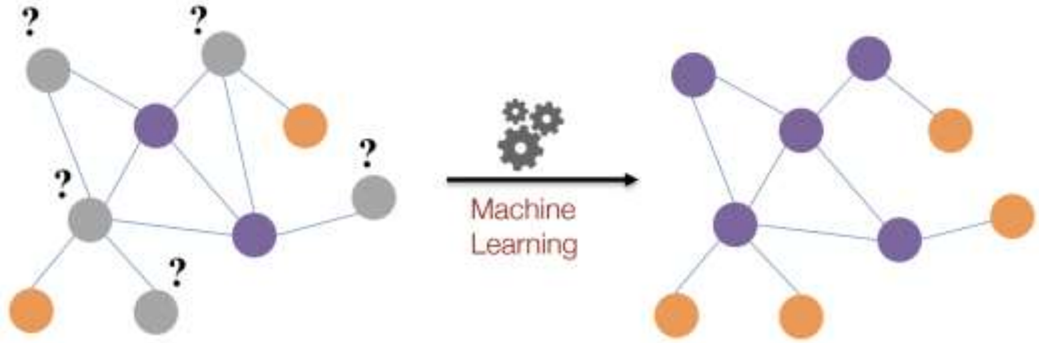
On the Equivalence of Decoupled GCN and Label Propagation

Hande Dong, Jiawei Chen, Fuli Feng, Xiangnan He, Shuxian Bi, Zhaolin Ding, Peng Cui

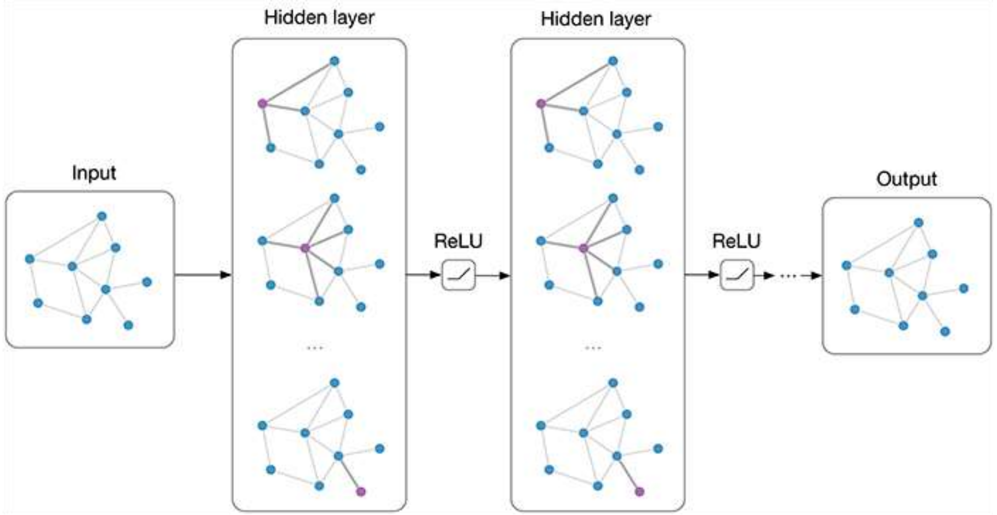
WWW2021

Background

Semi-supervised node classification task:



Graph convolution network (GCN):



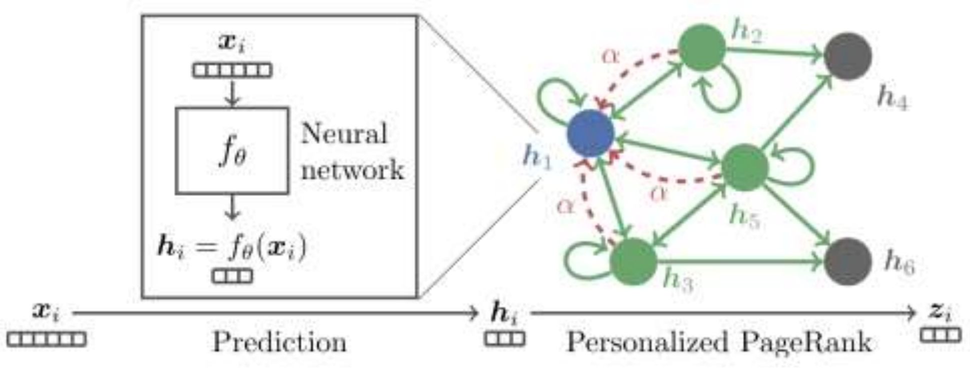
$$H^{(k+1)} = \sigma\left(\hat{A}H^{(k)}W^{(k)}\right)$$

Aggregation: $P^{(k)} = \hat{A}H^{(k)}$

coupling

Transformation: $\sigma(P^{(k)}W^{(k)})$

DECOUPLED GCN- APPNP



DECOUPLED GCN!

