中国人民大學 RENMIN UNIVERSITY OF CHINA

高瓴人工智能学院 Gaoling School of Artificial Intelligence

图神经网络理论基础

魏哲巍 教授

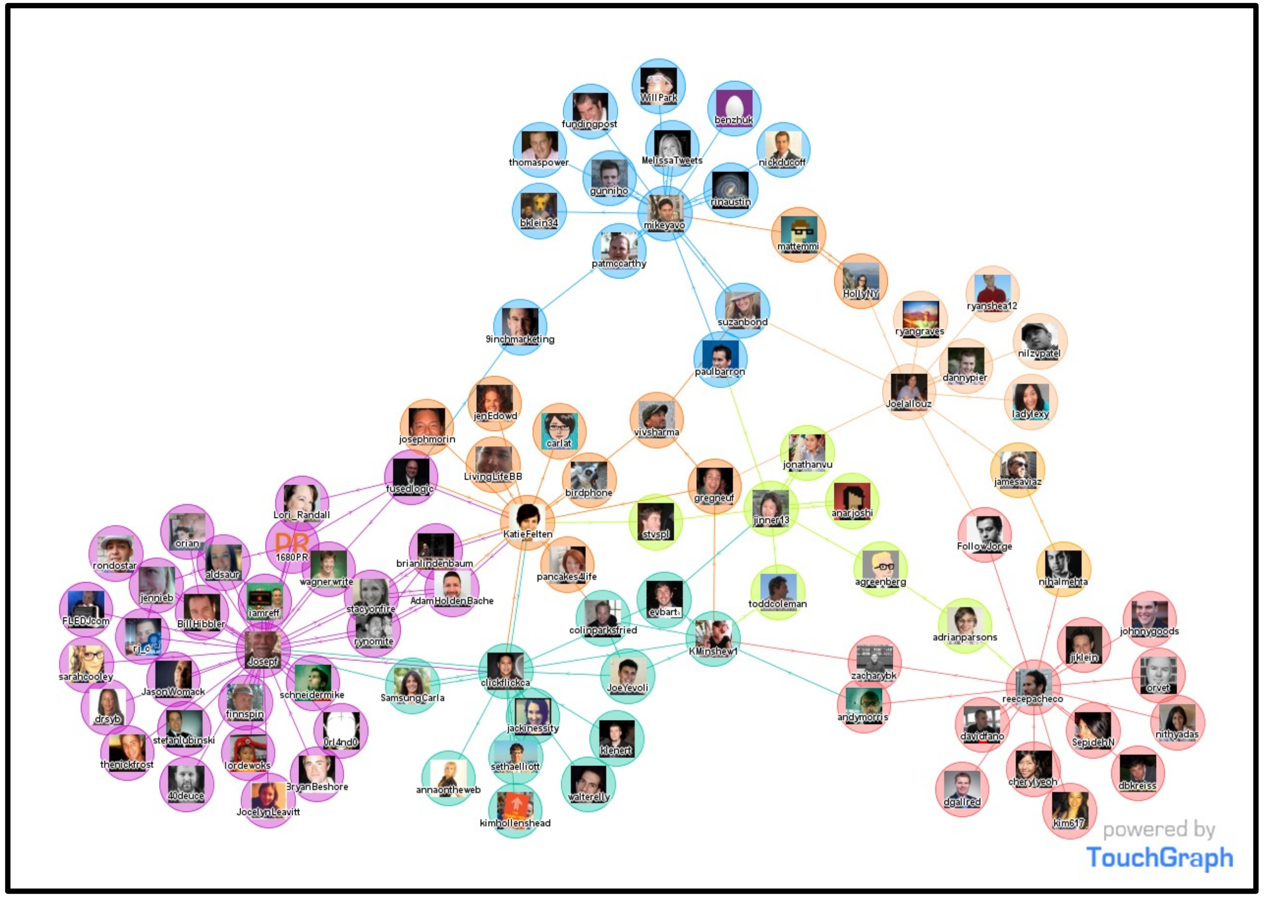
中国人民大学

报告提纲

- 图神经网络的应用和概述
- 图神经网络的三个视角
  - □ 滤波器 ⇨ 学习任意的滤波器
  - □ 随机游走 ⇨ 基于重启随机游走的深度GNN
  - □ 优化函数 ⇨ 统一的GNN优化函数
- 展望与总结

