Semi-Supervised Classification with Graph Convolutional Networks

Thomas N. Kipf Max Welling

Presented by Alexander Urzhumov

01.06.2018

Talk Outline

- Classification nodes in graph
  - NN and semi-supervised
- Spectral graph convolutions
  - Spectral convolutions
  - Approximation
  - GCN
- Semi supervised with node classification
- Experiments
- Results

Classification nodes in graph

- Problem of classification
- X – feature matrix, A – adjacency matrix
- Any differentiable function for classification
- Special loss function

$$\mathcal{L} = \mathcal{L}_0 + \lambda \mathcal{L}_{\text{reg}},$$

$$\mathcal{L}_{\text{reg}} = \sum A_{ij} ||f(X_i) - f\left(X_j\right)|| = f(X)^T \Delta \mathrm{f}(\mathrm{X})$$

$$\Delta = D - A \quad \textit{Graph Laplacian}$$

Classification nodes in graph

NN and semi-supervised

- f(X, A) - model
- $\mathcal{L} = \mathcal{L}_0 - \text{loss function}$ on labeled data
- Main idea: use A to distribute information from $\mathcal{L}_0$ and learn on labeled and unlabeled nodes

Spectral Graph Convolutions

spectral convolutions

$$L = I_N - D^{-\frac{1}{2}} A D^{-\frac{1}{2}} = U\Lambda \mathrm{U}^{\mathrm{T}} \ - \text{normalized graph Laplacian}$$
$$g_\theta = diag(\theta) - filter, \qquad \theta \ - parameters$$
$$x \in \ \mathbb{R}^N - signal, \qquad scalar\ for\ every\ node$$
$$g_\theta \star x = U g_\theta U^T x - spectral\ convolutions$$

Complexity O(N^2)

Spectral Graph Convolutions
approximation

$$T_k(x) = 2xT_{k-1} - T_{k-2}(x) - Chebyshev\ polynomials$$

$$g_{\theta'}(\Lambda) \approx \sum \theta'_k T_k(\tilde{\Lambda}), \qquad \tilde{\Lambda} = \frac{2}{\lambda_{\max}}\lambda - I_N$$

$$g_{\theta'} \star x \approx \sum \theta'_k T_k(\tilde{\mathrm{L}})x, \qquad \tilde{\mathrm{L}} = \frac{2}{\lambda_{\max}}L - I_N$$

Complexity O(|E|)

Spectral Graph Convolutions

further approximation and generalization

$$K = 2, \qquad \lambda_{max} \approx 2$$

$$g_{\theta'} \star x \approx \theta_0' x - \theta_1' D^{-\frac{1}{2}} A D^{-\frac{1}{2}} x \approx \theta' \left(I_N + D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \right) x = \theta \widetilde{D}^{-\frac{1}{2}} \tilde{A} \widetilde{D}^{-\frac{1}{2}} x$$

$$\tilde{A} = A + I_N$$

$$X \in \mathbb{R}^{N \times C}, \quad C - input\ channels$$

$$\Theta \in \mathbb{R}^{C \times F}, \quad F - output\ channels$$

$$\mathrm{Z} = \widetilde{D}^{-\frac{1}{2}} \tilde{A} \widetilde{D}^{-\frac{1}{2}} X \Theta$$

Complexity O(|E|FC)

Graph based NN

f(X, A) – neural network

$$H^{l+1} = \sigma\left(\widetilde{D}^{-\frac{1}{2}}\tilde{A}\widetilde{D}^{-\frac{1}{2}}H^{(l)}W^{l}\right) \text{ - } propagation\ rule$$

$$\tilde{A} = A + I_N$$

$$H^0 = X$$

$$W^l - weight\ matrix, \qquad \sigma - activation$$

Semi-Supervised Node Classification

$$\hat{A} = \widetilde{D}^{-\frac{1}{2}}\tilde{A}\widetilde{D}^{-\frac{1}{2}}$$

$$Z = f(X, A) = \ softmax\left(\hat{A}ReLU\left(\hat{A}XW^{(0)}\right)W^{(1)}\right)$$