APPNP decouples neighborhood aggregation and feature transformation:

- High efficiency
- SOTA performance

APPNP (ICLR 2019), DAGNN (KDD 2020)
SGCN (ICML 2019), lightGCN (SIGIR 2020)

DECOUPLED GCN

The general architecture of decoupled GCN:

$$\hat{Y} = \text{softmax}\left(\bar{A} f_\theta(X)\right) \qquad (1)$$

APPNP:

$$H^{(0)} = f_\theta(X)$$
$$H^{(k)} = (1-\alpha)\hat{A}H^{(k-1)} + \alpha H^{(0)}, \quad k = 1, 2, ...K-1 \longrightarrow \bar{A} = (1-\alpha)^K \hat{A}^K + \alpha \sum_{k=0}^{K-1} (1-\alpha)^k \hat{A}^k \qquad (2)$$
$$\hat{Y}_{APPNP} = \text{softmax}\left(H^{(K)}\right).$$

SGCN: $\hat{Y}_{\text{SGCN}} = \text{softmax}\left(S^K X\Theta\right) \longrightarrow f_\theta(X) = X\Theta \quad \bar{A} = S^K$ (3)

DAGNN: $\longrightarrow \bar{A} = \sum_{k=0}^{K} s_k \hat{A}^k$ (4)

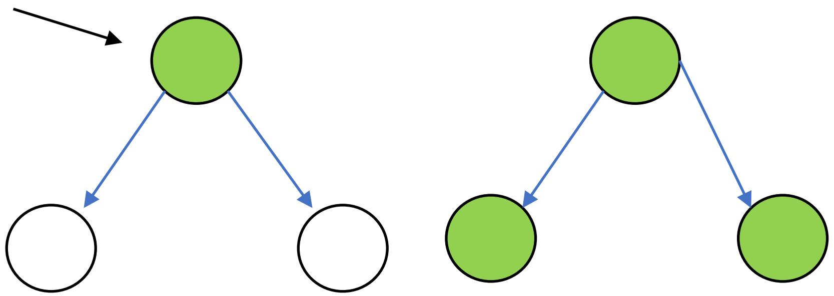
Propagation then Training

LPA:

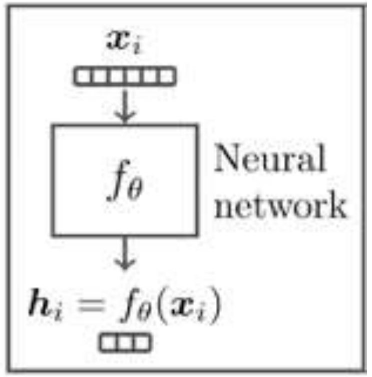
$$Y_{soft} = \hat{Y} = \bar{A}Y. \tag{5}$$

PT: The objective function

$$L(\theta) = \ell(f_\theta(X), \bar{A}Y), \tag{6}$$



Step 1: label propagation ~ pseudo-label



Step 2: Training

PT and DECOUPLED GCN

PT: The objective function

$$L_{PT} = \ell(f_\theta(X), \bar{A}Y) \qquad (7)$$

Decoupled GCN : The objective function

$$L_{DGCN} = \ell(\bar{A}f_\theta(X), Y) \qquad (8)$$

PT

PT: The objective function

$$L_{PT} = \ell(f_\theta(X), \bar{A}Y) \qquad (9)$$

$$\begin{aligned} L(\theta) &= -\sum_{i\in\mathcal{V},k\in C}\left(\sum_{j\in\mathcal{V}_l}\bar{a}_{ij}y_{jk}\right)\log f_{ik} \\ &= -\sum_{i\in\mathcal{V},j\in\mathcal{V}_l}\bar{a}_{ij}\sum_{k\in C}y_{jk}\log f_{ik} \\ &= \sum_{i\in\mathcal{V},j\in\mathcal{V}_l}\bar{a}_{ij}\,\mathrm{CE}\left(f_i,\boldsymbol{y}_j\right), \end{aligned} \qquad (10)$$