图结构无处不在

社交网络

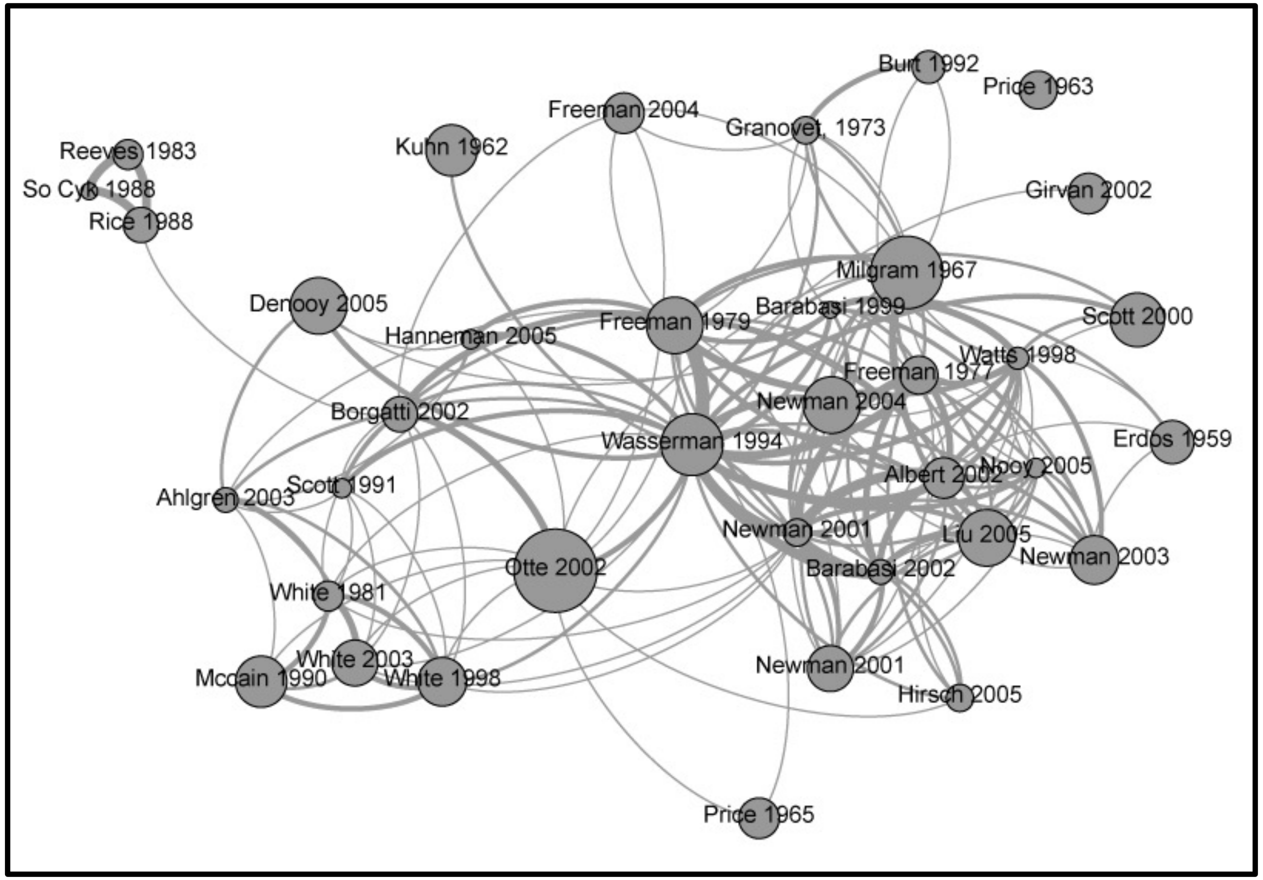


节点：用户

边 ：好友关系；
转发关系；
点赞关系
…

图结构无处不在

论文引用网络

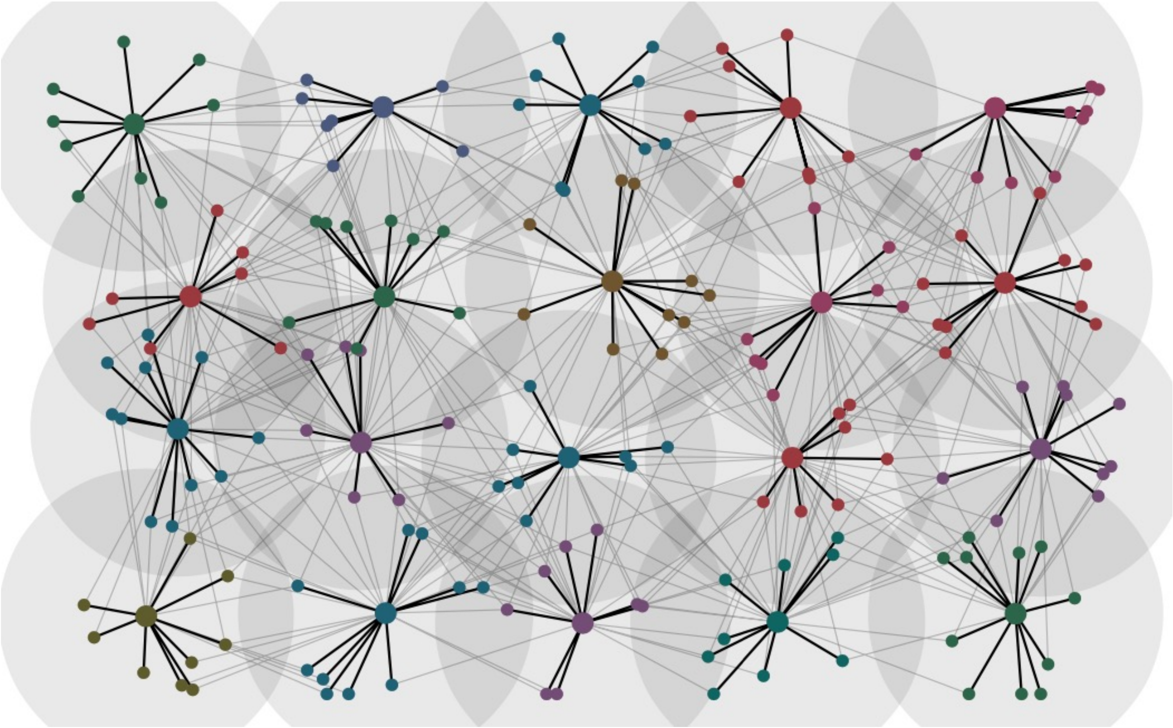


节点：论文

边　：引用关系

图结构无处不在

信号网络

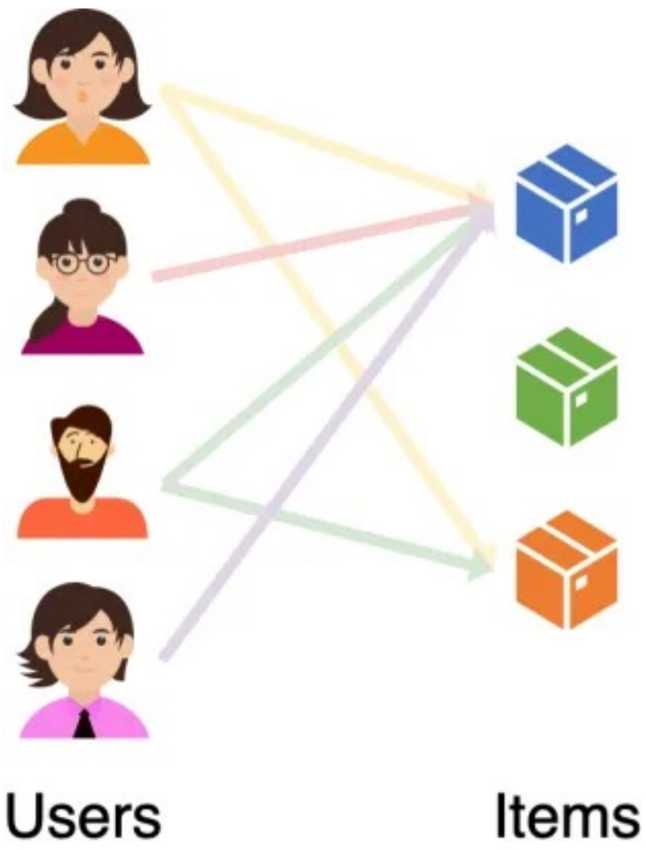


节点：基站、手机

边　：信号传输

图结构无处不在

商品用户二部图



节点：商品/用户

