Provable Robustness of GCN

Outlines

- Attack approaches for graph structure perturbation
- Problem we want to prove
- Optimize on graph structure
- Abstract Interpretation
- Compare with existing recent work

Deep Learning on Graphs

- GNN is the state-of-art on tasks such as
  - **Semi-supervised node classification**
  - Link prediction
  - Unsupervised representation learning (e.g., node embeddings)

Graph Neural Networks are not robust

Summary of adversarial attack works on graph data (time ascending).

Ref.	Year	Venue	Task	Model	Strategy	Approach	Baseline	Metric	Dataset
[27]	2017	CCS	Graph Clustering	SVD, Node2vec, Community detection algs	Noise injection, Small community attack	Add/Delete edges	-	ASR, FPR	NXDOMAIN, Reverse Engineered DGA Domains
[108]	2018	Nature Human Behavior	Hide nodes and communities in a graph	Community detection algs	Heuristic	Rewire edges	-	Concealment measures, Graph statistics	WTC 9/11, Scale-free Facebook, Twitter, Google+, Random
[145]	2018	KDD	Node classification	GCN, CLN, DeepWalk	Incremental attack	Add/Delete edges, Modify node features	Random, FGSM	Accuracy, Classifcation margin	Cora-ML, Citeseer, PolBlogs
[28]	2018	ICML	Graph classification, Node classification	GNN family models	Reinforcement learning	Add/Delete edges	Rnd. sampling, Genetic algs.	Accuracy	Citeseer,Finance, Pubmed, Cora
[109]	2018	Scientific Reports	Link prediction	Similarity measures	Heuristic	Add/Delete edges	-	AUC, AP	WTC 9/11, Random, Scale-Free, Facebook
[23]	2018	arXiv	Node classification, Community detection	DeepWalk, GCN, Node2vec, LINE	Check GCN gradients	Rewire edges	Random, DICE, Nettack	ASR, AML	Cora, Citeseer, PolBlogs
[106]	2018	arXiv	Node classification	GCN	Greedy, GAN	Add fake nodes with fake features	Random, Nettack	Accuracy, F1, ASR	Cora, Citeseer
[92]	2018	arXiv	Link prediction	GAE, DeepWalk, Node2vec, LINE	Project gradient descent	Add/Delete edges	Degree sum, Shortest path, Random, PageRank	AP, Similarity score	Cora, Citeseer, Facebook
[39]	2018	ACSAC	Recommender system	Random walk recommender algs	Optimization	Add nodes&edges	Bandwagon, Co-visitation, Random, Average	HR@N	MovieLens 100K, Amazon Video
[8]	2019	ICML	Node classification, Link prediction	Node2vec, GCN LP, DeepWalk	Check gradient, Approximate spectrum	Add/Delete edges	Random, Degree, Eigenvalue	F1 score, Misclassification rate	Cora, Citeseer, PolBlogs
[147]	2019	ICLR	Node classification	GCN, CLN DeepWalk	Meta learning	Add/Delete edges	DICE, Nettack, First-order attack	Accuracy, Misclassification rate	Cora, Pubmed, Citeseer, PolBlogs
[143]	2019	AAMAS	Link prediction	Local&Global Similarity measures	Submodular	Hide edges	Random, Greedy	Similarity score	Random, Facebook
[18]	2019	TCSS	Community detection	Community detection algs	Genetic algs	Rewire edges	Random, Degree, Community detection	NMI, Modularity	Karate, Dolphin, Football, Polbooks
[103]	2019	CCS	Node classification	LinBP, LBP, JW, DeepWalk, LINE, GCN, RW, Node2vec	Optimization	Add/Delete edges	Random, Nettack	FNR, FPR	Google+, Epinions, Twitter, Facebook, Enron
[136]	2019	IJCAI	Knowledge graph fact plausibility prediction	RESCAL, TransE, TransR	Check target entity embeddings	Add/Delete fact	Random	MRR, Hit Rate@K	FB15k, WN18
[2]	2019	arXiv	Vertex nomination	VN-GMM-ASE	Random	Add/Delete edges	-	Achieving rank	Bing entity transition graph
[12]	2019	arXiv	Node classification	GCN	Adversarial generation	Modify node features	Nettack	ASR	Cora, Citeseer

Cited from Lichao, Bo et. al.

