ACML 2023

Graph Structure Learning via Lottery Hypothesis at Scale

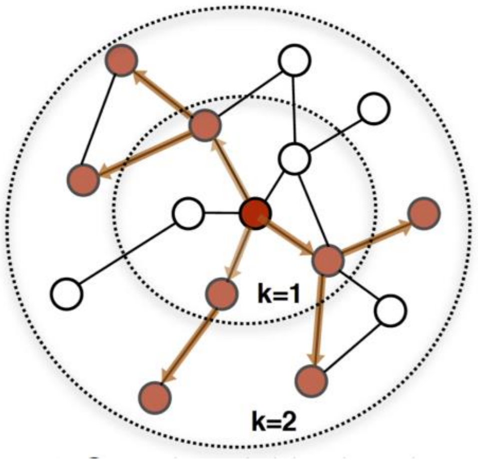
Yuxin Wang, Xiannian Hu*, Jiaqing Xie*, Zhangyue Yin*, Yunhua Zhou, Xuanjing Huang, Xipeng Qiu

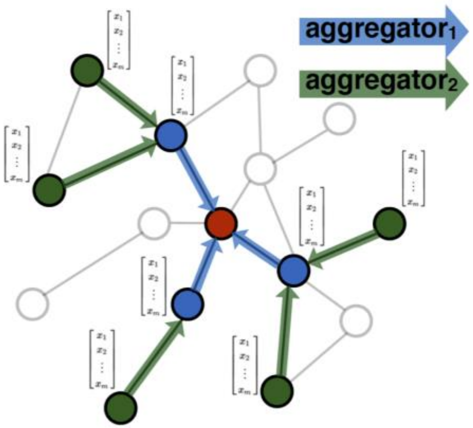
November 12th, 2023

Terminology Review

- Graph Neural Networks
- Graph Structure Learning
- Graph Attack / Defense
- Lottery Ticket Hypothesis

Graph Neural Networks





William L. et al. 2017

Graph Structure Learning

Main Idea: Learn Better Graph Structures

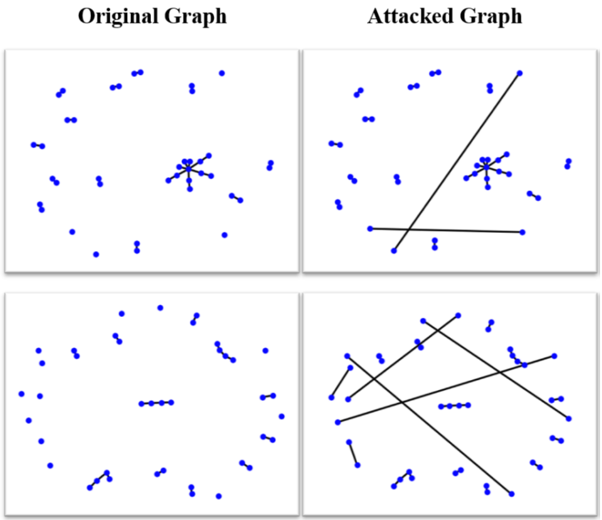
Methods:

- Similarity Metrics
- Graph Sparsification
- Graph Regularization
- Learning Paradigms

Graph(NN) Attack

Examples on Graph-level Attack:

- Remove / Add nodes (Xu et al. 2019a, Wu et al. 2019)



An example on Model-level Attack:

- Training surrogate models (Zugner et al. 2019)

Graph Defense

- Graph purification: an example of graph defense
- Previous graph purification are regularization methods

Disadvantage:

- Lack of scalability (Chen et al. 2020a, Jin et al. 2020)
- Top-k-features: not ensure dense graphs (Entezari et al. 2020)