边　：购买关系

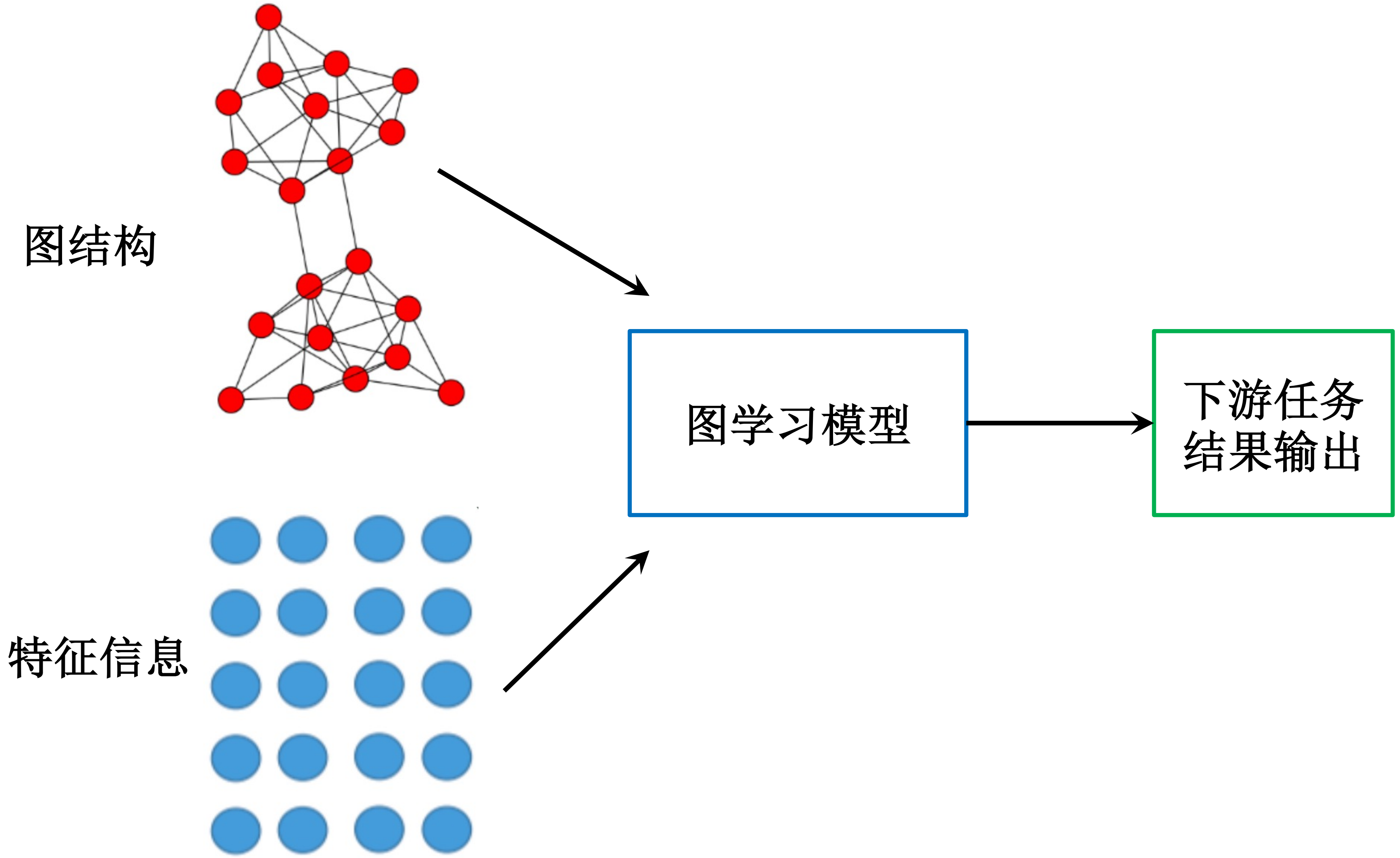
图结构无处不在

点云图

节点：激光点

边：最近邻

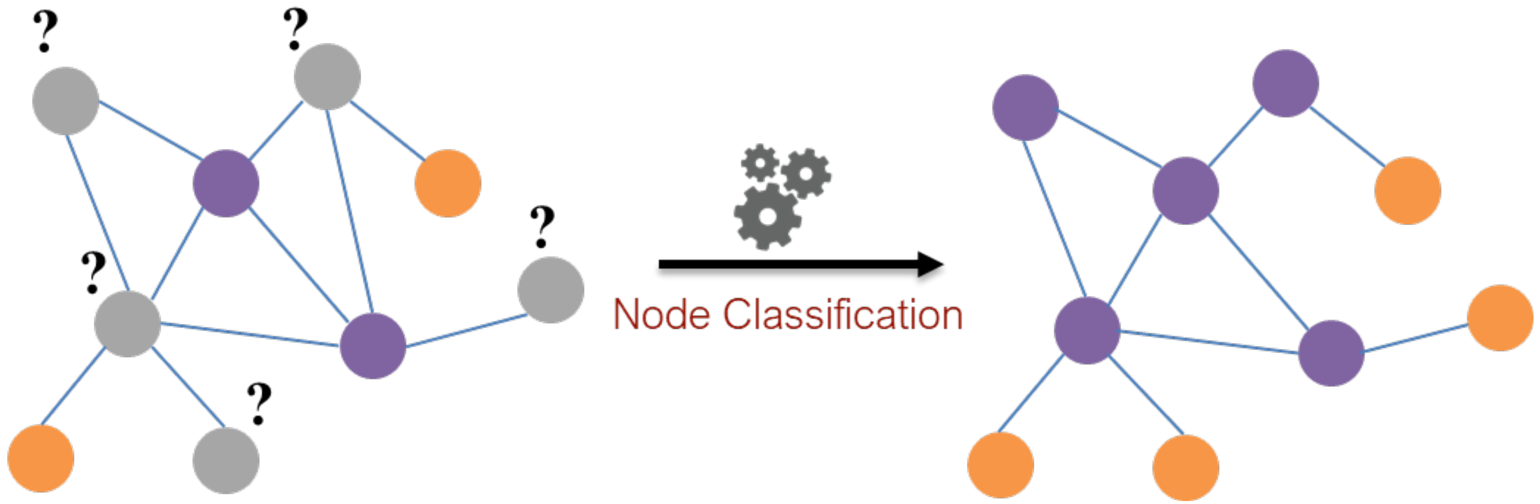
图学习



图学习的任务和应用

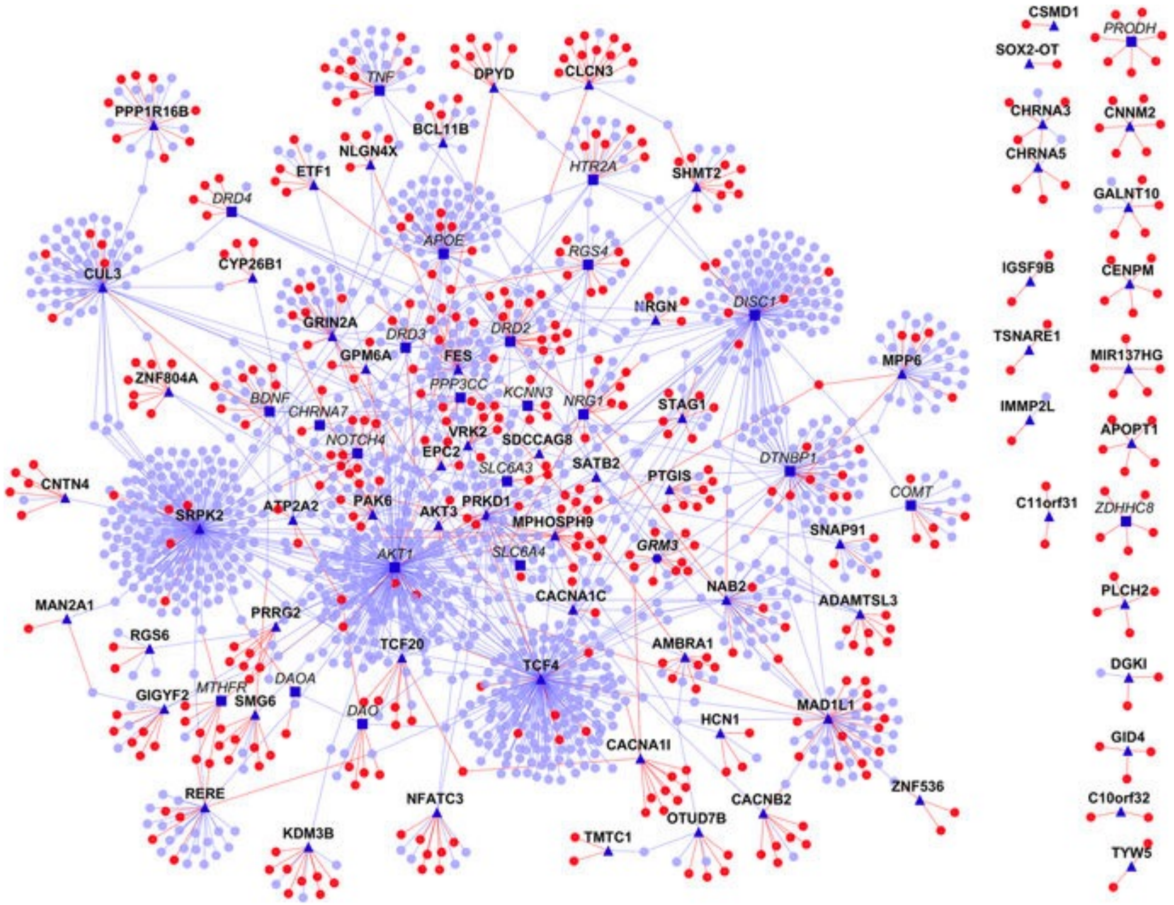
- 图学习的任务
  - 节点分类
  - 连接预测
  - 社区发现
- 图学习的应用
  - 交通预测
  - 物理过程预测
  - 药物研发
  - EDA开发