GNN Node Classification

- Given adjacency matrix A and node feature X.
  - Node classification is a task such that finds the labels of each node with A and X. That is

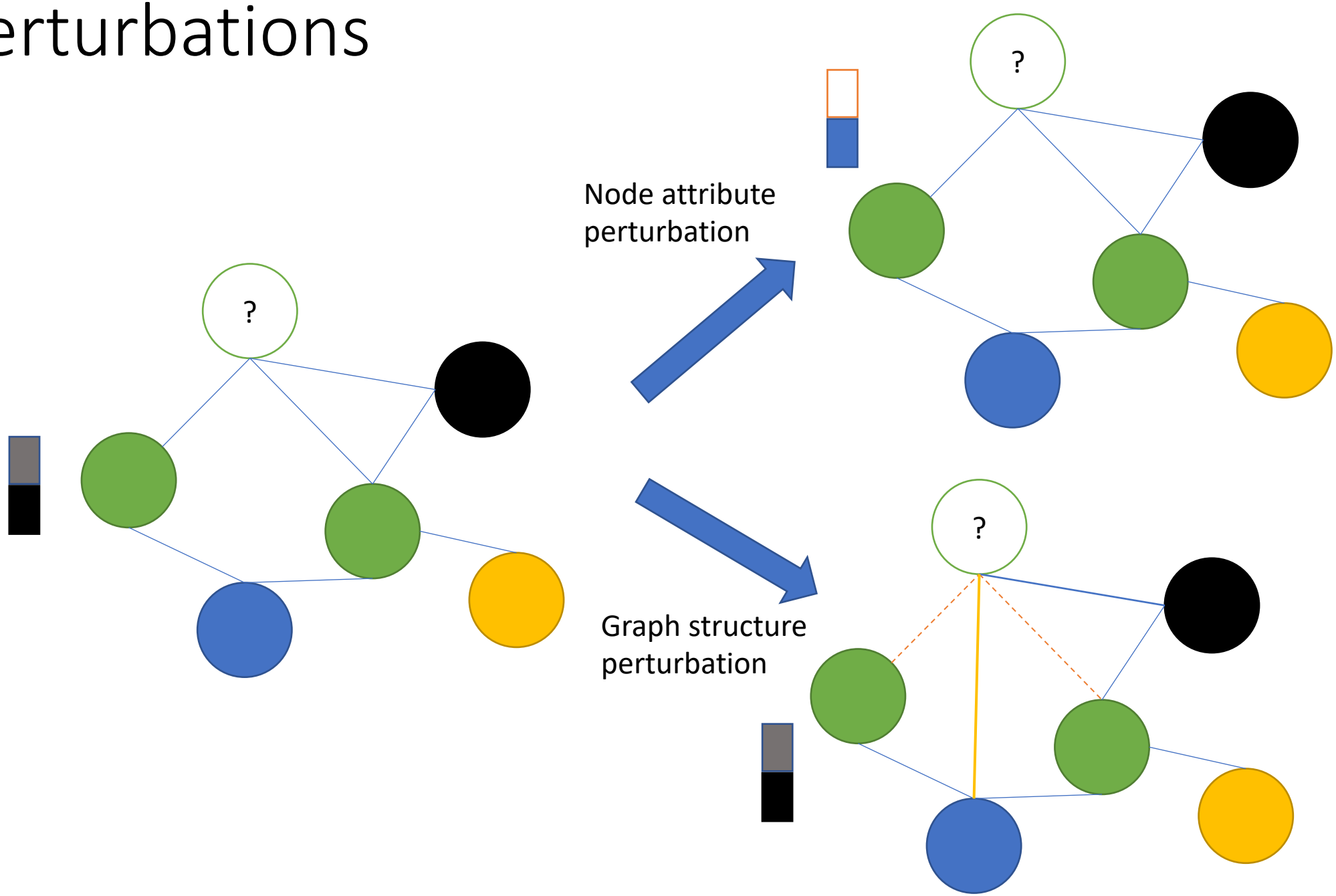
$$\hat{Y} = f(A, X).$$

Where f is the classifier, and $\hat{Y}$ is the predicted labels of each node.

