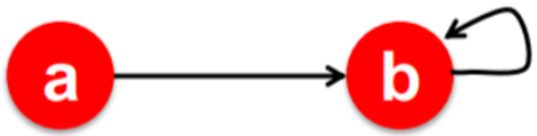
Summary Questions of the lecture

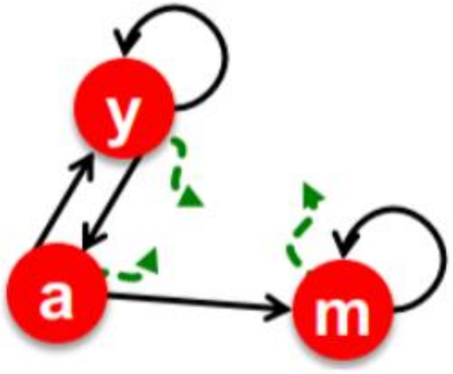
- Why are dead-ends and spider traps problems and why do teleports solve these problems?

$\rightarrow$ Spider traps absorb importance values into a region, yielding undesirable rank scores. By allowing random teleports that jump to a randomly selected page in a probability $(1-\beta)$, we can escape spider traps. Dead-ends leak importance values and break the random walk assumption of 'column stochastic'. They can be handled either by ordinary random teleports along with self-loop at a dead-end or by always teleporting when we meet a dead-end.



$$\begin{array}{ll} r_a & \rightarrow \\ r_b & \end{array} \quad \begin{array}{c|c|c|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{array} \quad \cdots\cdots$$

$$\begin{array}{ll} r_a & \rightarrow \\ r_b & \end{array} \quad \begin{array}{c|c|c|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{array} \quad \cdots\cdots$$



Summary Questions of the lecture

- Describe the PageRank algorithm with teleports based on sparse matrix formulation.

→ Storing the full transition matrix is memory-inefficient. Since M is a sparse matrix with non-zero entries only on elements corresponding to links, we may evaluate r_j on each time step using the sum of importances only for incoming links, i.e., $\beta \sum_{i \to j} r_i / d_i$, instead of one large matrix-vector multiplication. Then, the random teleports are implemented by simply adding $(1-\beta)/N$ to r_j.

$$\boldsymbol{r}^{new} = \boldsymbol{A}\boldsymbol{r}^{old}$$

$$\boxed{\boldsymbol{A} = \beta \boldsymbol{M} + (1-\beta)\left[\frac{1}{N}\right]_{N\times N}}$$

$$\boldsymbol{r} = \beta \boldsymbol{M}\boldsymbol{r} + [(1-\beta)/N]_N \underbrace{\sum_i r_i}_{\boxed{=1}} \leftarrow$$

Summary Questions of the lecture

- Describe the random walks with personalized teleports.

→ Given a set of query nodes, we would like to find out which nodes in the entire graph are more related to those in the query set, and which are less so. We perform the same random walk with teleports on the graph, but this time we teleport only to the nodes in the query set. Finally, the number of visits to each node represents its relevance score.

$$\mathcal{T}^T = [0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1]$$

$$\boxed{\boldsymbol{r} = (1-\alpha)\boldsymbol{M}\boldsymbol{r} + \alpha\mathcal{T}}$$

$$\mathcal{T}^T = [0.1, 0, 0, 0.2, 0, 0, 0.5, 0, 0, 0.2]$$

$$\mathcal{T}^T = [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0]$$

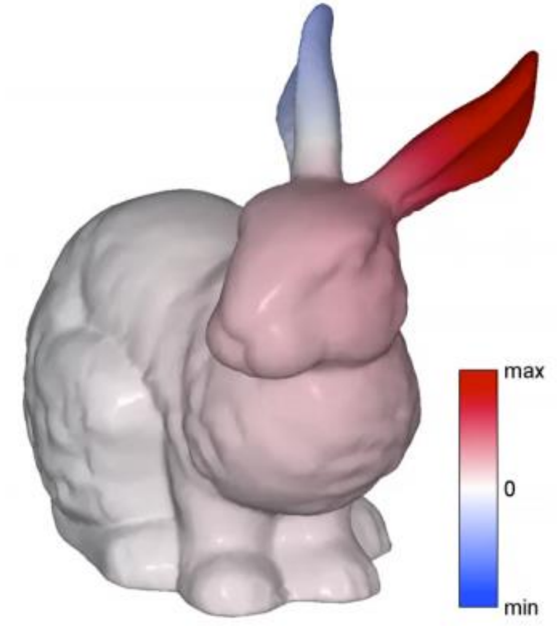
Outline of Lecture (5)