节点分类



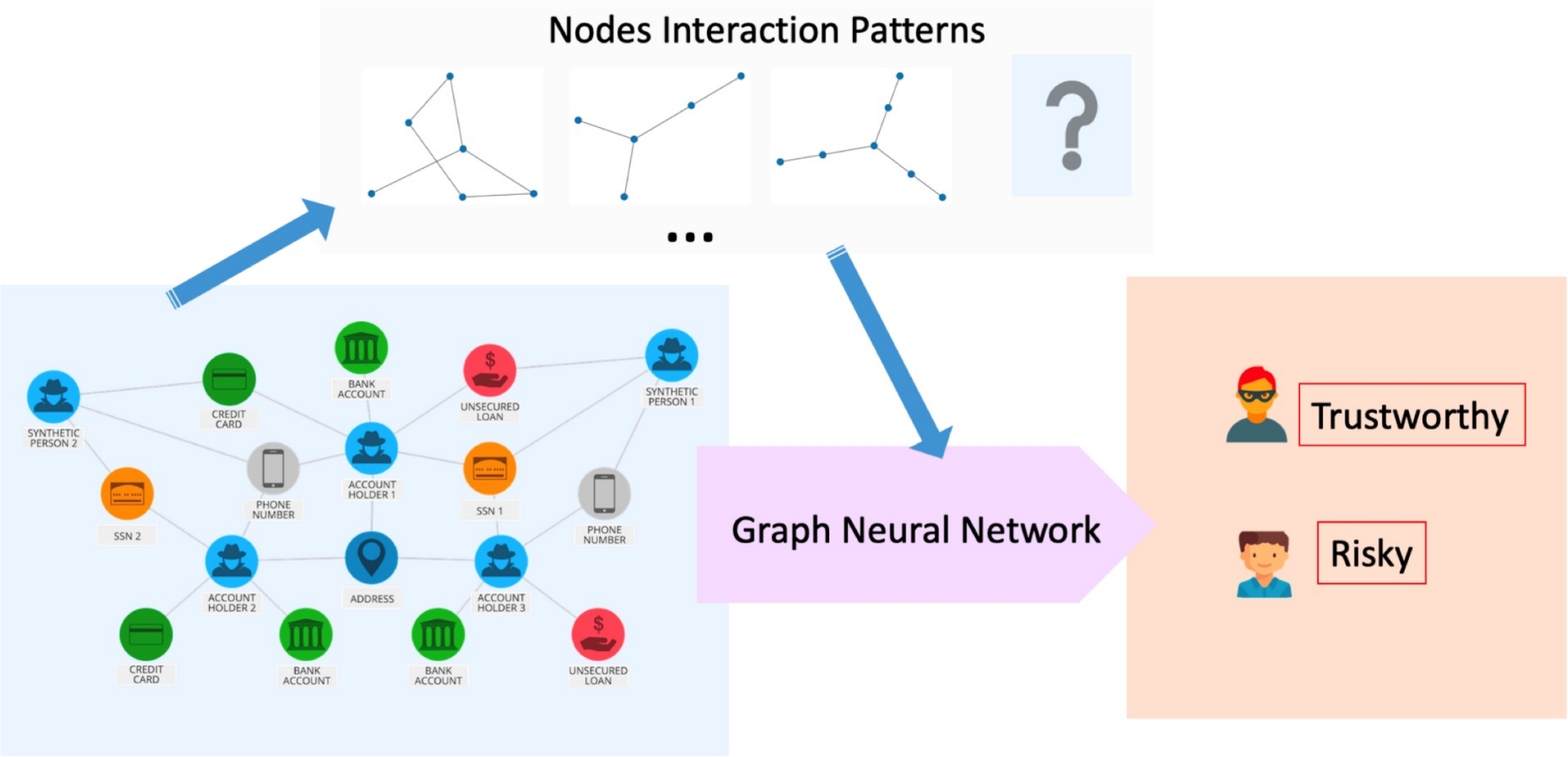
节点分类

- 蛋白质网络中的功能分类
  - 节点：蛋白质；边：蛋白质间的生物性联系；
  - 边特征：联系强度。

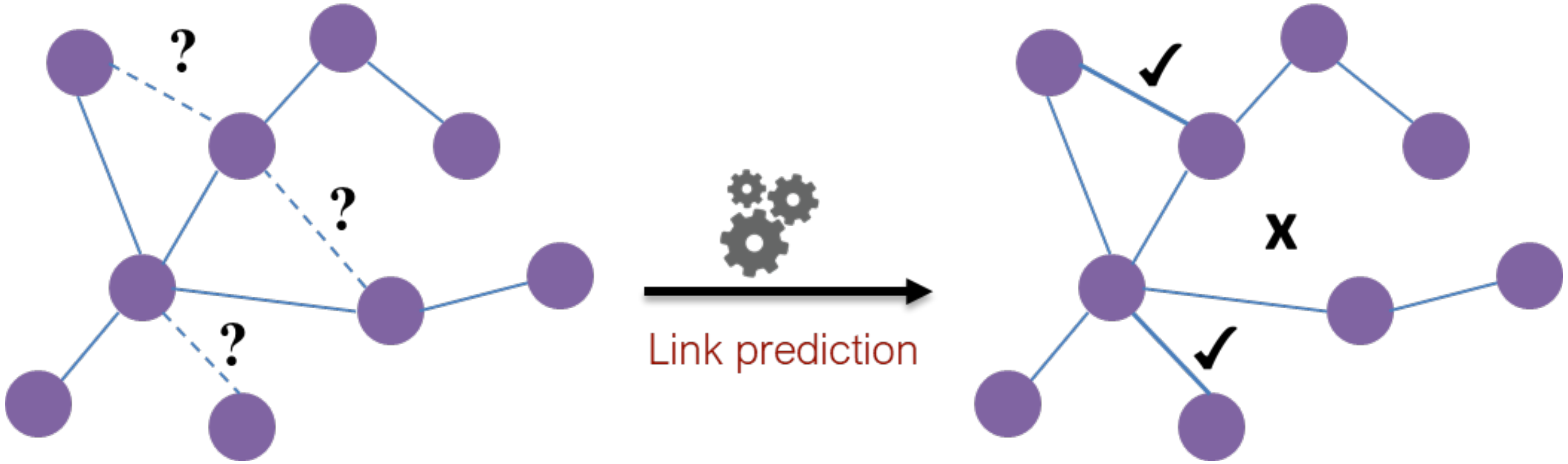


节点分类

- 金融网络中的风险监控
  - 节点：银行、客户...；边：借贷关系；
  - 节点特征：用户画像。

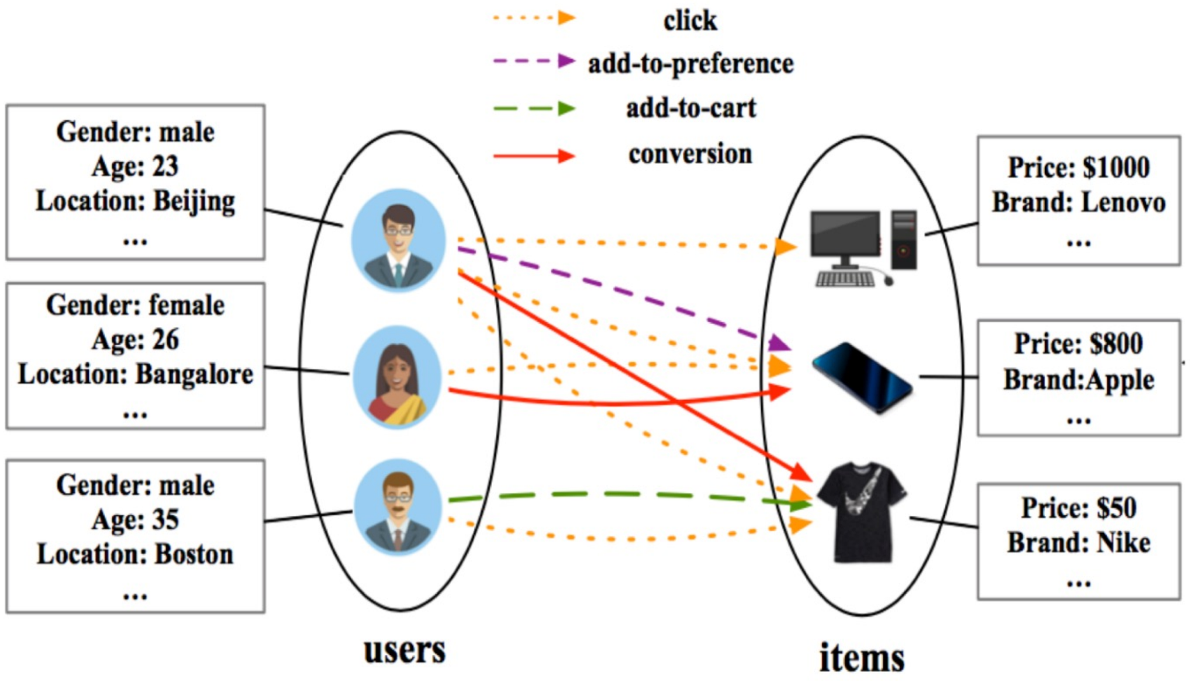


链接预测



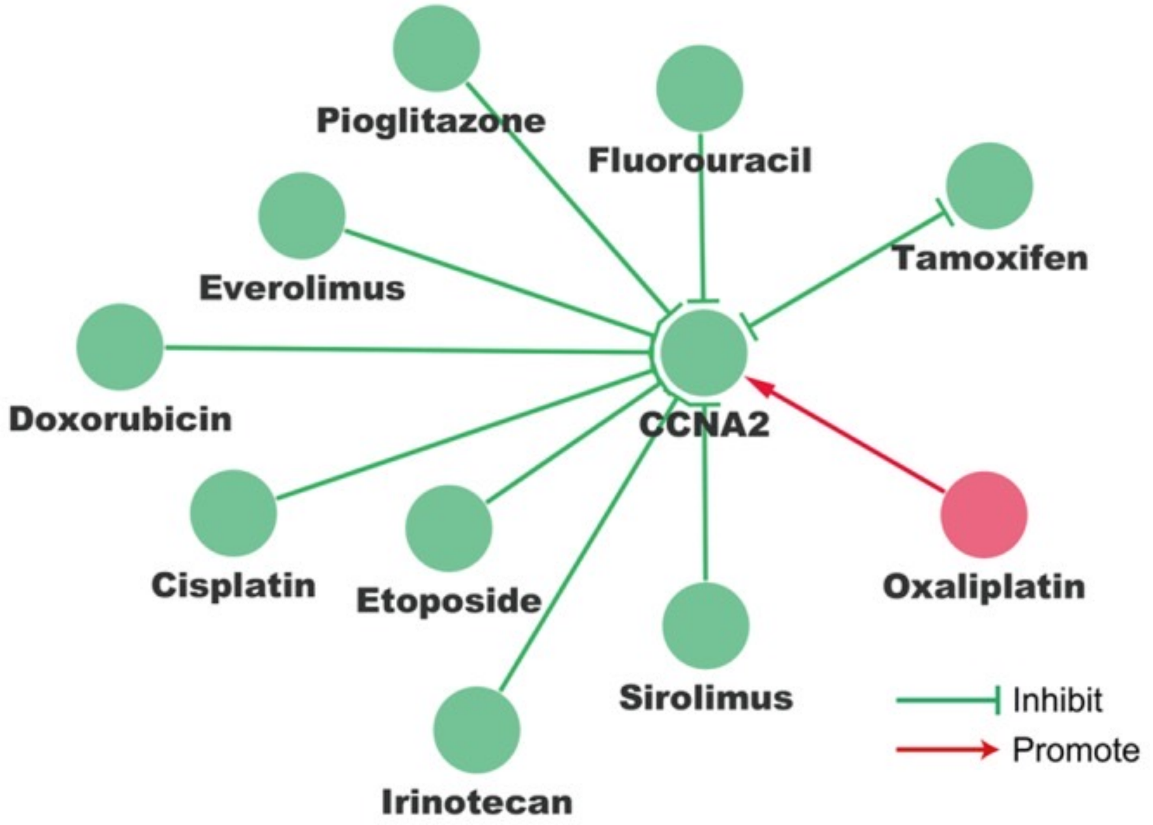
链接预测

- 商品推荐中的链接预测
  - 节点：商品、用户；边：购买、浏览关系；
  - 节点特征：用户属性、商品属性；
  - 边特征：点击动作、收藏动作、下单动作...

