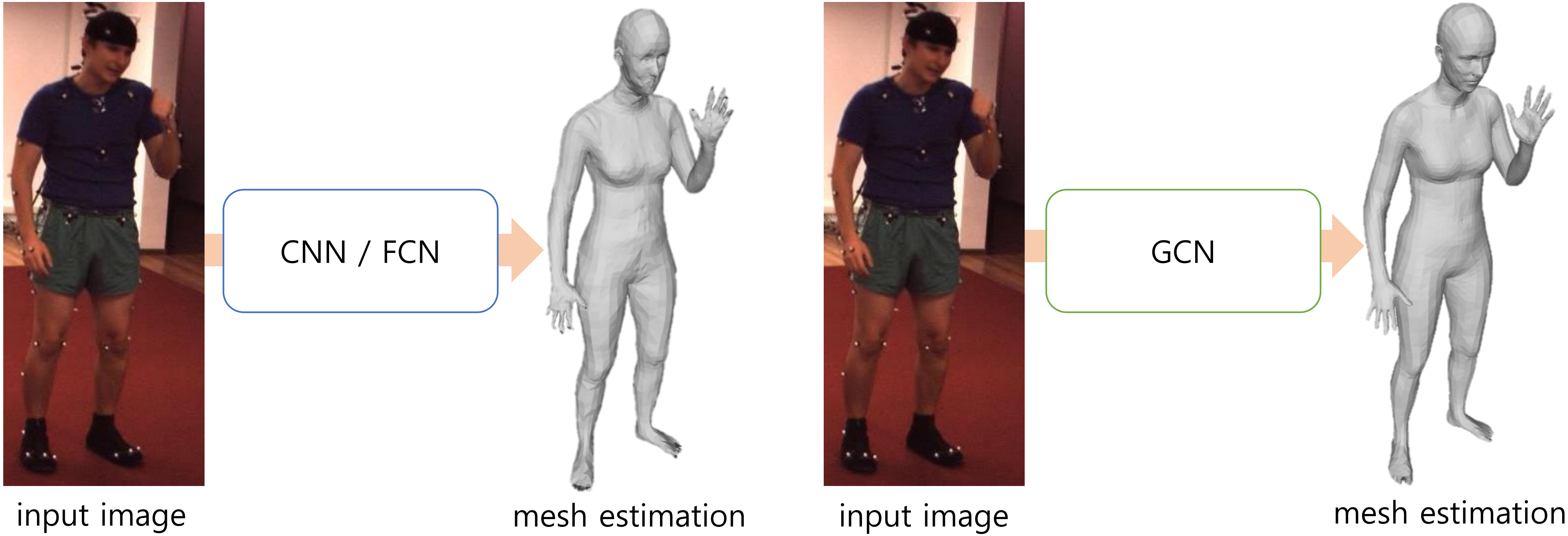
Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning

(accepted to 2018 AAAI)

Hongsuk Choi

CVLAB, Seoul National University

Why I picked this paper



Needs Laplacian smoothing term!

No need for Laplacian smoothing term!

This paper explains…

- Why GCN works
- When GCN fails
- How to solve the failure cases

Why GCN works

- The propagation rule of each layer
  - FCN layer: $H^{(l+1)} = \sigma\left(H^{(l)}\Theta^{(l)}\right)$
  - GCN layer: $H^{(l+1)} = \sigma\left(\underline{\widetilde{D}^{-\frac{1}{2}}\tilde{A}\widetilde{D}}^{-\frac{1}{2}}H^{(l)}\Theta^{(l)}\right)$
- Laplacian smoothing[1] on on each channel of the input features:

$$Y = \left(I - \gamma\widetilde{D}^{-1}\widetilde{L}\right)X$$

set $\gamma = 1$

$$Y = \widetilde{D}^{-1}\tilde{A}X$$

Replace the normalized Laplacian with the symmetrically normalized Laplacian

$$Y = \underline{\widetilde{D}^{-\frac{1}{2}}\tilde{A}\widetilde{D}}^{-\frac{1}{2}}X$$

[1] Taubin, G. 1995. A signal processing approach to fair surface design. In Proceedings of the 22nd annual conference on Computer graphics and interactive techniques, 351–358. ACM

Why GCN works

- The graph convolution is a special form of Laplacian smoothing

- The smoothing makes the classification task easier by making nodes in the same cluster have similar features

Table 1: GCNs vs. Fully-connected networks

One-layer FCN	Two-layer FCN	One-layer GCN	Two-layer GCN
0.530860	0.559260	**0.707940**	**0.798361**