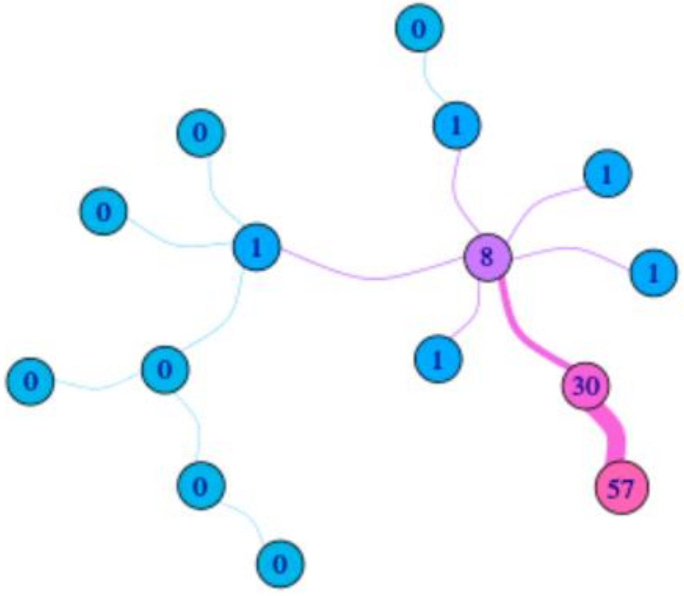
- Link Analysis
  - Directed Graph
    - Strongly Connected Graph
    - Directed Acyclic Graph
  - Link Analysis Algorithms
    - PageRank (Ranking of Nodes)
    - Random Teleports
    - Google Matrix
    - Sparse Matrix Formulation
    - Personalized PageRank
    - Random Walk with Restart

- Random Walks and Diffusion
- Diffusion in GCN
  - Propagation using graph diffusion
    - APPNP: Predict Then Propagate [ICLR'19]
    - Graph Diffusion-Embedding Networks [CVPR'19]
  - Making a new graph
    - Diffusion Improves Graph Learning [NIPS'19]
    - SSL with Graph Learning-Convolutional Networks [CVPR'19]