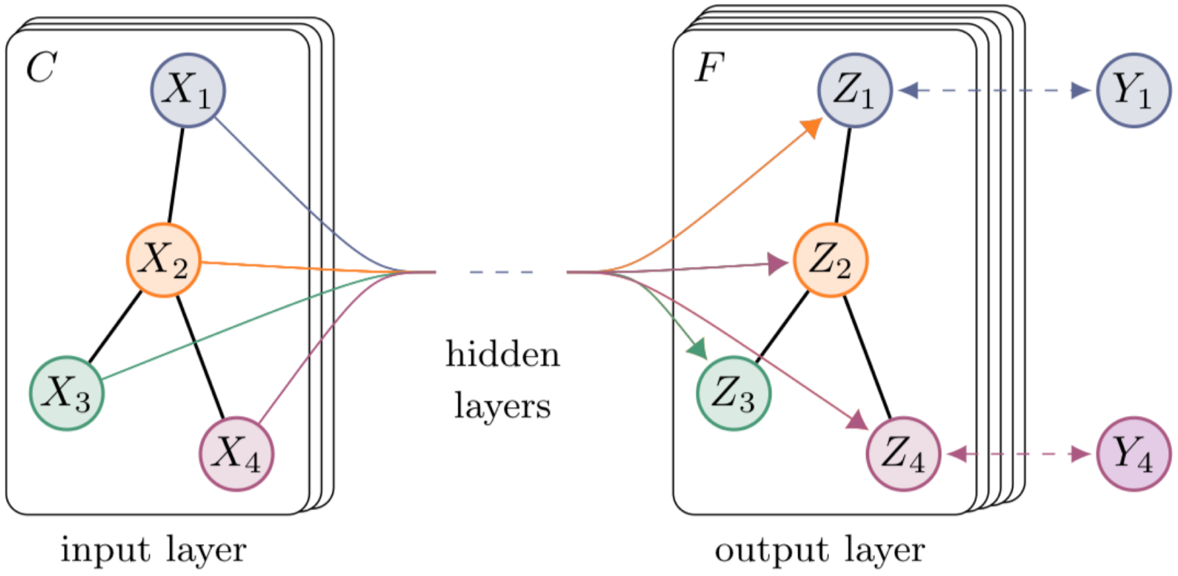
$$\mathcal{L} = -\sum_{y \in Y_L}\sum_{f=1\ldots F} y_{lf} lnZ_{lf}$$

X - full train set

Train on X

Y_L - labeled nodes

Compute loss on Y_L



Experiments
Datasets

Dataset	Type	Nodes	Edges	Classes	Features	Label rate
Citeseer	Citation network	3,327	4,732	6	3,703	0.036
Cora	Citation network	2,708	5,429	7	1,433	0.052
Pubmed	Citation network	19,717	44,338	3	500	0.003
NELL	Knowledge graph	65,755	266,144	210	5,414	0.001

Experiments
Set up

- Two layer GCN
- Metric – accuracy (1000 size of test set)
- Baselines:
    - Label propagation
    - Semi-supervised embedings
    - Manifold regularization
    - Iterative classification algorithm
    - Planetoid