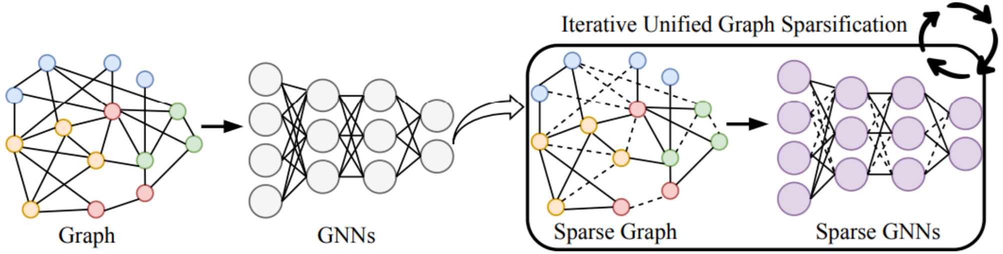
Lottery Ticket Hypothesis

DNN:

Find a pruned small sub-network that performed on par with the original large-scale network.

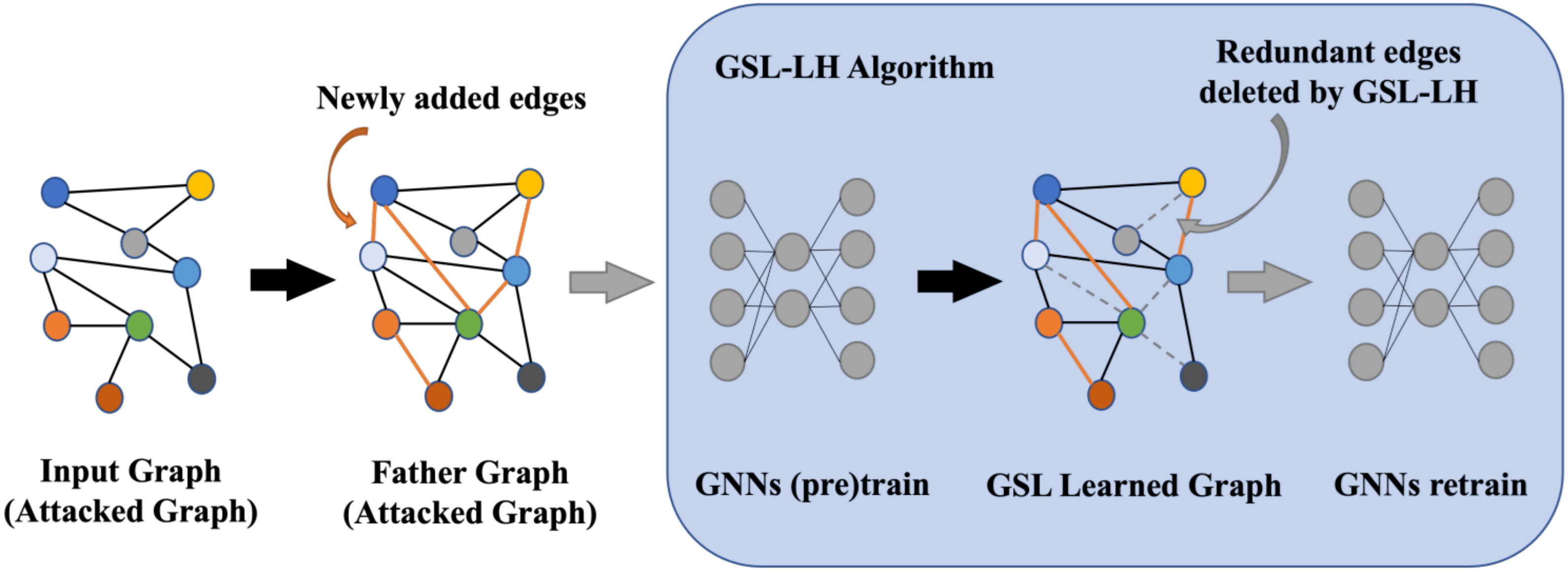
GNN:

Treat entries of adjacency matrix as parameters. Prune this matrix simultaneously with graph neural networks.



Chen et al. 2021

Graph Structure Learning with Lottery Ticket



Graph Structure Learning with Lottery Ticket

Algorithm 1 GSL-LH algorithms

Input: Feature $\boldsymbol{X}$, label $\boldsymbol{y}$, input father graph $\mathcal{G}_F$, graph neural network $f(\boldsymbol{\Theta}, \mathbf{A}; \cdot)$, adjacent mask logits $\boldsymbol{\pi_{adj}}$, weight mask logits $\boldsymbol{\pi_{\Theta}}$, adjacent sparsity s_{adj}, weight sparsity s_{Θ}, number of steps for model and lottery search training N_1, N_2.