$$L_{PT} = L(\theta) = \sum_{i\in\mathcal{V},j\in\mathcal{V}_l} w_{ij}\,\mathrm{CE}\left(f_i,\boldsymbol{y}_j\right), \qquad (11)$$

$$\nabla_\theta L_{PT} = \sum_{j\in\mathcal{V}_l,i\in\mathcal{V}} w_{ij}\nabla_\theta\,\mathrm{CE}\left(f_i,\boldsymbol{y}_j\right) \qquad (12)$$

DECOUPLED GCN

Decoupled GCN : The objective function

$$L_{DGCN} = \ell(\bar{A}f_\theta(X), Y) \quad (13)$$

$$\begin{aligned} L_{DGCN} &= \ell(\bar{A}f_\theta(X), Y) \\ &= -\sum_{j\in\mathcal{V}_l, k\in C} y_{jk}\left(\log \sum_{i\in\mathcal{V}} \bar{a}_{ji} f_{ik}\right). \end{aligned} \quad (14)$$

$$\begin{aligned} \nabla_\theta L_{DGCN} &= -\sum_{j\in\mathcal{V}_l, k\in C} y_{jk} \nabla_\theta \left(\log \sum_{i\in\mathcal{V}} \bar{a}_{ji} f_{ik}\right) \\ &= -\sum_{j\in\mathcal{V}_l, k\in C} y_{jk} \frac{\sum_{i\in\mathcal{V}} \bar{a}_{ji} \nabla_\theta f_{ik}}{\sum_{q\in\mathcal{V}} \bar{a}_{jq} f_{qk}}. \end{aligned} \quad (15)$$

Gradient analysis:

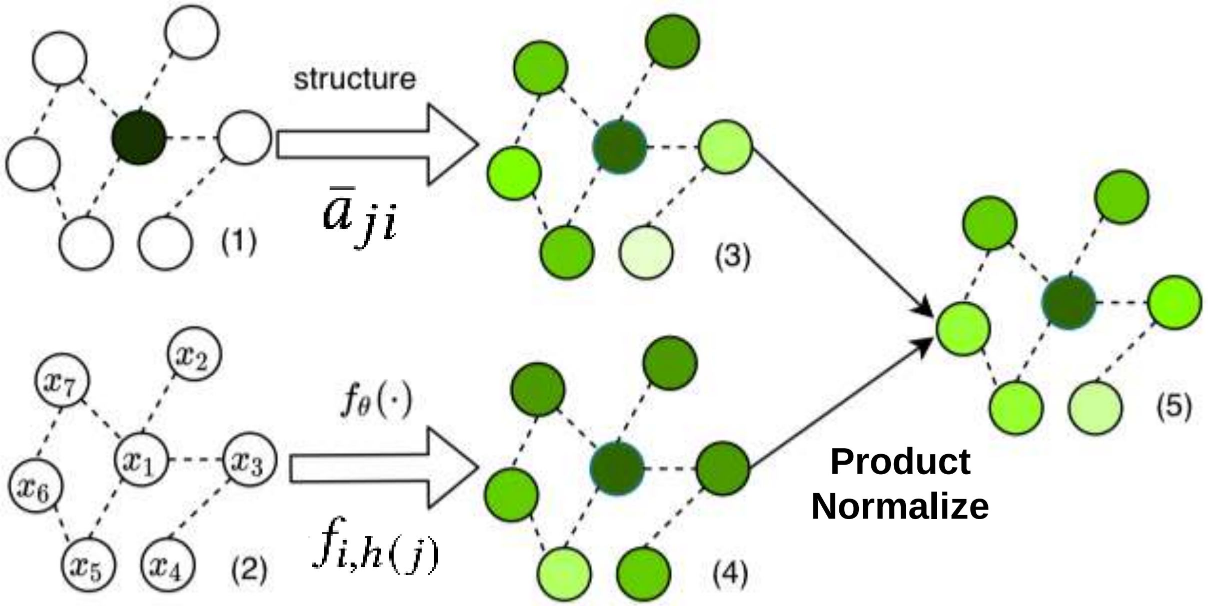
$$\begin{aligned} \nabla_\theta L_{DGCN} &= -\sum_{j\in\mathcal{V}_l} y_{j,h(j)} \frac{\sum_{i\in\mathcal{V}} \bar{a}_{ji} \nabla_\theta f_{i,h(j)}}{\sum_{q\in\mathcal{V}} \bar{a}_{jq} f_{q,h(j)}} \\ &= -\sum_{i\in\mathcal{V}, j\in\mathcal{V}_l} \frac{\bar{a}_{ji} f_{i,h(j)}}{\sum_{q\in\mathcal{V}} \bar{a}_{jq} f_{q,h(j)}} y_{j,h(j)} \frac{\nabla_\theta f_{i,h(j)}}{f_{i,h(j)}} \\ &= \sum_{i\in\mathcal{V}, j\in\mathcal{V}_l} \frac{\bar{a}_{ji} f_{i,h(j)}}{\sum_{q\in\mathcal{V}} \bar{a}_{jq} f_{q,h(j)}} \nabla_\theta \operatorname{CE}(f_i, y_j). \end{aligned} \quad (16)$$