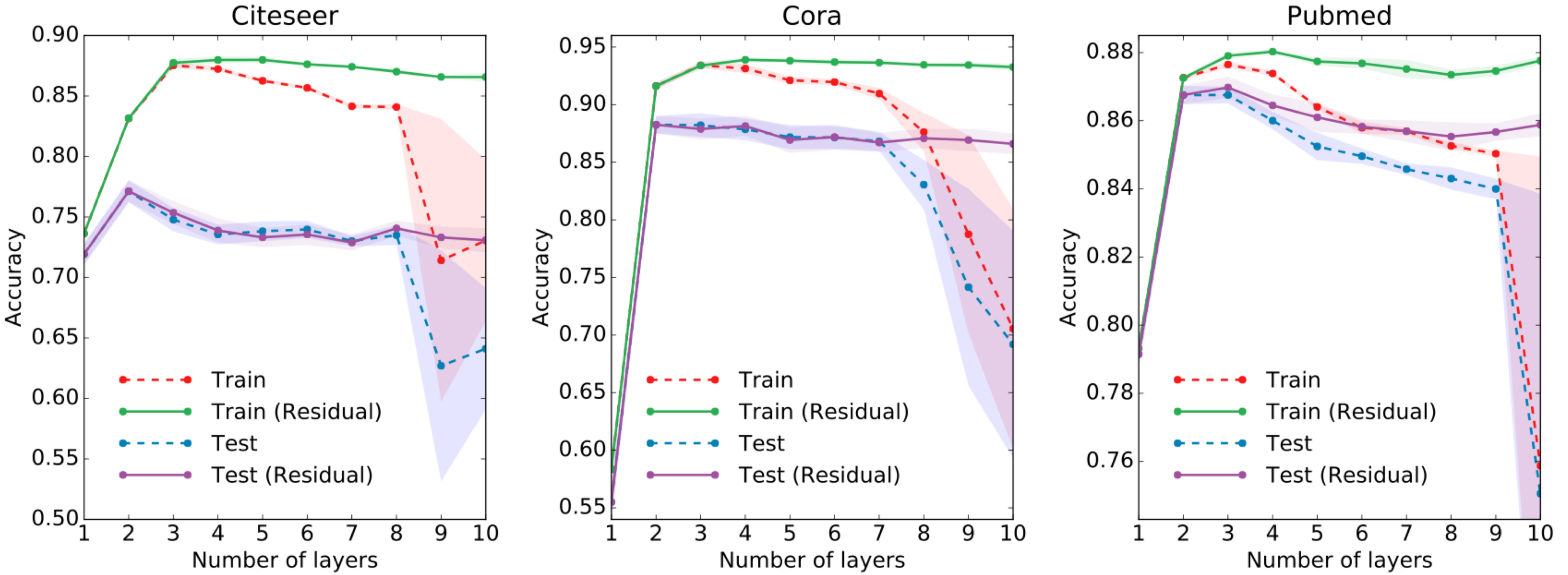
Experiments
Results

Method	**Citeseer**	**Cora**	**Pubmed**	**NELL**
ManiReg [3]	60.1	59.5	70.7	21.8
SemiEmb [28]	59.6	59.0	71.1	26.7
LP [32]	45.3	68.0	63.0	26.5
DeepWalk [22]	43.2	67.2	65.3	58.1
ICA [18]	69.1	75.1	73.9	23.1
Planetoid* [29]	64.7 (26s)	75.7 (13s)	77.2 (25s)	61.9 (185s)
GCN (this paper)	**70.3** (7s)	**81.5** (4s)	**79.0** (38s)	**66.0** (48s)
GCN (rand. splits)	67.9 ± 0.5	80.1 ± 0.5	78.9 ± 0.7	58.4 ± 1.7