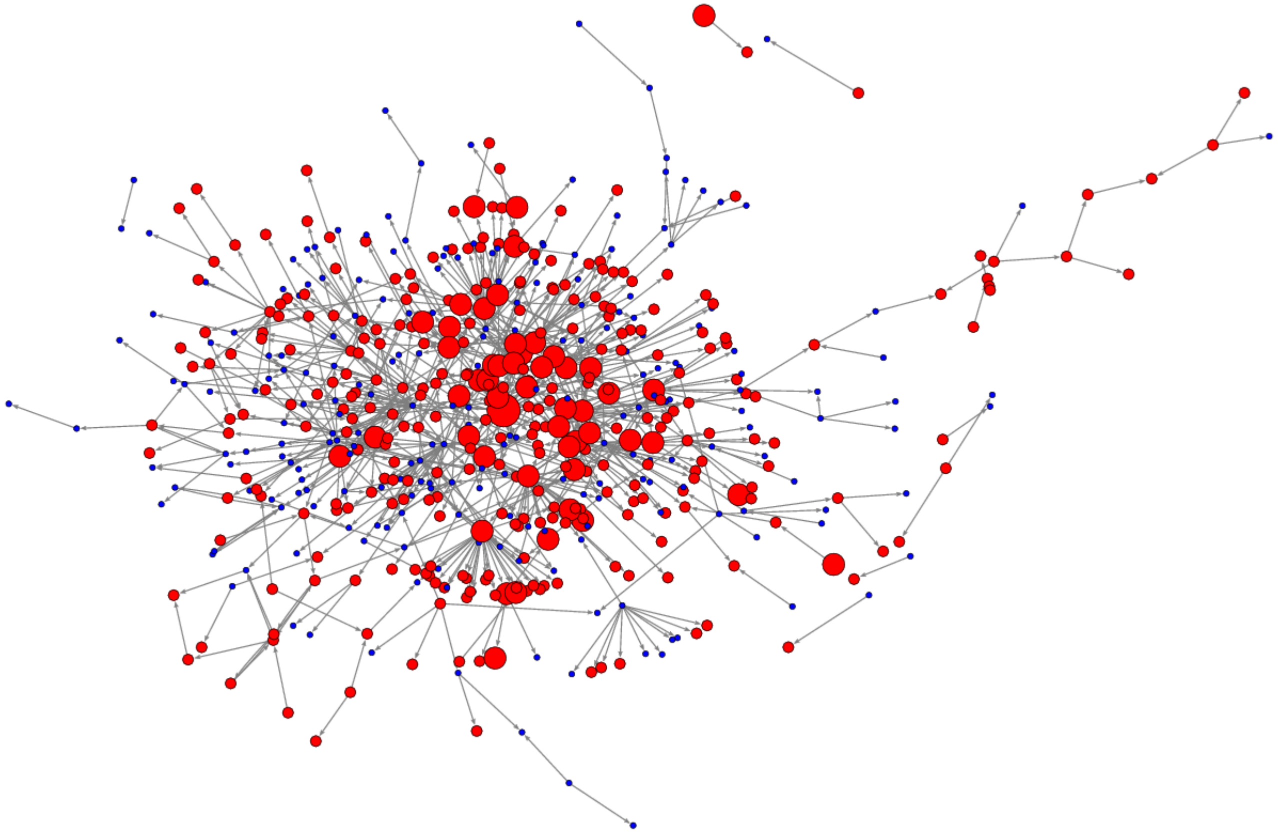
GCN:Graph Diffusion

Random Walks and Diffusion, Diffusion in GCN





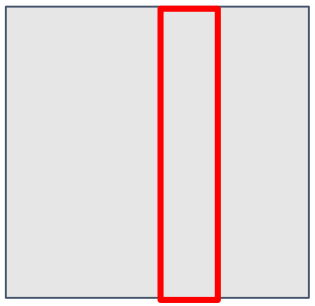
GCN: Diffusion



GCN: Random Walks (R)

- Transition matrix P (column stochastic)

P



probability that
a random walker is in i

$$\boldsymbol{r}^{(t+1)} = \boldsymbol{P}\boldsymbol{r}^{(t)}$$

$$r_i = \sum_j P_{ij} r_j$$

$$\sum_i r_i = 1$$

$$\sum_i P_{ij} = 1$$

transition probability from
j to i

- Random Walk: $P_{ij} = {W_{ji}}/{d_j} \rightarrow \boldsymbol{P} = \boldsymbol{W}^T\boldsymbol{D}^{-1}$

$$\boxed{P_{ij} = {W_{ij}}/{d_i} \rightarrow \boldsymbol{P} = \boldsymbol{D}^{-1}\boldsymbol{W} \rightarrow \boldsymbol{r}^{(t+1)} = \boldsymbol{r}^{(t)}\boldsymbol{P}}$$

GCN: Page Rank (R)

- PageRank Beyond the Web [Society for Industrial and Applied Mathematics (SIAM) 2015]
- Google's PageRank
- Random Walk + teleport probability
  - In the PageRank model, a random walker moves through the nodes in a graph, at each step moving to a new node by transitioning along an edge (with probability β) or by "teleporting" to a position independent of the previous location (with probability $(1 - \beta)$).
  - $\boldsymbol{r} = \beta \boldsymbol{P} \boldsymbol{r} + (1-\beta)v, v = [1/N, \ldots., 1/N]^T$
  - After converging, this distribution implies the importance of each node in light of the entire graph structure.

GCN: Personalized (or Localized) Page Rank (R)

- Goal: illuminate a region of a large graph around a query set of interest
  - $\boldsymbol{r} = \beta \boldsymbol{P}\boldsymbol{r} + (1-\beta)v, v = [0.2, 0, 0, \ldots, 0.3, \ldots, 0.9, \ldots, 0,0,0.6]^T$

    $v \rightarrow$ indicating the query node set

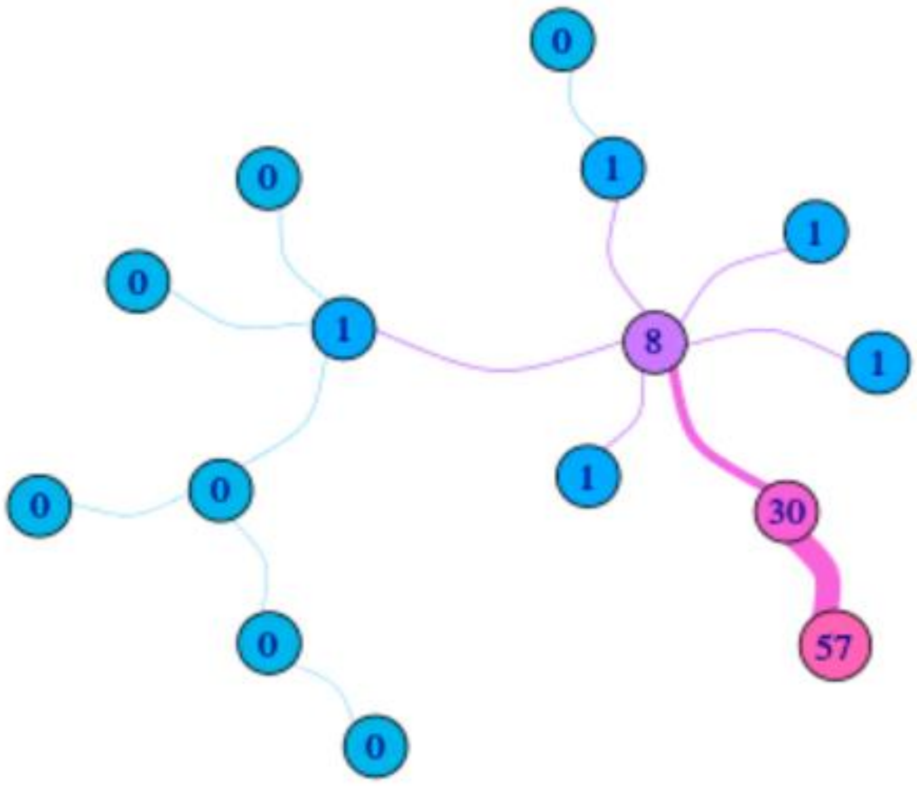
  - Random surfer in a large graph periodically teleports back to a node in the query set.