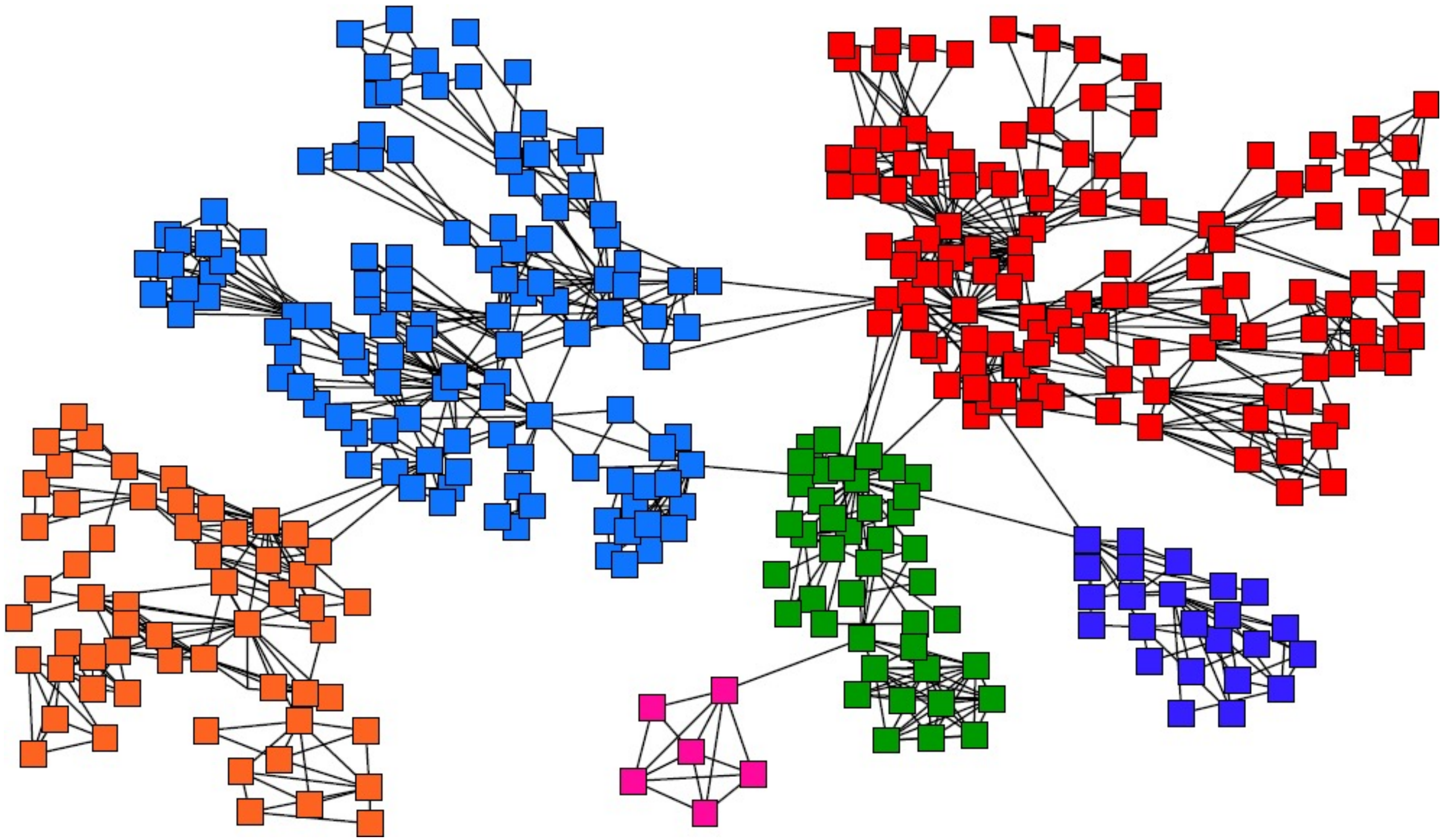


链接预测

- 药物网络中的链接预测
  - □ 节点：药物；边：药物的作用关系；
  - □ 边特征：作用关系类别（抑制、促进）。

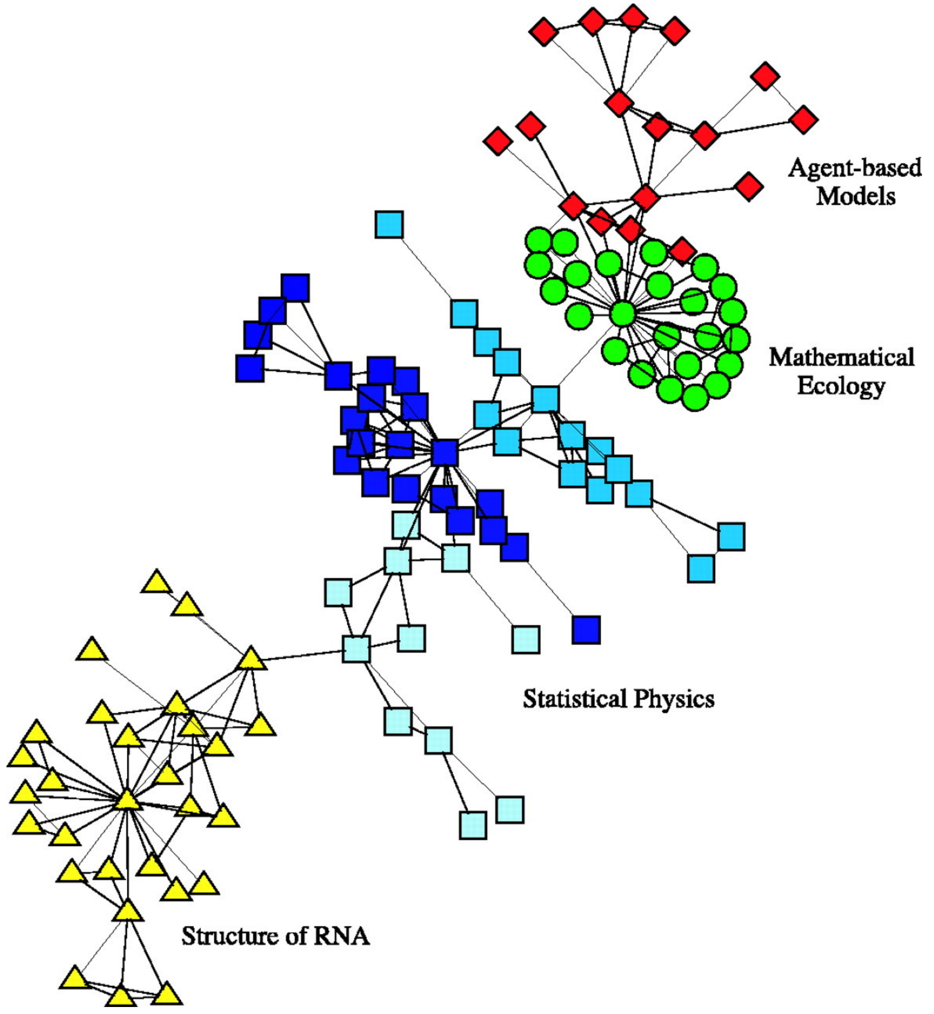


社区发现



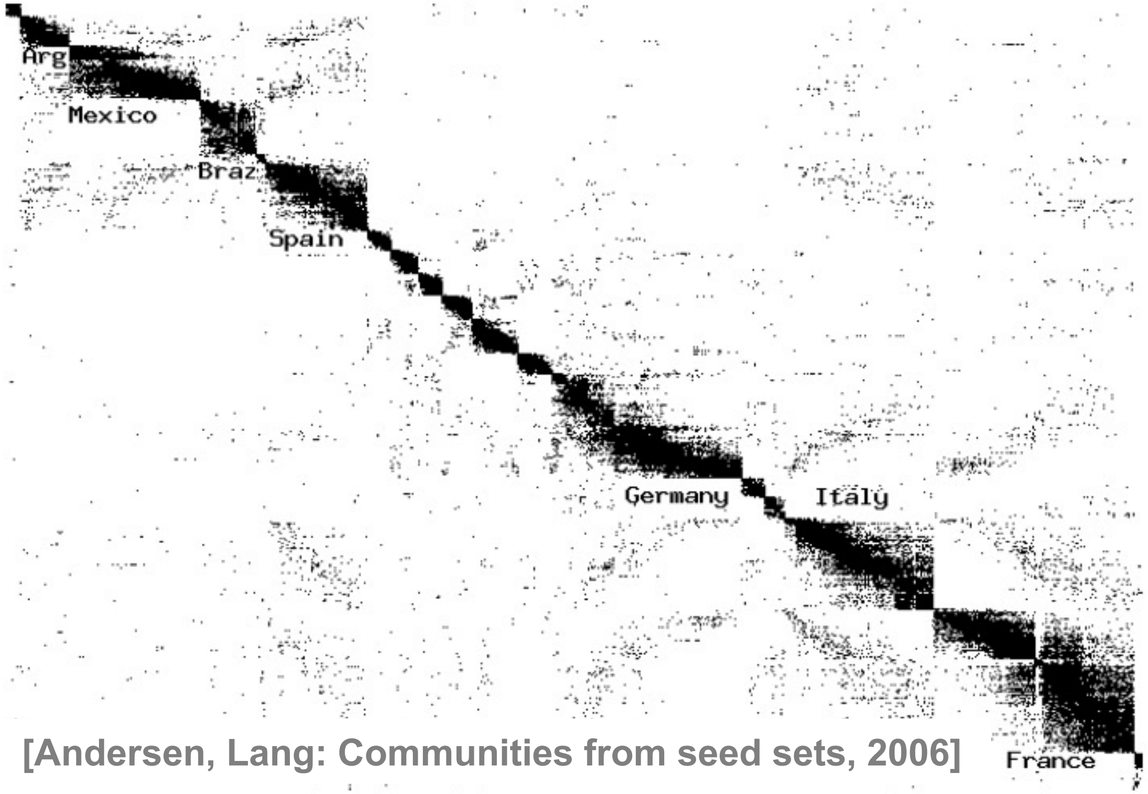
社区发现