E.g., 2-layer *GCN:*

$$\hat{Y} = softmax\left(\hat{A}\,\sigma\left(\hat{A}\,X\,W^{1} + b^{1}\right)W^{2} + b^{2}\right)$$

Perturbations



Graph Structure Perturbation

Allowed Perturbation $\mathcal{P}(A)$

- At most q adversarial edges can be added per node **(local budget)**
- At most Q adversarial edges cab be added in total (over all nodes, **global budget**)

$$\mathcal{P}(A) = \{A' \mid \forall i, j \in V, |A_{ij} - A'_{ij}| < q \\ \wedge \ |A - A'|_0 < 2Q\}$$

Certificate Score $CS(y, y^*)$

$$CS(y, y^*) = \underset{A' \in \mathcal{P}(A)}{minimize}\ f_\theta(A', X)_{y^*} - f_\theta(A', X)_y$$

Prove that $\boldsymbol{CS(y, y^*) > 0}$

Optimize on Graph Structure

Enumerating all allowed perturbation $\mathcal{P}(A)$ is intractable, even for small budget

$$\mathcal{P}(A) = \{A' \mid \forall i, j \in V, |A_{ij} - A'_{ij}| < q \\ \wedge \ \ |A - A'|_0 < 2Q\}$$

Linear programming and continuous relaxation on A? (I.e., $A_{ij} \in R$)

Optimize on Graph Structure

Enumerating all allowed perturbation $\mathcal{P}(A)$ is intractable, even for small q and Q

$$\mathcal{P}(A) = \{A' \mid \forall i, j \in V, |A_{ij} - A'_{ij}| < q \\ \wedge \ |A - A'|_0 < 2Q\}$$

Continuous relaxation on A? (I.e., $A_{ij} \in R$)

Continuous relaxation is useful, but not that simple.

In GCN, we use degree-normalized message passing matrix

$$\hat{A}_{ij} = \begin{cases} \frac{1}{\sqrt{d_i d_j}} & A_{ij} = 1 \vee i = j \\ 0 & else \end{cases}$$

But $\hat{A}_{ij}$ is **non-convex**, still hard to get a "**provable**" result.

budgets for $\hat{A}$

Recall that the local budget q and global budgets Q of allowed purtabation set $\mathcal{P}(A)$ is defined on adjacency matrix A.

At most q adversarial edges are added for per node

At most Q adversarial edges are added in total

$$\hat{P}(A) = \{\hat{A} \in [0,1]^{N\times N} | \hat{A} = \hat{A}^T \wedge \forall i,j, L_{ij} < \hat{A}_{ij} < U_{ij}$$
$$\wedge \forall i: \; L_i^r \leq \sum_j \hat{A}_{ij} \leq U_i^r \wedge \forall i \sum_j |\hat{A}_{ij} - \hat{A}_{ij}^0| \leq \bar{U}_i^r$$
$$\wedge \forall i < j, \sum |\hat{A}_{ij} - \hat{A}_{ij}^0| \leq \bar{U}_i^{global}\}$$

Certifiable Robustness of Graph Convolutional Networks under Structure Perturbations [KDD'20]

$L_{ij}, U_{ij}, L_i^r, \bar{U}_i^{global}$ are functions related with q, Q and A which can be pre-computed

Continuous relaxation on $\hat{A}$ and linear constraints!

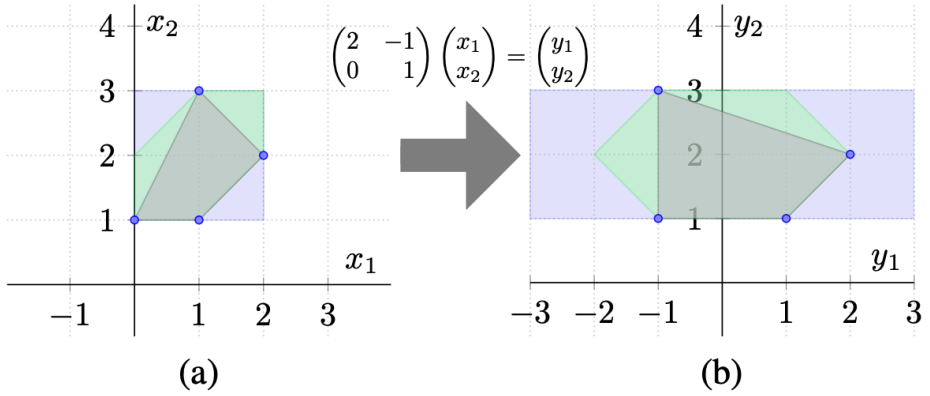
But the problem is still non-convex because the activation layer of neural network is nonconvex