Stage 1: full model (pre-) train
Get model initialization $\boldsymbol{\Theta}_0$, $\mathbf{A}_F$.
for n=1 **to** N_1 **do**
  Update f with $\mathcal{L}_{\text{CE}}(\boldsymbol{\Theta}, \mathbf{A}_F) := \sum_{i,c} y_{i,c} \log f(\boldsymbol{\Theta}, \mathbf{A}_F; x_i)_c$.
end for
Stage 2: subgraph and subnetwork lottery searching
Initialize weight mask logits $\boldsymbol{\pi_{\Theta}}$ and adjacent mask logits $\boldsymbol{\pi_{adj}}$ to $\mathbf{1}$.
for n=1 **to** N_2 **do**
  Update module $\boldsymbol{\pi_{\Theta}}$ and $\boldsymbol{\pi_{adj}}$ with
    $\mathcal{L}_{GSL} = \mathcal{L}_{\text{CE}}(\boldsymbol{\pi_{adj}} \odot \boldsymbol{A}_F, \boldsymbol{\pi_{\Theta}} \odot \boldsymbol{\Theta}_*)$.
end for
Stage 3: subgraph and subnetwork retrain
Obtain the module by pruning with sparsity s_{adj} and s_{Θ}.
    $\boldsymbol{m}_{adj} = \text{Prune}(\boldsymbol{\pi_{adj}}, s_{adj})$
    $\boldsymbol{m}_{\Theta} = \text{Prune}(\boldsymbol{\pi_{\Theta}}, s_{\Theta})$.
Set model parameters back to $\boldsymbol{\Theta}_0$.
for n=1 **to** N_1 **do**
  Update f with $\mathcal{L}_{\text{CE}}(\boldsymbol{m}_{adj} \odot \boldsymbol{A_F}, \boldsymbol{m}_{\Theta} \odot \boldsymbol{\Theta}_0)$.
end for

Sampling Methods

- Random Sampling
  - Sample randomly from neighbour nodes
- Feature Sampling
  - Sample by thresholding similarity of node pairs
- Lottery Sampling
  - Obtain attention matrix P by normalizing inner products of node pairs

$$\boldsymbol{P}(i,j) = \boldsymbol{v_i}^T \boldsymbol{v_j}$$

  - Randomly sample r. If r in $\left(\sum_{k=1}^{j-1} \boldsymbol{P}(i,k), \sum_{k=1}^{j} \boldsymbol{P}(i,k)\right]$, add j to neighbours of i

Time Complexity Analysis

- Time Complexity for pruning and masking: O(E)
- Training GNN: O(E) Wu et al. 2020
- Overall O(E)
- Better than graph decomposition
  - Example 1: spectral decomposition O(N^3) >> O(E)
  - Example 2: decomposition by cuts O(N^2 log N) >> O(E)