$$\nabla_\theta L_{PT} = \sum_{j\in\mathcal{V}_l, i\in\mathcal{V}} w_{ij} \nabla_\theta \operatorname{CE}(f_i, y_j) \quad (17)$$

DECOUPLED GCN

Weight of decoupled GCN:

$$w_{ij} = \frac{\bar{a}_{ji} f_{i,h(j)}}{\sum_{q \in \mathcal{V}} \bar{a}_{jq} f_{q,h(j)}}$$



Weaknesses:

- **Unstable to initialization** ~ $f_{i,h(j)}$ is reliable?
- **Sensitive to label noise** ~ equal weight?

$$\sum_{i \in \mathcal{V}} w_{ij} = 1$$

The accumulated weight of the pseudo-labeled data generated from a specific labeled source node **equals 1.**

PTA: How to improve d-GCN

Weight of decoupled GCN: $$w_{ij} = \frac{\bar{a}_{ji} f_{i,h(j)}}{\sum_{q \in \mathcal{V}} \bar{a}_{jq} f_{q,h(j)}}$$

Design 1 - **remove normalization**:

- Different label weight $$w_{ij} = \bar{a}_{ji} f_{i,h(j)}$$
- Robust to label noise

Design 2 - **add adaptive factor**: $$w_{ij} = \bar{a}_{ji} f_{i,h(j)}^{\gamma}$$

- Factor: $\gamma = \log(1 + e/\epsilon)$
- The model becomes more and more reliable, $\gamma \uparrow$
- stable to initialization

In the **early stage of training**, when the neural network generates **relatively unreliable prediction**, PTA reduces the impact of model prediction on weighting pseudo-labeled data.

PTA:

$$\frac{\bar{a}_{ji} f_{i,h(j)}}{\sum\limits_{q \in \mathcal{V}} \bar{a}_{jq} f_{q,h(j)}} \xrightarrow{\text{remove normalization}} \bar{a}_{ji} f_{i,h(j)} \xrightarrow{\text{adaptive factor}} \bar{a}_{ji} f_{i,h(j)}^{\gamma}$$

- Element form of loss function:

$$L_{PTA}(\theta) = \sum_{i \in V, j \in V_l} w_{ij} \, \mathrm{CE}\left(\boldsymbol{f}_i, \boldsymbol{y}_j\right), \qquad w_{ij} = \bar{a}_{ji} f_{i,h(j)}^{\gamma}.$$

- **Matrix form** of loss function:

$$L_{PTA}(\theta) = -SUM\left(Y_{soft} \otimes \boldsymbol{f}(\boldsymbol{X})^{\gamma} \otimes log\left(\boldsymbol{f}_{\theta}(\boldsymbol{X})\right)\right),$$

f(x) does not propagate gradients backword

- No neighborhood aggregation. **Much faster!**
  - ➢ Decoupled GCN aggregates neighborhood
  - ➢ Neighborhood aggregation is hard to parallelly compute

Advantages: stable, robust to label noise.