  - Then, the surfer will never move far from the query node set, but the frequency with which the surfer visits nodes reveals their proximity to the nodes in the query set.

  - Can be used to recommendation system or searching application...

GCN: Personalized (or Localized) Page Rank (R)

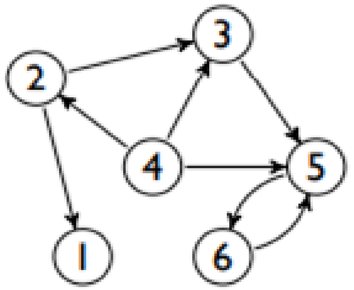
- Goal: illuminate a region of a large graph around a query set of interest
  - $\boldsymbol{r} = \beta \boldsymbol{P} \boldsymbol{r} + (1-\beta)v,$
  - $v = [0.2, 0, 0, \ldots, 0.3, \ldots, 0.9, \ldots, 0,0,0.6]^T$
  - $v \rightarrow$ indicating the query node set



GCN: Page Rank Variants

- $\boldsymbol{r} = \beta \boldsymbol{P}\boldsymbol{r} + (1-\beta)v,$

$\boxed{s = sA}$ $\boxed{s = As}$



A directed graph

$$\mathbf{A} = \begin{bmatrix} 0&0&0&0&0&0\\ 1&0&1&0&0&0\\ 0&0&0&0&1&0\\ 0&1&1&0&1&0\\ 0&0&0&0&0&1\\ 0&0&0&0&1&0 \end{bmatrix} \quad \mathbf{d} = \begin{bmatrix} 0\\2\\1\\3\\1\\1 \end{bmatrix} \quad \mathbf{c} = \begin{bmatrix} 1\\0\\0\\0\\0\\0 \end{bmatrix}$$

The adjacency matrix, degree vector, and correction vector

Random walk

$$\bar{\mathbf{P}} = \begin{bmatrix} 0&1/2&0&0&0&0\\ 0&0&0&1/3&0&0\\ 0&1/2&0&1/3&0&0\\ 0&0&0&0&0&0\\ 0&0&1&1/3&0&1\\ 0&0&0&0&1&0 \end{bmatrix}$$

$\bar{\mathbf{P}} = \mathbf{A}^T\mathbf{D}^+$

Strongly preferential

$$\mathbf{P} = \begin{bmatrix} 0&1/2&0&0&0&0\\ 0&0&0&1/3&0&0\\ 1/3&1/2&0&1/3&0&0\\ 1/3&0&0&0&0&0\\ 1/3&0&1&1/3&0&1\\ 0&0&0&0&1&0 \end{bmatrix}$$

$\mathbf{P} = \bar{\mathbf{P}} + \mathbf{v}\mathbf{c}^T$

Weakly preferential

$$\mathbf{P} = \begin{bmatrix} 1/6&1/2&0&0&0&0\\ 1/6&0&0&1/3&0&0\\ 1/6&1/2&0&1/3&0&0\\ 1/6&0&0&0&0&0\\ 1/6&0&1&1/3&0&1\\ 1/6&0&0&0&1&0 \end{bmatrix}$$

$\mathbf{P} = \bar{\mathbf{P}} + \mathbf{u}\mathbf{c}^T$
$\mathbf{u} \neq \mathbf{v}$

Reverse

$$\bar{\mathbf{P}} = \begin{bmatrix} 0&0&0&0&0&0\\ 1&0&1/2&0&0&0\\ 0&0&0&0&1/3&0\\ 0&1&1/2&0&1/3&0\\ 0&0&0&0&0&1\\ 0&0&0&0&1/3&0 \end{bmatrix}$$

$\bar{\mathbf{P}} = \mathbf{A}\,\mathrm{diag}(\mathbf{A}^T\mathbf{e})^+$

Dirichlet

$$\bar{\mathbf{P}} = \begin{bmatrix} 0&0&1/3&0&0\\ 1/2&0&1/3&0&0\\ 0&0&0&0&0\\ 0&1&1/3&0&1\\ 0&0&0&1&0 \end{bmatrix}$$