Baselines

- GAT
- GCN
- RGCN
- GCN-Jaccard
- GCN-SVD
- Pro-GNN-fs

Datasets

- Cora
- Cora-ML
- Citeseer
- PubMed
- Arxiv

Results

Dataset	Ptb Rate (%)	GCN	GAT	RGCN	GCN-Jaccard	GCN-SVD	Pro-GNN-fs	GSL-LH
Cora	0	83.50±0.44	83.97±0.65	83.09±0.44	82.05±0.51	80.63±0.45	83.42±0.52	**84.33±0.27**
	5	76.55±0.79	80.44±0.74	77.42±0.39	79.13±0.59	78.39±0.54	**82.78±0.39**	82.00±0.27
	10	70.39±1.28	75.61±0.59	72.22±0.38	75.16±0.76	71.47±0.83	77.91±0.86	**79.31±0.49**
	15	65.10±0.71	69.78±1.28	66.82±0.39	71.03±0.64	66.69±1.18	76.01±1.12	**77.95±0.40**
	20	59.56±2.72	59.94±0.92	59.27±0.37	65.71±0.89	58.94±1.13	68.78±5.84	**74.97±0.31**
	25	47.53±1.96	54.78±0.74	50.51±0.78	60.82±1.08	52.06±1.19	56.54±2.58	**68.92±0.60**
Cora ML	0	85.25±0.33	85.49±0.24	86.48±0.16	84.82±0.27	80.97±0.31	85.06±0.33	**86.50±0.17**
	5	79.19±0.36	81.06±0.59	81.62±0.14	80.23±0.40	80.23±0.34	83.25±0.71	**85.95±0.15**
	10	73.83±0.45	76.26±0.99	74.46±0.18	75.21±0.23	80.61±0.37	81.52±0.93	**84.78±0.10**
	15	54.35±0.66	57.96±1.34	54.87±0.33	57.45±0.65	**73.54±0.43**	53.81±0.27	71.66±0.30
	20	43.11±3.30	42.69±1.21	46.62±0.66	45.77±1.30	46.94±1.69	47.54±0.37	**70.37±1.70**
	25	48.45±0.48	43.74±4.44	50.15±0.36	49.05±0.41	56.28±0.86	50.99±0.27	**71.37±0.74**
Citeseer	0	71.96±0.55	73.26±0.83	71.20±0.83	72.10±0.63	70.65±0.32	73.26±0.38	**76.78±0.29**
	5	70.88±0.62	72.89±0.83	70.50±0.43	70.51±0.97	68.84±0.72	73.09±0.34	**74.96±0.25**
	10	67.55±0.89	70.63±0.48	67.71±0.30	69.54±0.56	68.87±0.62	72.43±0.52	**74.37±0.31**
	15	64.52±1.11	69.02±1.09	65.69±0.37	65.95±0.94	63.26±0.96	70.82±0.87	**71.73±0.56**
	20	62.03±3.49	61.04±1.52	62.49±1.22	59.30±1.40	58.55±1.09	66.19±2.38	**66.26±0.60**
	25	56.94±2.09	61.85±1.12	55.35±0.66	59.89±1.47	57.18±1.87	66.40±2.57	**66.77±0.37**
Pubmed	0	87.19±0.09	83.73±0.40	86.16±0.18	87.06±0.06	83.44±0.21	87.33±0.18	**87.46±0.05**
	5	83.09±0.13	78.00±0.44	81.08±0.20	86.39±0.06	83.41±0.15	87.25±0.09	**87.27±0.05**
	10	81.21±0.09	74.93±0.38	77.51±0.27	85.70±0.07	83.27±0.21	**87.25±0.09**	87.18±0.06
	15	78.66±0.12	71.13±0.51	73.91±0.25	84.76±0.08	83.10±0.18	**87.20±0.09**	86.90±0.03
	20	77.35±0.19	68.21±0.96	71.18±0.31	83.88±0.05	83.01±0.22	**87.09±0.10**	86.61±0.05
	25	75.50±0.17	65.41±0.77	67.95±0.15	83.66±0.06	82.72±0.18	**86.71±0.09**	86.37±0.07
Arxiv	0	71.03±0.27	**71.18±0.11**	-	-	-	-	70.90±0.20
	5	53.78±0.23	**55.74±0.27**	-	-	-	-	54.56±0.49
	10	46.83±0.15	48.68±0.33	-	-	-	-	**50.12±0.67**
	15	42.68±0.23	44.34±0.44	-	-	-	-	**50.95±0.07**
	20	39.61±0.53	41.36±0.20	-	-	-	-	**50.40±0.27**
	25	37.51±0.17	39.36±0.51	-	-	-	-	**50.02±0.20**

Table 2: Node classification performance (Accuracy±Std) under attack. We use metattack for regular-size graphs and PR-BCD for Arxiv. "-" means not applicable. Performances of all baselines in Cora ML are not available and are run by ourselves. Other baseline performances are from Pro-GNN Jin et al. (2020). Bold symbols and underlines mean the first and second best performances respectively.