- 论文引用网络中的子领域发现
  - 节点：论文；
  - 边：引用关系；
  - 节点特征：论文标题、摘要…



社区发现

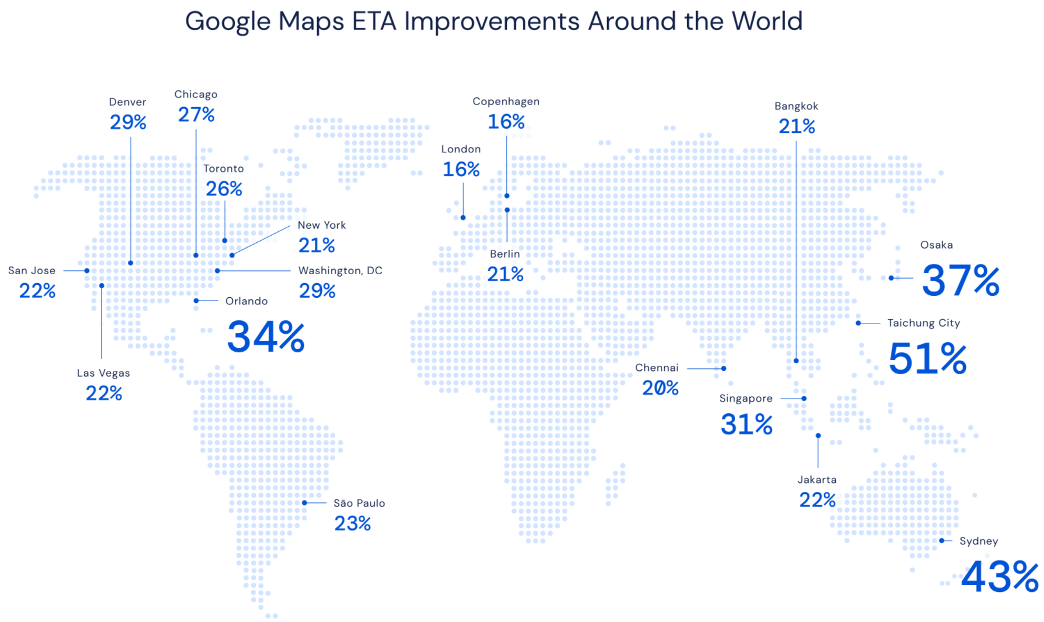
- 电影-演员网络中的社区发现
  - □ 节点：电影、演员；边：演电影关系
  - □ 节点特征：电影属性、演员特征。



[Andersen, Lang: Communities from seed sets, 2006]