$S = \{2,3,4,5,6\}$

$\bar{\mathbf{P}} = \bar{\mathbf{P}}_{\bar{S},\bar{S}}$
$S \subset V$

Weighted

$$\bar{\mathbf{P}} = \begin{bmatrix} 0&1/4&0&0&0&0\\ 0&0&0&3/10&0&0\\ 0&3/4&0&3/10&0&0\\ 0&0&0&0&0&0\\ 0&0&1&4/10&0&1\\ 0&0&0&0&1&0 \end{bmatrix}$$

$\bar{\mathbf{P}} = (\mathbf{D}_W\mathbf{A}^T)\,\mathrm{diag}(\mathbf{A}\mathbf{D}_W\mathbf{e})^+$
$\mathbf{D}_W$ is a diagonal weighting matrix, e.g. total degree here

GCN: APPNP

- APPNP: Approximated Personalized Propagation of Neural Prediction
- Predict Then Propagate: Graph Neural Networks Meet Personalized Pager ank [ICLR'19]

$$\boldsymbol{P} = \boldsymbol{D}^{-1}\boldsymbol{A} \quad \boxed{\leftarrow \boldsymbol{Ra.Walk}}$$

$$\hat{\tilde{\boldsymbol{A}}} = \tilde{\boldsymbol{D}}^{-1/2}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}}^{-1/2} \quad \boxed{\leftarrow \boldsymbol{ChevNet}}$$

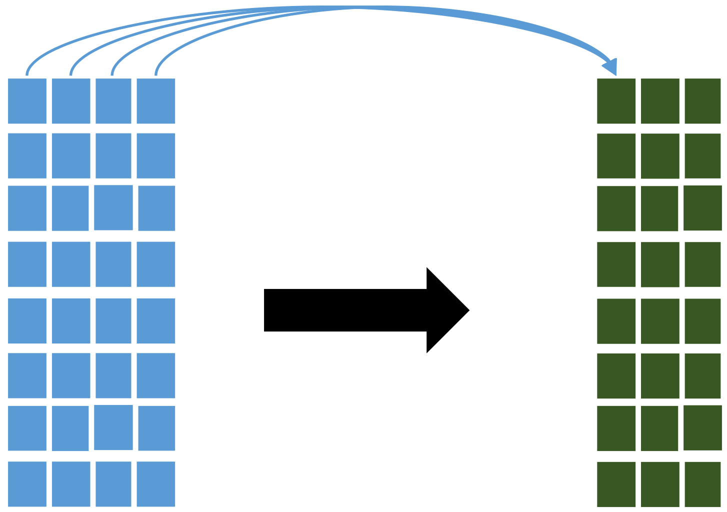
$$\boldsymbol{Z}^{(0)} = \boldsymbol{H} = f_\theta(\boldsymbol{X}),$$

$$\boldsymbol{Z}^{(k+1)} = (1-\alpha)\hat{\tilde{\boldsymbol{A}}}\boldsymbol{Z}^{(k)} + \alpha\boldsymbol{H}, k = 0, \ldots, K-2$$

$$\boxed{\boldsymbol{Z}^{(K)} = \operatorname{softmax}\left((1-\alpha)\hat{\tilde{\boldsymbol{A}}}\boldsymbol{Z}^{(K-1)} + \alpha\boldsymbol{H}\right),}$$

$K = 10$ is used

$$\boldsymbol{F}_O = \left(\widetilde{\boldsymbol{D}}^{-1/2}\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{D}}^{-1/2}\right)\boldsymbol{F}_I\boldsymbol{\Theta}$$



Simplified ChevNet

$$\boxed{\boldsymbol{r} = \beta\boldsymbol{P}\boldsymbol{r} + (1-\beta)v, v = [0.2, 0, 0, \ldots, 0.3, \ldots, 0.9, \ldots, 0,0,0.6]^T}$$

