P_n(\hat{L})]\frac{\partial H_{k-1}}{\partial c_{\mathbf{s},n}})\frac{\partial c_{\mathbf{s},n}}{\partial \mathbf{s}}\end{aligned} \tag{3}$$

  - The gradient on scale for graph classification can be computed as

$$\begin{aligned}\frac{\partial \mathcal{L}_{\mathbf{err}}}{\partial \mathbf{s}} =& (\hat{Y} - Y) \times \frac{\partial H_R}{\partial H_K} \times \sigma_k'([\sum_{n=0}^{m} c_{\mathbf{s},n}P_n(\hat{L})]H_{k-1}W_k)W_k^{\mathsf{T}} \\ & \times (\sum_{n=0}^{m} P_n(\hat{L})H_{k-1}^{\mathsf{T}} + [\sum_{n=0}^{m} c_{\mathbf{s},n}P_n(\hat{L})]\frac{\partial H_{k-1}}{\partial c_{\mathbf{s},n}})\frac{\partial c_{\mathbf{s},n}}{\partial \mathbf{s}}\end{aligned} \tag{4}$$

  where H_R is output layer transforming node embeddings to a graph embedding.
- LSAP-C, LSAP-H, and LSAP-L correspond to approximation frameworks with each polynomial P_n and expansion coefficient $c_{s,n}$, and the model with exact computation of the heat kernel convolution is referred as Exact.

Node/Graph Classification Dataset

- Semi-supervised Node Classification - 6 standard datasets
- Graph Classification - Alzheimer's Disease Neuroimaging Initiative (ADNI) dataset
  - Diagnostic labels: Control (CN), Significant Memory Concern (SMC), Early/Late Mild Cognitive Impairment (EMCI/LMCI), Alzheimer's Disease (AD)

Dataset	Nodes	Edges	Classes	Features
Cora	2,708	5,429	7	1,433
Citeseer	3,327	4,732	6	3,703
Pubmed	19,717	44,338	3	500
Amazon Computers	13,752	245,861	10	767
Amazon Photo	7,650	119,081	8	745
Coauthor CS	18,333	81,894	15	6,805

Biomarker	Category	CN	SMC	EMCI	LMCI	AD
Cortical Thickness	# of subjects	359	181	437	180	166
	Gender (M / F)	178 / 181	69 / 112	249 / 188	119 / 61	102 / 64
	Age (Mean±Std)	72.8±1.4	72.0±5.2	71.0±7.9	70.9±6.1	74.8±8.7
FDG	# of subjects	345	186	461	231	162
	Gender (M / F)	173 / 172	66 / 120	262 / 199	152 / 79	102 / 60
	Age (Mean±Std)	73.0±1.3	71.7±5.2	71.7±7.8	71.1±7.0	74.9±8.8

Table: Left: Summary of node classification datasets. Right: Demographics of the ADNI dataset.

Classification Performance

Model	Cora	Citeseer	Pubmed
GCN	81.50	70.30	78.60
GAT	83.00	72.50	79.00
APPNP	83.30	71.80	80.10
GDC†	82.20	71.80	79.10
SGC	81.70	71.30	78.90
Bayesian GCN	81.20	72.20	-
Shoestring	81.90	69.50	79.70
GraphHeat†	83.70	72.50	80.50
g-U-Nets	84.40	73.20	79.60
GCNII	85.50	73.40	80.30
GRAND	85.40	75.40	82.70
DAGNN	84.40	73.30	80.50
SelfSAGCN	83.80	73.50	80.70
DIAL-GNN	84.50	74.10	-
SuperGAT	84.30	72.60	81.70
GRAND†	83.60	74.10	78.80
ADC†	84.50	74.50	83.00
SEP-N	84.80	72.90	80.20
LSAP-C	**87.90**	**76.50**	83.30
LSAP-H	85.00	76.10	82.60
LSAP-L	85.90	75.90	**84.10**
Exact	88.20	78.10	85.30

†: graph diffusion-based models.

Feature	Model	Classification (ADNI)		
		Accuracy (%)	Precision	Recall
Cortical Thickness	SVM (Linear)	82.39 ± 2.73	0.822 ± 0.033	0.852 ± 0.025
	MLP (2-layers)	78.76 ± 2.21	0.792 ± 0.036	0.799 ± 0.026
	GCN	61.37 ± 3.09	0.598 ± 0.025	0.626 ± 0.044
	GAT	64.17 ± 5.46	0.627 ± 0.067	0.668 ± 0.046
	GDC	77.10 ± 4.25	0.769 ± 0.050	0.785 ± 0.044
	GraphHeat	70.90 ± 3.17	0.703 ± 0.030	0.718 ± 0.026
	ADC	82.10 ± 2.41	0.776 ± 0.019	0.728 ± 0.067
	LSAP-C	**87.00 ± 2.16**	**0.868 ± 0.027**	**0.885 ± 0.027**
	LSAP-H	85.41 ± 2.32	0.859 ± 0.031	0.867 ± 0.030
	LSAP-L	85.64 ± 1.86	0.859 ± 0.022	0.866 ± 0.022
	Exact	86.24 ± 1.96	0.866 ± 0.017	0.867 ± 0.023
FDG	SVM (Linear)	85.27 ± 2.09	0.857 ± 0.027	0.869 ± 0.021
	MLP (2-layers)	87.51 ± 1.62	0.882 ± 0.024	0.882 ± 0.014
	GCN	68.81 ± 1.95	0.677 ± 0.028	0.697 ± 0.025
	GAT	69.24 ± 7.13	0.670 ± 0.106	0.736 ± 0.037
	GDC	86.21 ± 3.24	0.867 ± 0.033	0.870 ± 0.029
	GraphHeat	76.97 ± 2.42	0.775 ± 0.035	0.773 ± 0.010
	ADC	88.60 ± 2.81	0.708 ± 0.062	0.753 ± 0.053
	LSAP-C	89.24 ± 2.23	0.895 ± 0.022	0.904 ± 0.023
	LSAP-H	90.11 ± 2.44	0.903 ± 0.027	0.910 ± 0.022
	LSAP-L	**90.40 ± 1.38**	**0.909 ± 0.018**	**0.914 ± 0.015**
	Exact	90.18 ± 2.67	0.907 ± 0.028	0.907 ± 0.028