Abstract Interpretation for Neural Network

Only Implemented ReLU for now based on DEEPOLY [POPL 2019].

But theoretically it can support all activate functions.

AI intuition: express **all possible inputs** with **linear constraints** and **propagate** them through different functions, while keep the constraints are linear.



Abstract Interpretation (AI)

x = [-1,1]

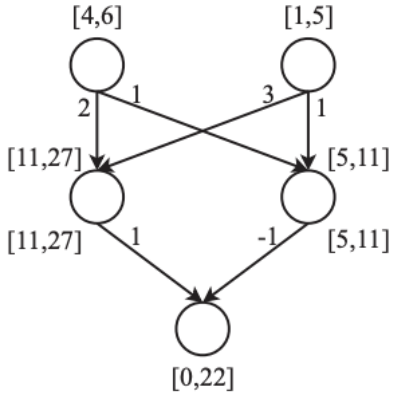
f = x - x = ?

Naive interval:
f = [-1,1] - [-1,1] = [-2,2]

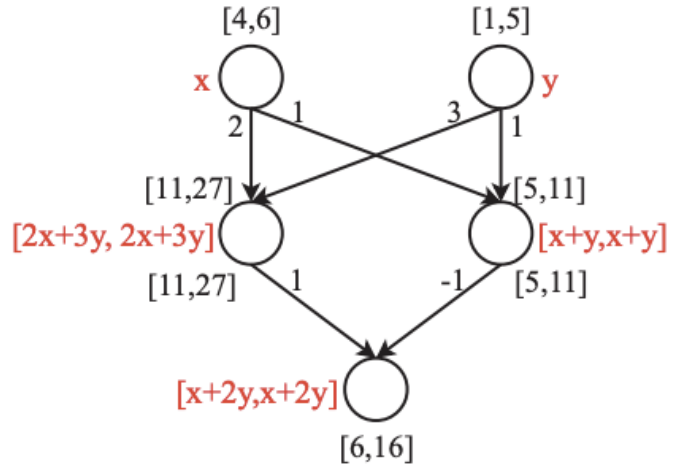
True bound:
f = x – x = [0,0]

How to propagate is tricky.

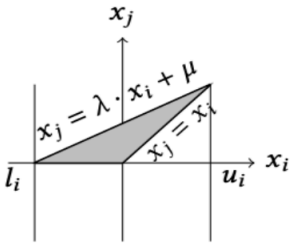
Deepoly Explained



(a) Naive interval propagation

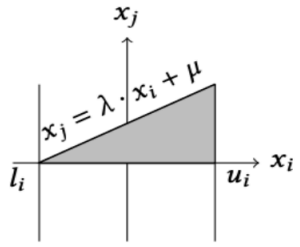


(b) Symbolic interval propagation



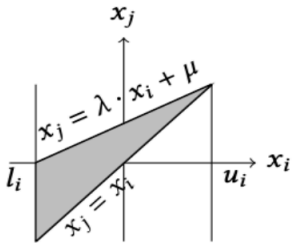
(a)

$$x_i \le x_j, 0 \le x_j,$$
$$x_j \le u_i(x_i - l_i)/(u_i - l_i).$$
$$l_j = 0, u_j = u_i$$