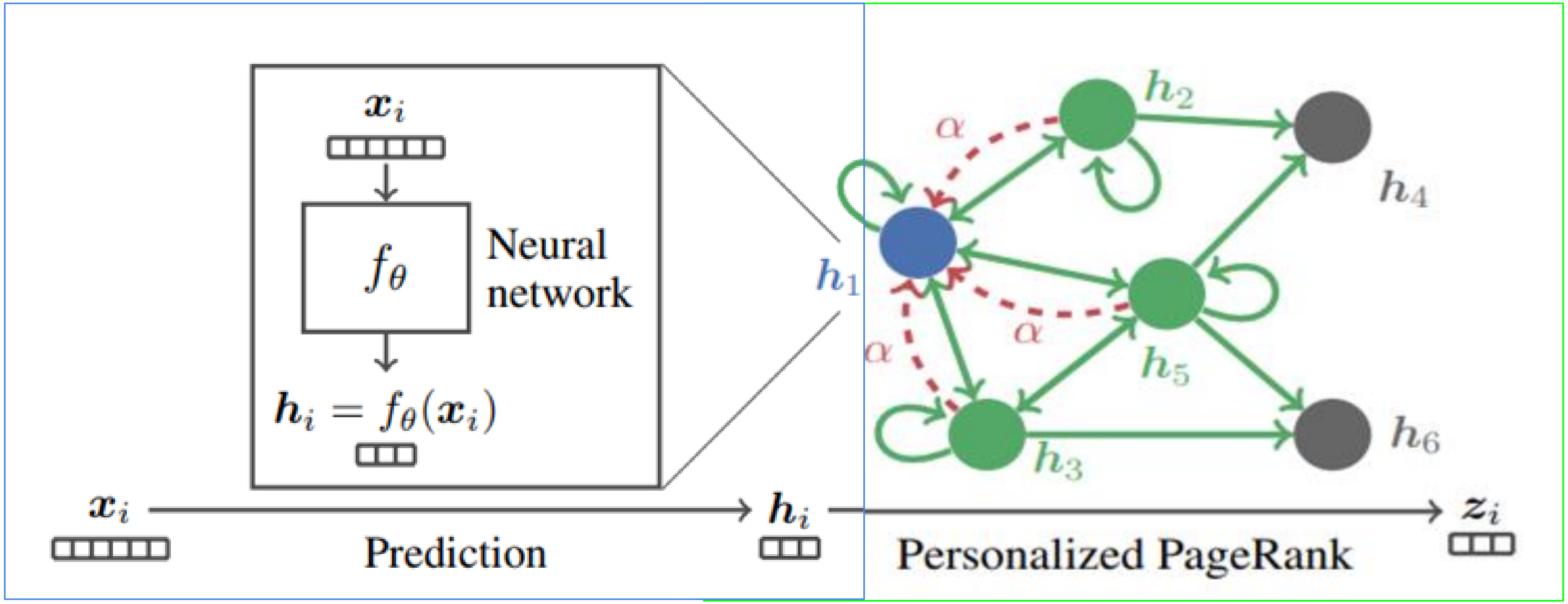
GCN: APPNP

- APPNP: Approximated Personalized Propagation of Neural Prediction

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

$$Z^{(0)} = H = f_\theta(X),$$

$$Z^{(k+1)} = (1-\alpha)\hat{\tilde{A}}Z^{(k)} + \alpha H,$$



GCN: APPNP

- APPNP: Approximated Personalized Propagation of Neural Prediction

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

$$Z^{(0)} = H = f_\theta(X),$$

$$Z^{(k+1)} = (1-\alpha)\hat{\tilde{A}}Z^{(k)} + \alpha H,$$

$$Z^{(K)} = (1-\alpha)\hat{\tilde{A}}Z^{(K-1)} + \alpha H$$

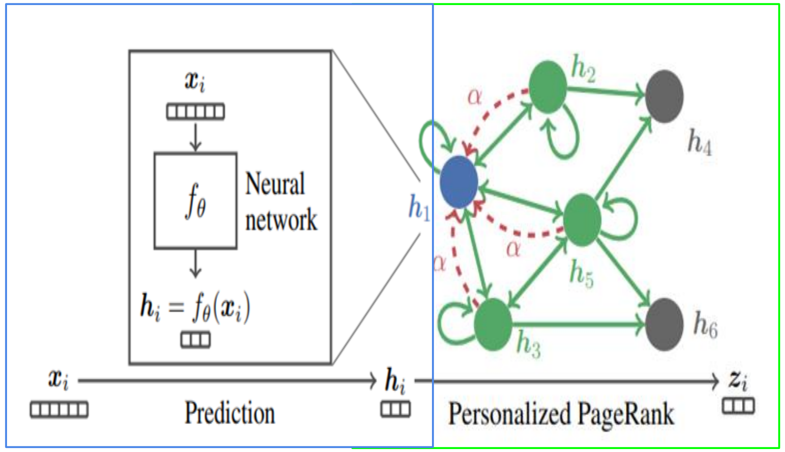
$$Z^{(K)} = (1-\alpha)\hat{\tilde{A}}\left((1-\alpha)\hat{\tilde{A}}Z^{(K-2)} + \alpha H\right) + \alpha H$$

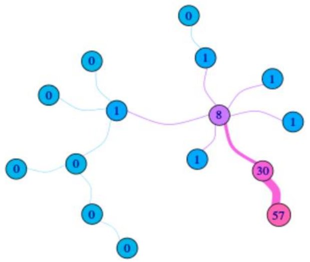
$$Z^{(K)} = (1-\alpha)^2\hat{\tilde{A}}^2 Z^{(K-2)} + (1-\alpha)\alpha\hat{\tilde{A}}H + \alpha H$$

$$Z^{(K)} = (1-\alpha)^K\hat{\tilde{A}}^K H + \cdots + (1-\alpha)\alpha\hat{\tilde{A}}H + \alpha H$$

K −hop aggregation

What is the meaning of each term? Over-smoothing?