Experiments
Comparison of propagation models

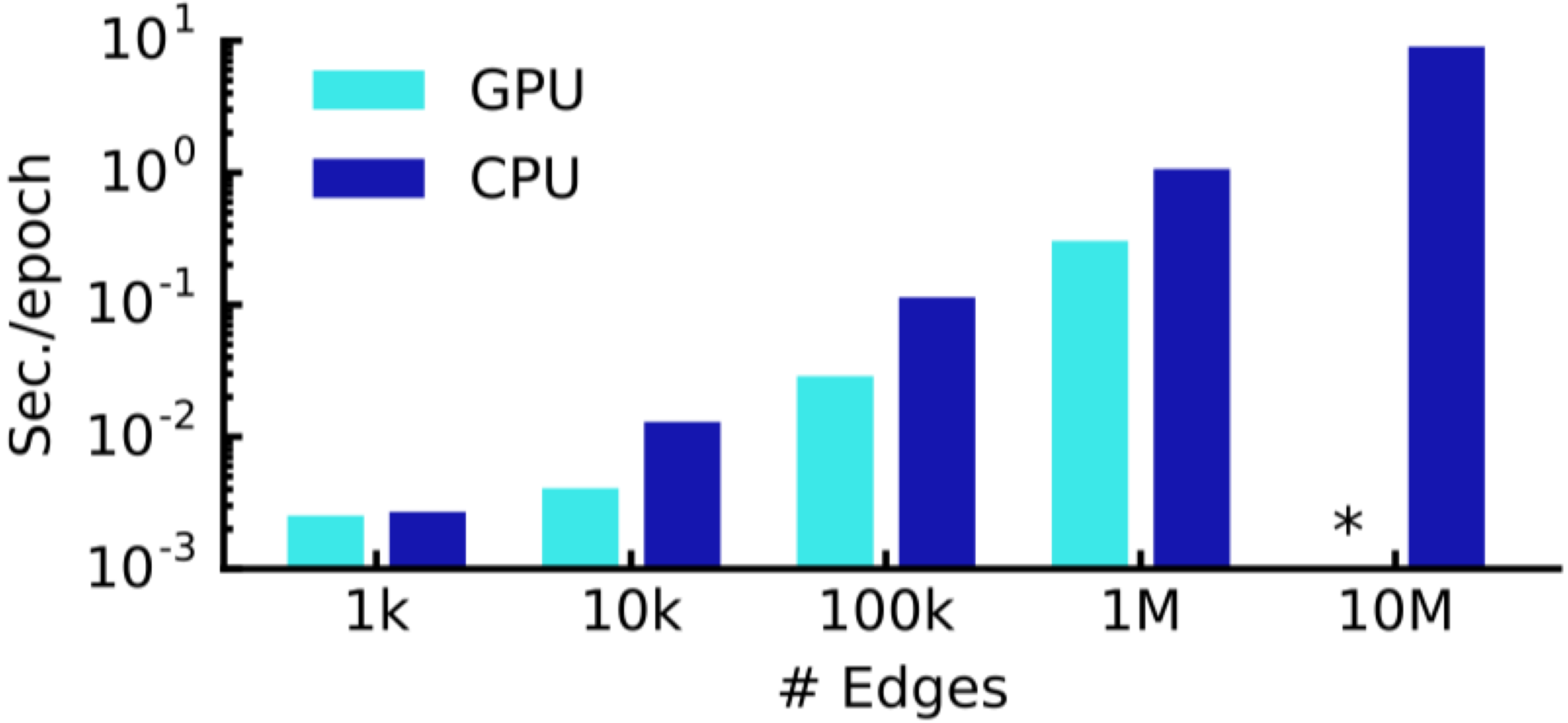
Description		Propagation model	Citeseer	Cora	Pubmed
Chebyshev filter (Eq. 5)	$K=3$	$\sum_{k=0}^{K} T_k(\tilde{L}) X \Theta_k$	69.8	79.5	74.4
	$K=2$		69.6	81.2	73.8
1st-order model (Eq. 6)		$X\Theta_0 + D^{-\frac{1}{2}} A D^{-\frac{1}{2}} X \Theta_1$	68.3	80.0	77.5
Single parameter (Eq. 7)		$(I_N + D^{-\frac{1}{2}} A D^{-\frac{1}{2}}) X \Theta$	69.3	79.2	77.4
Renormalization trick (Eq. 8)		$\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} X \Theta$	**70.3**	**81.5**	**79.0**
1st-order term only		$D^{-\frac{1}{2}} A D^{-\frac{1}{2}} X \Theta$	68.7	80.5	77.8
Multi-layer perceptron		$X\Theta$	46.5	55.1	71.4

Experiments
number of layers



Experiments

Time



Limitations

- Memory requirement
- Only undirected graph
- No trade-off parameter $\tilde{A} = A + \lambda I_N$

Conclusion

- GCN novel approach for semi-supervised on graph data
- Encode both graph structure and node features in a way useful for semi-supervised
- Outperforms other recent methods
- Computational efficient

- https://arxiv.org/pdf/1609.02907.pdf