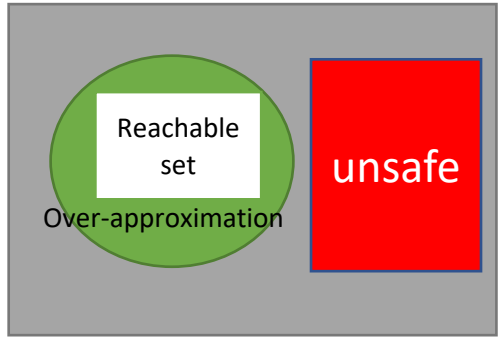
(b)

$$0 \le x_j,$$
$$x_j \le u_i(x_i - l_i)/(u_i - l_i),$$
$$l_j = 0, \; u_j = u_i$$



(c)

$$x_i \le x_j,$$
$$x_j \le u_i(x_i - l_i)/(u_i - l_i),$$
$$l_j = l_i, \; u_j = u_i$$



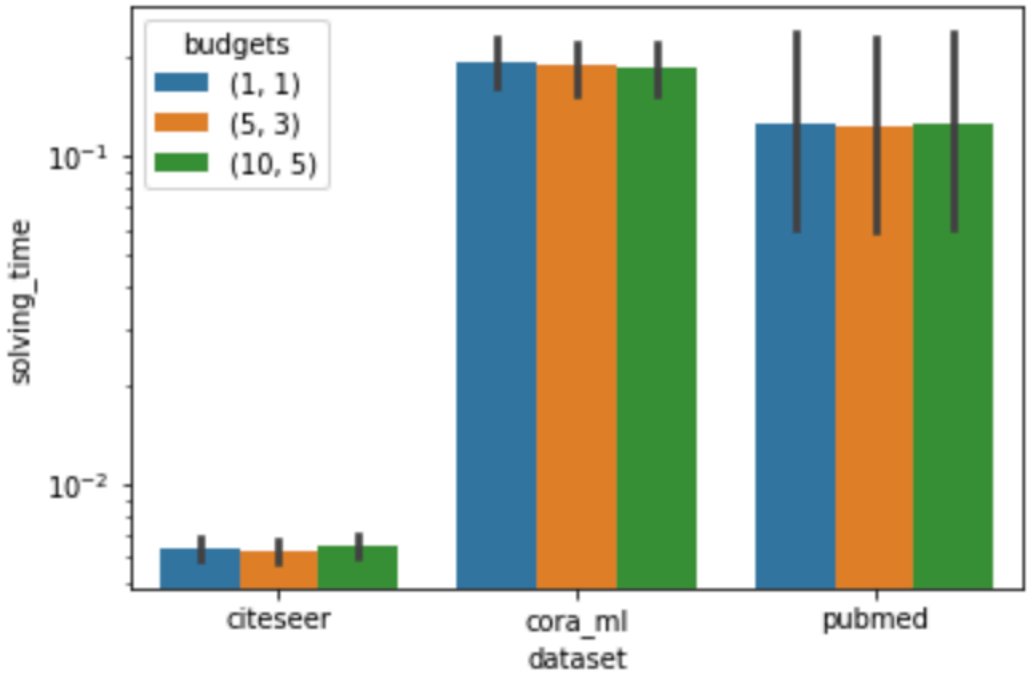
Approach Summary

- $CS(y, y^*) = \underset{A' \in \mathcal{P}(A)}{minimize}\ f_\theta(A', X)_{y^*}$ - $f_\theta(A', X)_y$
- Continuous relaxation on $\hat{A}$ and get linear constraints
- Apply abstract interpretation to get over approximated output domain (i.e., a batch of inequation).
- Finally, we we get a batch of linear constraints.
- Lower bound of $CS(*, y^*) > 0$, proved robust
- Upper bound of $CS(*, y^*) < 0$, proved not robust
- Unsure about other cases

Results

	dataset	budgets	cert	uncert	not_sure
0	citeseer	(1, 1)	0.712	0.016	0.272
1	citeseer	(5, 3)	0.676	0.012	0.312
2	citeseer	(10, 5)	0.472	0.005	0.523
3	cora_ml	(1, 1)	0.843	0.000	0.157
4	cora_ml	(5, 3)	0.769	0.000	0.231
5	cora_ml	(10, 5)	0.704	0.000	0.296
6	pubmed	(1, 1)	0.855	0.015	0.130
7	pubmed	(5, 3)	0.646	0.003	0.351
8	pubmed	(10, 5)	0.612	0.004	0.384

Uncert means any perturbation will make the nodes label change.