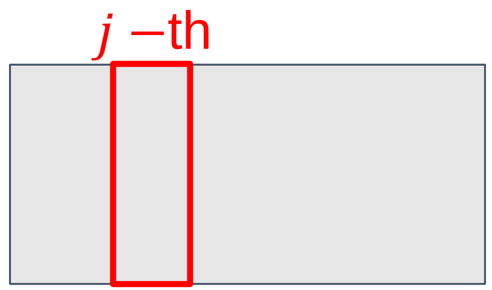
GCN: Stationary Solution of APPNP

- Scalar feature (ex, PageRank, SSL)

$$\boldsymbol{z} = (1-\alpha)\hat{\hat{A}}\boldsymbol{z} + \alpha\boldsymbol{h}$$

$$(\mathbf{I} - (1-\alpha)\hat{\hat{A}})\boldsymbol{z} = \alpha\boldsymbol{h}$$

$$\boldsymbol{z} = \alpha(\mathbf{I} - (1-\alpha)\hat{\hat{A}})^{-1}\boldsymbol{h}$$



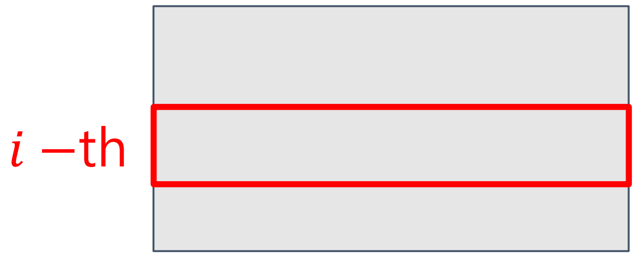
Each component is an importance score from j–th source node (sum to 1)

- Vector feature (PPNP)

$$\boldsymbol{Z} = (1-\alpha)\hat{\hat{A}}\boldsymbol{Z} + \alpha\boldsymbol{H}$$

$$\boldsymbol{Z} = \underline{\alpha(\mathbf{I} - (1-\alpha)}\hat{\hat{A}})^{-1}\boldsymbol{H}$$

$$\boldsymbol{Z}_{\boldsymbol{PPNP}} = softamx[\alpha\left(\mathbf{I} - (1-\alpha)\hat{\hat{A}}\right)^{-1}\boldsymbol{H}]$$



Each component is an importance score from each source node to i –th target node

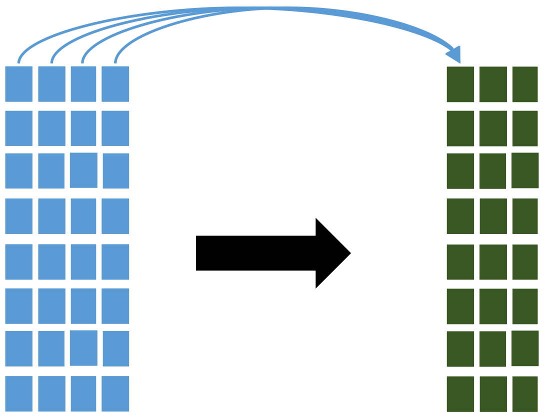
GCN: Stationary Solution of APPNP

- Vector feature (PPNP)

$$\boldsymbol{Z} = (1-\alpha)\hat{\tilde{A}}\boldsymbol{Z} + \alpha\boldsymbol{H}$$

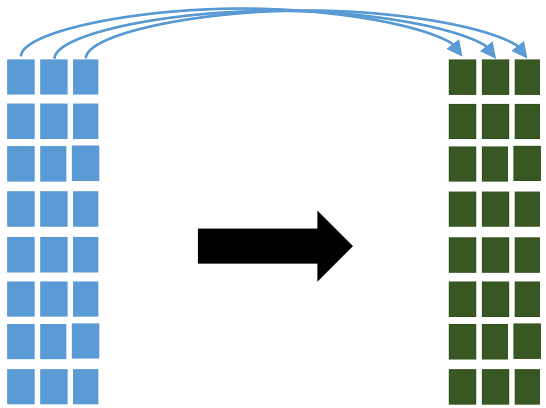
$$\boldsymbol{Z} = \alpha\underline{(\mathbf{I} - (1-\alpha)\hat{\tilde{A}})^{-1}}\boldsymbol{H}$$