Table: Left: Accuracy (%) on Cora, Citeseer, and Pubmed. LSAP yields better performances over existing baselines (in bold) similar to Exact achieving the best results (underline). Right: Classification performances on ADNI dataset (for CN / SMC / EMCI / LMCI / AD).

Model Behavior Analysis

- **Scales for Graph Classification**
  - The trained model yields node-wise optimized scale where the node corresponds to specific regions of interest (ROI) in the brain.

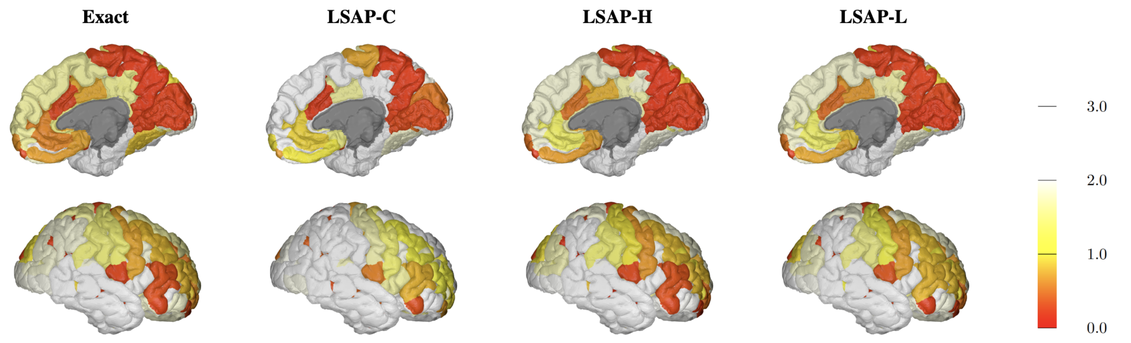

Figure: Visualization of the learned scales on the cortical regions of a brain. This visualization shows the scale of each ROI from the classification result using FDG feature. Top: Inner part of right hemisphere, Bottom: Outer part of right hemisphere.

- **Computation Time with Kernel Convolution**
  - We compared averaged empirical time (in *ms*) spent for one epoch of training process Exact and LSAP on node and graph classification task with 10 replicates.
- **Effect of K**
  - We examined the performance of LSAP with respect to the number of convolution layers K on Cora and ADNI experiments.

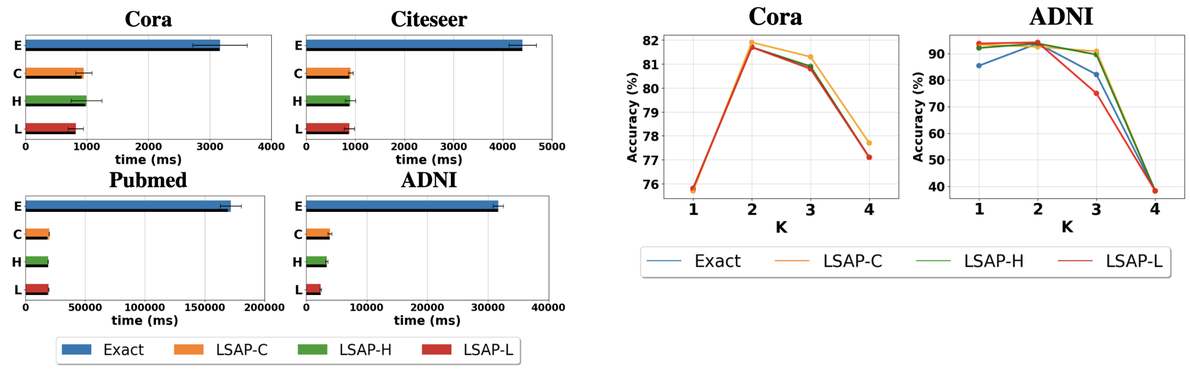

Figure: Left: Comparisons of computation time (in *ms*) for one epoch (Forward and backpropagation). Within the epoch, time for heat kernel convolution is given in black bar. Results were obtained with 10 repetitions. Right: Effect of the number of layers K on model performance (Cora and ADNI).

Conclusion

In this work, we proposed efficient trainable methods to bypass exact computation of spectral kernel convolution that define adaptive ranges of neighbor for each node. We have derived closed-form derivatives on polynomial coefficients to train the scale with conventional backpropagation, and the developed framework LSAP demonstrates *SOTA* performance on node classification and brain network classification. The brain network analysis provides neuroscientifically interpretable results corroborated by previous AD literature.

Acknowledgment

This research was supported by NRF-2022R1A2C2092336 (50%), IITP-2022-0-00290 (20%), IITP-2019-0-01906 (AI Graduate Program at POSTECH, 10%) funded by MSIT, HU22C0171 (10%), HU22C0168 (10%) funded by MOHW from South Korea, and NSF IIS CRII 1948510 from the U.S.