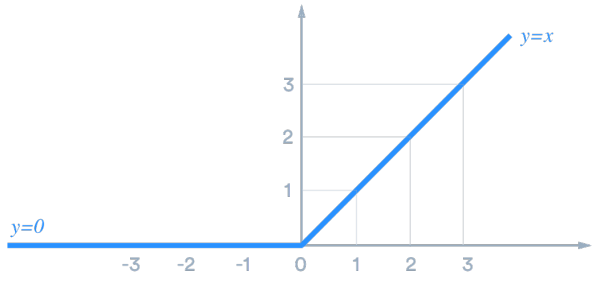


Very fast, less then 1 seconds.

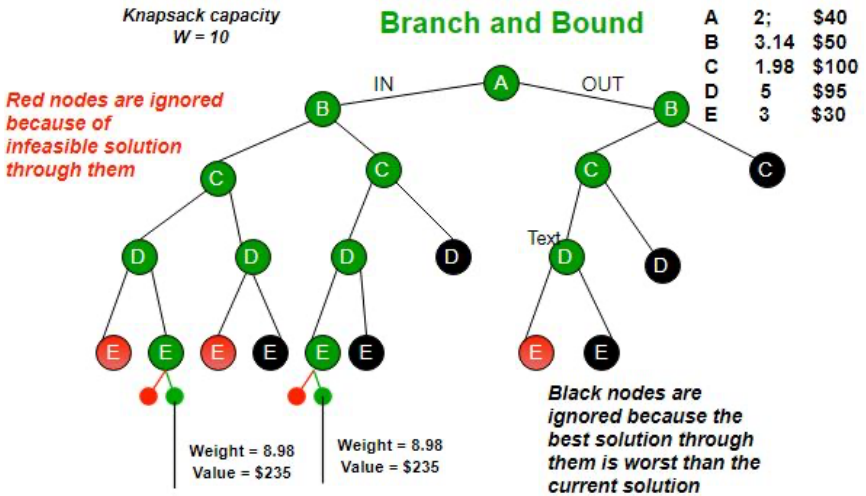
Alternative Approach for Abstract Interpretation - B & B

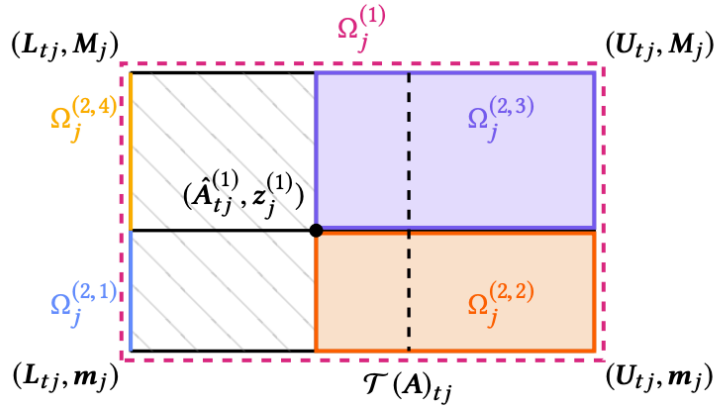
- Certifiable Robustness of Graph Convolutional Networks under Structure Perturbations [KDD'20]
- Only support ReLU
- Running time is significant longer than abstract interpretation.

Branch & Bound High-Level Overview



Split ReLU, solve “path-by-path”

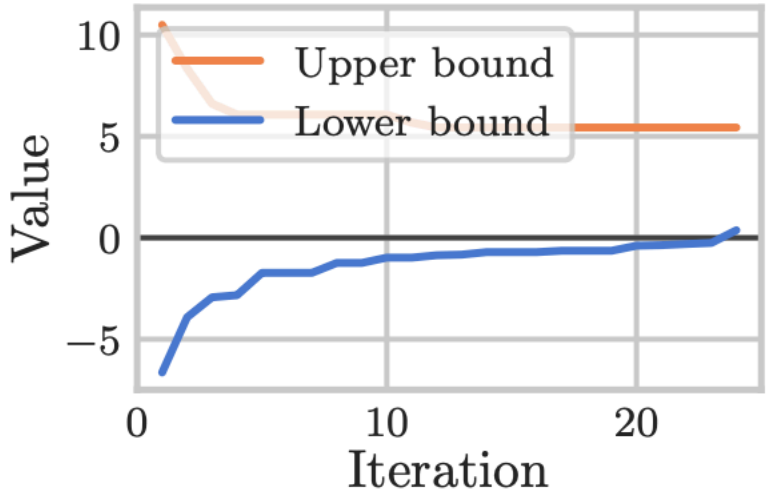




When solving LP, we need to check corner points, if any the best case in a region cannot be better than previous results, than pass the rest part.