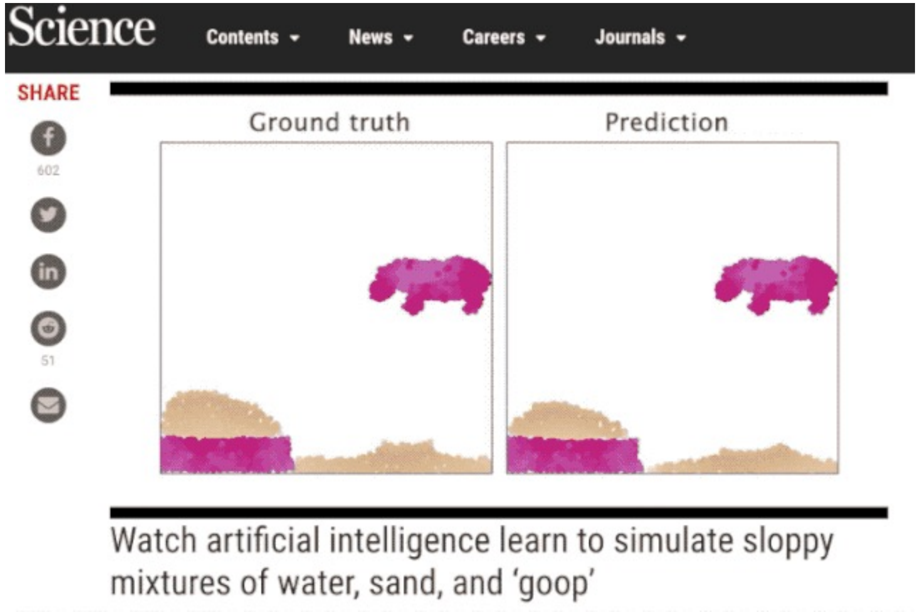
交通预测

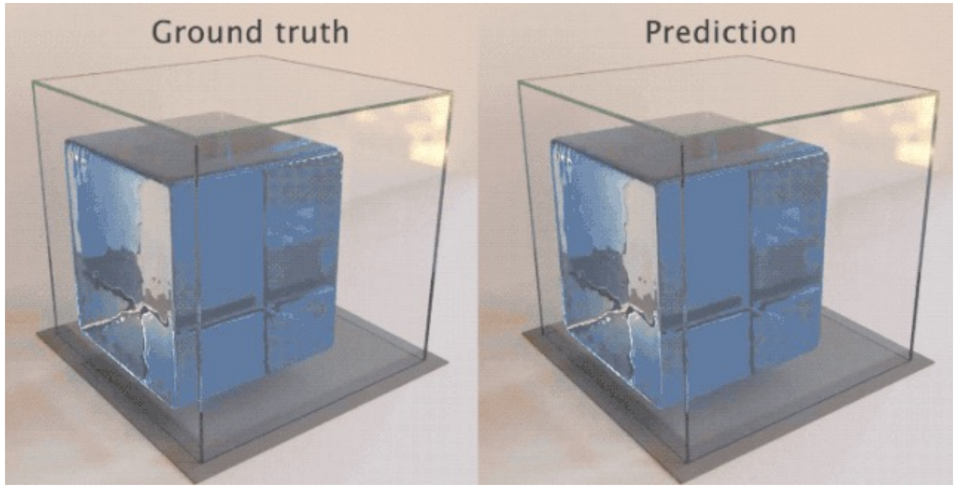
- DeepMind利用图神经网络改进Google Map的预测出行时间

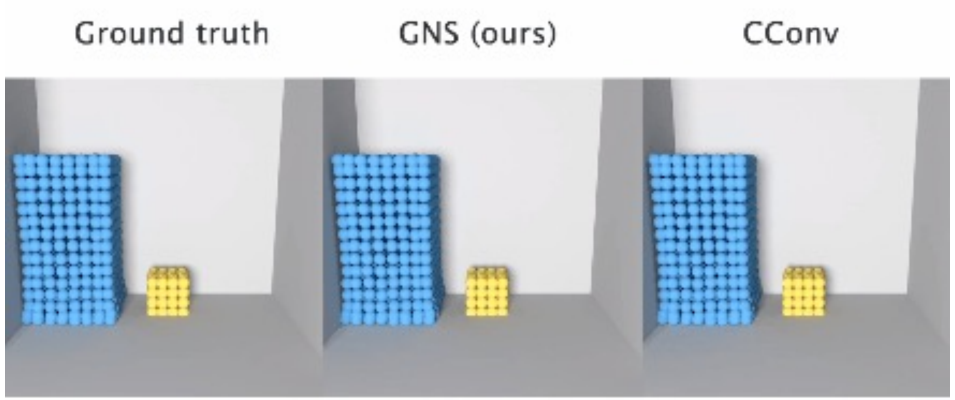


物理过程预测

- DeepMind于ICML2020发表论文，利用图神经网络模拟复杂物理过程。