Experiments

Table 2: Statistics of the datasets.

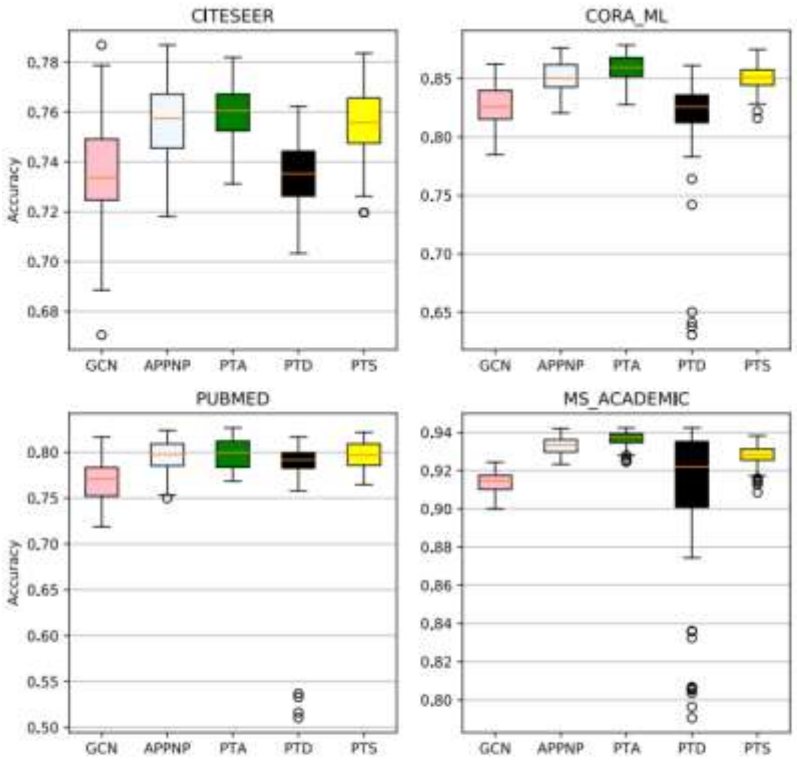
Dataset	Nodes	Edges	Features	Classes
CITESEER	2,110	3,668	3,703	6
CORA_ML	2,810	7,981	2,879	7
PUBMED	19,717	44,324	500	3
MS_ACADEMIC	18,333	81,894	6,805	15

Method	CITESEER	CORA_ML	PUBMED	MS_ACA
MLP	63.98 ± 0.44	68.42 ± 0.34	69.47 ± 0.47	89.69 ± 0.10
GCN	73.62 ± 0.39	82.70 ± 0.39	76.84 ± 0.44	91.39 ± 0.10
SGCN	75.57 ± 0.28	75.97 ± 0.72	71.24 ± 0.86	91.03 ± 0.16
APPNP #	75.73 ± 0.30	85.09 ± 0.25	79.73 ± 0.31	93.27 ± 0.08
DAGNN	74.53 ± 0.38	85.75 ± 0.23	79.59 ± 0.37	92.29 ± 0.07
APPNP	75.48 ± 0.29	85.07 ± 0.25	79.61 ± 0.33	93.31 ± 0.08
PTA	$\mathbf{75.98 \pm 0.24}$	$\mathbf{85.90 \pm 0.21}$	$\mathbf{79.89 \pm 0.31}$	$\mathbf{93.64 \pm 0.08}$
p-value	5.56×10^{-4}	1.81×10^{-9}	1.09×10^{-2}	1.57×10^{-8}

PTA is better than APPNP with t-test.