Early Stop

- $CS(y, y^*) = \underset{A' \in \mathcal{P}(A)}{minimize}\ f_\theta(A', X)_{y^*} - f_\theta(A', X)_y$



Still very very slow…

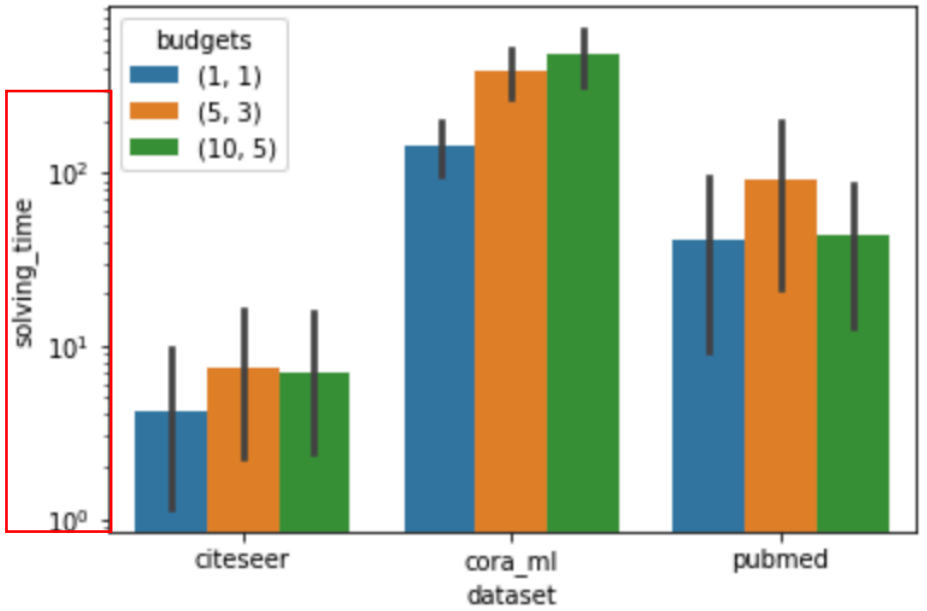
Results

	dataset	budgets	cert	uncert	not_sure
0	citeseer	(1, 1)	0.91	0.09	0.00
1	citeseer	(5, 3)	0.79	0.14	0.07
2	citeseer	(10, 5)	0.75	0.15	0.10
3	cora_ml	(1, 1)	0.78	0.20	0.02
4	cora_ml	(5, 3)	0.39	0.36	0.25
5	cora_ml	(10, 5)	0.17	0.38	0.46
6	pubmed	(1, 1)	0.74	0.24	0.02
7	pubmed	(5, 3)	0.49	0.44	0.07
8	pubmed	(10, 5)	0.46	0.49	0.05

B&B

	dataset	budgets	cert	uncert	not_sure
0	citeseer	(1, 1)	0.712	0.016	0.272
1	citeseer	(5, 3)	0.676	0.012	0.312
2	citeseer	(10, 5)	0.472	0.005	0.523
3	cora_ml	(1, 1)	0.843	0.000	0.157
4	cora_ml	(5, 3)	0.769	0.000	0.231
5	cora_ml	(10, 5)	0.704	0.000	0.296
6	pubmed	(1, 1)	0.855	0.015	0.130
7	pubmed	(5, 3)	0.646	0.003	0.351
8	pubmed	(10, 5)	0.612	0.004	0.384

abstract-interpretation



More than 1000x time spent than abstract-interpretation-based approach

Outlines

- Attack approach for graph structure perturbation
- Problem we want to prove
- Optimize on graph structure
- Abstract Interpretation
- Compare with existing recent work

Thank you & Questions