$\boldsymbol{F}_O = \left(\widetilde{\boldsymbol{D}}^{-1/2}\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{D}}^{-1/2}\right)\boldsymbol{F}_I\boldsymbol{\Theta}$



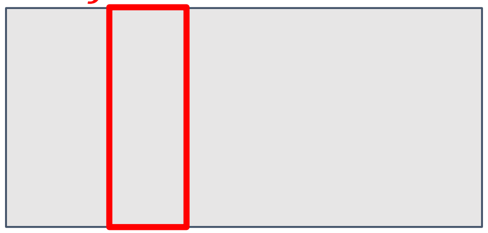
Simplified ChevNet

$\boldsymbol{Z} = (1-\alpha)\left(\widetilde{\boldsymbol{D}}^{-1/2}\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{D}}^{-1/2}\right)\boldsymbol{Z} + \alpha\boldsymbol{H}$



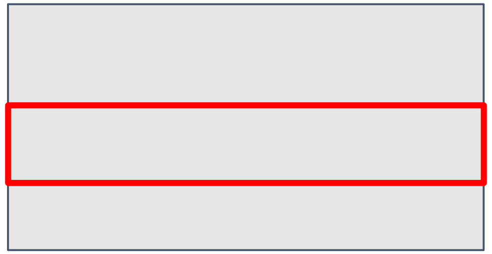
PPNP

j –th



Each component is an importance score from j–th source node (sum to 1)

i –th



Each component is an importance score from each node to i –th target node

GCN: APPNP

- Predict Then Propagate: Graph Neural Networks Meet Personalized Pagerank [ICLR'19]
- PPNP: Personalized Propagation of Neural Prediction

$$\boldsymbol{Z}_{\boldsymbol{PPNP}} = \mathrm{softamx}[\alpha \left(\mathbf{I} - (1-\alpha)\hat{\tilde{A}}\right)^{-1} \boldsymbol{H}]$$

- APPNP: Approximated Personalized Propagation of Neural Prediction

$$\hat{\tilde{\boldsymbol{A}}} = \tilde{\boldsymbol{D}}^{-1/2}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}}^{-1/2}$$

$$\boldsymbol{Z}^{(0)} = \boldsymbol{H} = f_\theta(\boldsymbol{X}),$$

$$\boldsymbol{Z}^{(k+1)} = (1-\alpha)\hat{\tilde{\boldsymbol{A}}}\boldsymbol{Z}^{(k)} + \alpha\boldsymbol{H}, k = 0, \ldots, K-2$$

$$\boxed{\boldsymbol{Z}^{(K)} = \mathrm{softmax}\left((1-\alpha)\hat{\tilde{\boldsymbol{A}}}\boldsymbol{Z}^{(K-1)} + \alpha\boldsymbol{H}\right),}$$

GCN: APPNP

- Predict Then Propagate: Graph Neural Networks Meet Personalized Pagerank [ICLR'19]

Table 2: Average accuracy with uncertainties showing the 95 % confidence level calculated by bootstrapping. Previously reported improvements vanish on our rigorous experimental setup, while PPNP and APPNP significantly outperform the compared models on all datasets.

Model	CITESEER	CORA-ML	PUBMED	MS ACADEMIC
V. GCN	73.51 ± 0.48	82.30 ± 0.34	77.65 ± 0.40	91.65 ± 0.09
GCN	75.40 ± 0.30	83.41 ± 0.39	78.68 ± 0.38	92.10 ± 0.08
N-GCN	74.25 ± 0.40	82.25 ± 0.30	77.43 ± 0.42	92.86 ± 0.11
GAT	75.39 ± 0.27	84.37 ± 0.24	77.76 ± 0.44	91.22 ± 0.07
JK	73.03 ± 0.47	82.69 ± 0.35	77.88 ± 0.38	91.71 ± 0.10
Bt. FP	73.55 ± 0.57	80.84 ± 0.97	72.94 ± 1.00	91.61 ± 0.24
PPNP*	$\mathbf{75.83 \pm 0.27}$	$\mathbf{85.29 \pm 0.25}$	-	-
APPNP	75.73 ± 0.30	85.09 ± 0.25	$\mathbf{79.73 \pm 0.31}$	$\mathbf{93.27 \pm 0.08}$

*out of memory on PUBMED, MS ACADEMIC (see efficiency analysis in Section 3)

Summary Questions of the lecture

- Describe the key idea of APPNP: Approximated Personalized Propagation of Neural Prediction.
- What are the benefits of APPNP: Approximated Personalized Propagation of Neural Prediction.
- Discuss the difference among personalized PageRank, ShevNet, and APPNP.