Experiments

- Box plot of accuracy of different models:



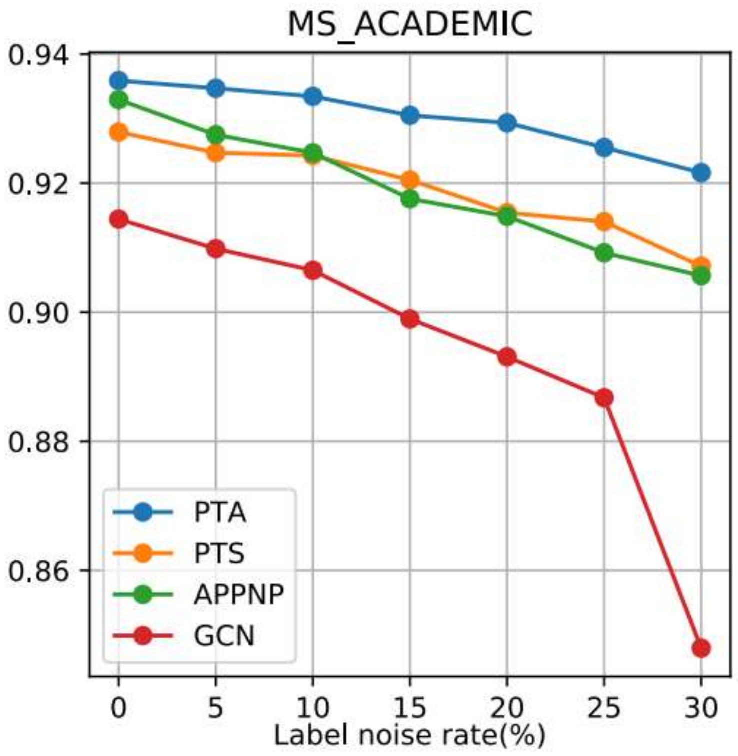
PTA>APPNP, PTD: adaptive strategy in PTA is useful

$$L_{PTA}(\theta) = -SUM\left(Y_{soft} \otimes f(X)^{\gamma} \otimes log\left(f_{\theta}(X)\right)\right),$$

```
if self.mode == 2:
    loss = - torch.sum(torch.mul(torch.log_softmax(y_hat, dim=-1), torch.mul(y_soft, y_hat_con**exp))) / self.number_class  # PTA
elif self.mode == 1:
    loss = - torch.sum(torch.mul(torch.log_softmax(y_hat, dim=-1), torch.mul(y_soft, y_hat_con))) / self.number_class  # PTD
else:
    loss = - torch.sum(torch.mul(torch.log_softmax(y_hat, dim=-1), y_soft)) / self.number_class # PTS
```

Experiments

- Simulate label noise
  - Add noise to label in the training set



The margins of PTA and APPNP become larger with noise increasing: PTA is more robust to label noise.

PTA>others for all different label noise rate.

Experiments

- Compare the efficient of PTA and APPNP

two models. Note that estimating performance of PTA on the early-stopping set is time-consuming, we further design a fast mode of PTA (PTA(F)), which directly $\boxed{\text{use } f_\theta(x)}$ instead of ensemble results for early-stopping estimation. The performance of PTA(F) comparing with PTA and APPNP is presented in Table 8. From the tables, We can conclude that PTA is much faster than APPNP: on average,

$$L_{PTA}(\theta) = -SUM\left(Y_{soft} \otimes f(X)^{\gamma} \otimes log\left(f_\theta(X)\right)\right),$$

Table 6: The training time per epoch of PTA and APPNP.

Method	CITESEER	CORA_ML	PUBMED	MS_ACA
APPNP	34.73ms	28.60ms	34.98ms	30.51ms
PTA	3.33*ms*	3.35*ms*	3.27*ms*	3.33*ms*

10-times faster

Table 7: The total training time of PTA and APPNP.

Method	CITESEER	CORA_ML	PUBMED	MS_ACA
APPNP	52.75s	75.30s	49.39s	134.23s
PTA	10.14*s*	11.95*s*	10.59*s*	17.12s
PTA(F)	1.19*s*	1.25*s*	1.40*s*	3.92*s*

PTA: 5 times faster
PTA(F): 50 times faster

Table 8: The accuracy of PTA(F).

Method	CITESEER	CORA_ML	PUBMED	MS_ACA
APPNP	75.48 ± 0.29	85.07 ± 0.25	79.61 ± 0.33	93.31 ± 0.08
PTA(F)	75.51 ± 0.24	85.73 ± 0.22	79.45 ± 0.40	93.62 ± 0.08
PTA	75.98 ± 0.24	85.90 ± 0.21	79.89 ± 0.31	93.64 ± 0.08