When GCN fails

- Too many layers
  - mathematically proved how over-smoothing harms learning

Theorem 1. *If a graph has no bipartite components, then for any* $\mathbf{w} \in \mathbb{R}^n$, *and* $\alpha \in (0,1]$,

$$\lim_{m \to +\infty} (I - \alpha L_{rw})^m \mathbf{w} = [\mathbf{1}^{(1)}, \mathbf{1}^{(2)}, \ldots, \mathbf{1}^{(k)}]\theta_1,$$

$$\lim_{m \to +\infty} (I - \alpha L_{sym})^m \mathbf{w} = D^{-\frac{1}{2}}[\mathbf{1}^{(1)}, \mathbf{1}^{(2)}, \ldots, \mathbf{1}^{(k)}]\theta_2,$$

where $\theta_1 \in \mathbb{R}^k, \theta_2 \in \mathbb{R}^k$, *i.e., they converge to a linear combination of* $\{\mathbf{1}^{(i)}\}_{i=1}^k$ *and* $\{D^{-\frac{1}{2}}\mathbf{1}^{(i)}\}_{i=1}^k$ *respectively.*

- Too few labels
  - a shallow GCN cannot sufficiently propagate the label information to the entire graph

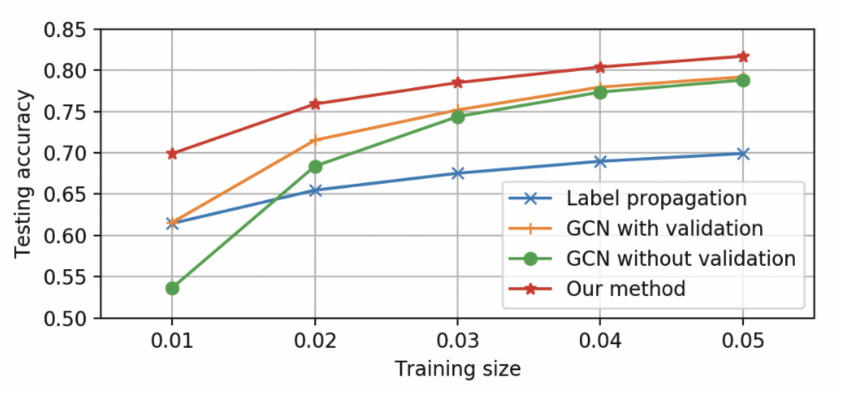


Figure 1: Performance comparison of GCNs, label propagation, and our method for semi-supervised classification on the Cora citation network.

When GCN fails

- Too many layers
  - mathematically proved how over-smoothing harms learning
- Too few labels
  - a shallow GCN cannot sufficiently propagate the label information to the entire graph

➔get lower bound of # of labels by solving:

$$(\hat{d})^{\tau} * \eta \approx n$$

, where $\hat{d}$ is the average degree of the graph, τ is the number of layers, η is the lower bound, and n is the number of nodes in the graph

Solutions: expand the training set!

- Co-Train a GCN with a Random Walk Model

Algorithm 1 Expand the Label Set via ParWalks

$P := (L + \alpha\Lambda)^{-1}$
for each class k **do**
    $\boldsymbol{p} := \sum_{j \in \mathcal{S}_k} P_{:,j}$
    Find the top t vertices in $\boldsymbol{p}$
    Add them to the training set with label k
end for

- GCN Self-Training

Algorithm 2 Expand the Label Set via Self-Training

$\boldsymbol{Z} := GCN(X) \in \mathbb{R}^{n \times F}$, the output of GCN
for each class k **do**
    Find the top t vertices in $Z_{i,k}$
    Add them to the training set with label k
end for

Experiments

Table 3: Classification Accuracy On Cora

	Cora					
Label Rate	0.5%	1%	2%	3%	4%	5%
LP	56.4	62.3	65.4	67.5	69.0	70.2
Cheby	38.0	52.0	62.4	70.8	74.1	77.6
GCN-V	42.6	56.9	67.8	74.9	77.6	79.3
GCN+V	50.9	62.3	72.2	76.5	78.4	79.7
Co-training	56.6	66.4	73.5	75.9	78.9	80.8
Self-training	53.7	66.1	73.8	77.2	79.4	80.0
Union	**58.5**	**69.9**	**75.9**	**78.5**	**80.4**	**81.7**
Intersection	49.7	65.0	72.9	77.1	79.4	80.2

Table 6: Accuracy under 20 Labels per Class

Method	**CiteSeer**	**Cora**	**Pubmed**
ManiReg	60.1	59.5	70.7
SemiEmb	59.6	59.0	71.7
LP	45.3	68.0	63.0
DeepWalk	43.2	67.2	65.3
ICA	69.1	75.1	73.9
Planetoid	64.7	75.7	77.2
GCN-V	68.1	80.0	78.2
GCN+V	68.9	80.3	**79.1**
Co-training	64.0	79.6	77.1
Self-training	67.8	80.2	76.9
Union	65.7	**80.5**	78.3
Intersection	**69.9**	79.8	77.0

Thank you