Results

Ptb Rate (%)	GCN	GSL-LH	PPRGO	SoftMedian+PPRGO	GSL-LH+PPRGO
5	53.43 ± 0.27	54.56 ± 0.49	**58.18 ± 0.23**	57.24 ± 0.16	57.96 ± 0.48
10	46.75 ± 0.47	50.12 ± 0.67	53.39 ± 0.24	55.39 ± 0.29	**56.37 ± 0.47**
15	43.39 ± 0.62	50.95 ± 0.07	51.27 ± 0.17	54.56 ± 0.31	**54.83 ± 0.43**
20	40.17 ± 0.63	50.40 ± 0.27	50.31 ± 0.32	**54.40 ± 0.12**	53.98 ± 0.42
25	37.77 ± 0.65	50.02 ± 0.24	48.58 ± 0.40	**54.59 ± 0.26**	53.34 ± 0.40

Table 3: Node classification performance (Accuracy±Std) on ogbn-Arxiv under PR-BCD of different methods based on PPRGO.

Ablation Study

- Sampling

Ptb Rate (%)	Random	Feature	Lottery
5	52.84 ± 1.33	53.73 ± 0.63	**54.56 ± 0.49**
10	49.54 ± 0.71	49.20 ± 3.70	**50.12 ± 0.67**
15	50.70 ± 0.16	50.93 ± 0.29	**50.95 ± 0.07**
20	50.08 ± 0.19	45.27 ± 7.34	**50.40 ± 0.27**
25	49.69 ± 0.07	**50.22 ± 0.24**	50.02 ± 0.20

Table 4: Performances of different sampling methods in GSL-LH under different perturbation rates.

Ablation Study

- Prune sparsity

Acc. Adj Sparsity \ Weight Sparsity	None	0.2	0.4	0.6	0.8
None	**54.56 ± 0.49**	53.28 ± 1.53	52.73 ± 1.44	51.84 ± 0.85	49.76 ± 1.21
0.2	51.85 ± 0.52	50.89 ± 1.32	51.20 ± 0.73	51.02 ± 0.84	47.65 ± 1.15
0.4	48.90 ± 0.96	49.31 ± 0.52	47.45 ± 1.20	48.20 ± 1.16	47.02 ± 0.41
0.6	51.14 ± 0.17	49.80 ± 0.43	47.72 ± 1.44	43.39 ± 3.29	44.12 ± 3.78
0.8	52.34 ± 0.11	50.75 ± 0.62	48.60 ± 1.75	43.57 ± 4.59	11.00 ± 11.50

Table 5: Performances of different adjacent sparsity and weight sparsity under perturbation rate of 0.05. None means we don't prune the mask and maintain the trained scores in the model retrain step.

Acc. Adj Sparsity \ Weight Sparsity	None	0.2	0.4	0.6	0.8
None	42.12 ± 1.67	41.57 ± 0.79	41.15 ± 0.45	42.12 ± 0.21	39.49 ± 1.25
0.2	41.33 ± 0.24	41.56 ± 0.19	40.17 ± 0.55	40.68 ± 0.18	25.14 ± 8.01
0.4	43.55 ± 0.21	43.57 ± 0.35	41.49 ± 1.07	38.79 ± 2.43	25.77 ± 9.12
0.6	46.36 ± 0.17	45.25 ± 0.41	43.51 ± 1.20	36.55 ± 5.23	26.12 ± 10.20
0.8	**50.02 ± 0.20**	48.55 ± 0.71	46.36 ± 1.72	41.37 ± 4.16	26.62 ± 11.33

Table 6: Performances of different adjacent sparsity and weight sparsity under perturbation rate of 0.25. None means we don't prune the mask and maintain the trained scores in the model retrain step.

Future works

- Test with more recent advanced model (Dual Graph Lottery Ticket)
  - Kun et al. ICLR 2023
- Test on more large-scale graph datasets

Resources

Code Repo: https://github.com/jiaqingxie/GSL-LH

Paper: Proceedings of Machine Learning Research 222, 2023