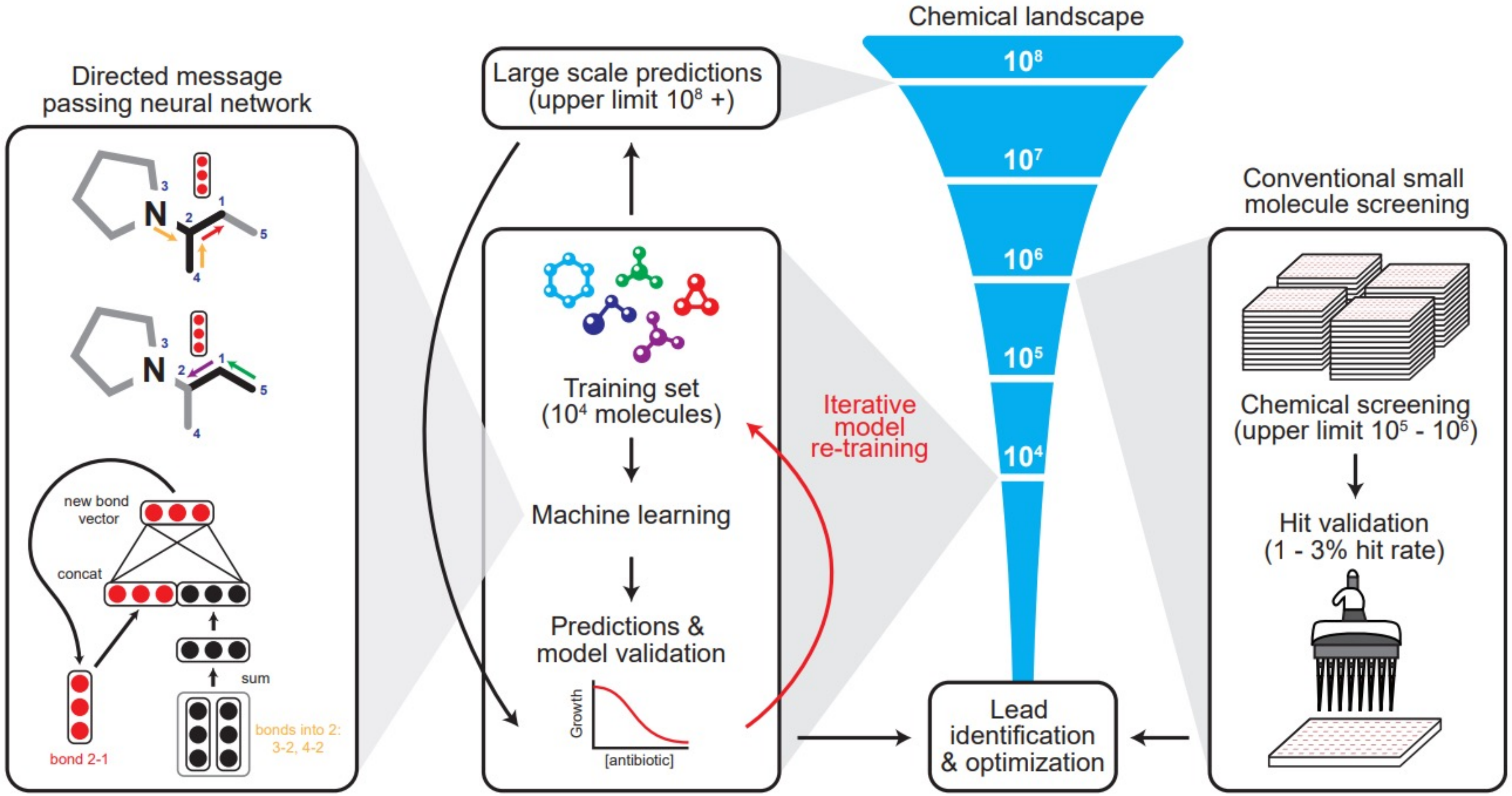






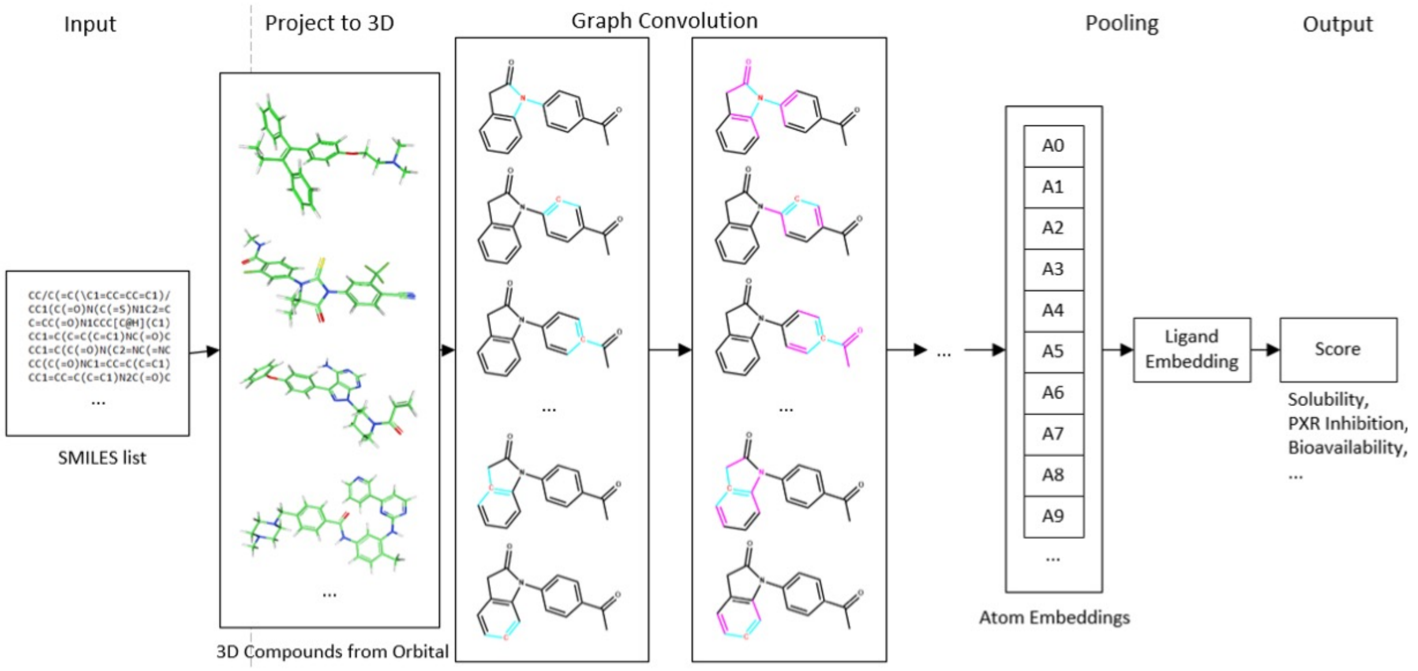
药物研发

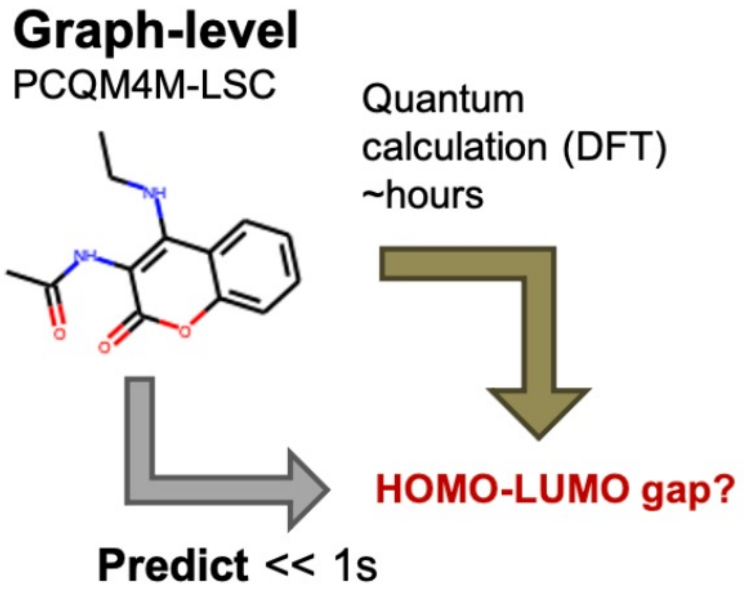
- MIT于Cell2020发表论文，训练深层GNN模型Chemprop以预测分子属性



药物研发

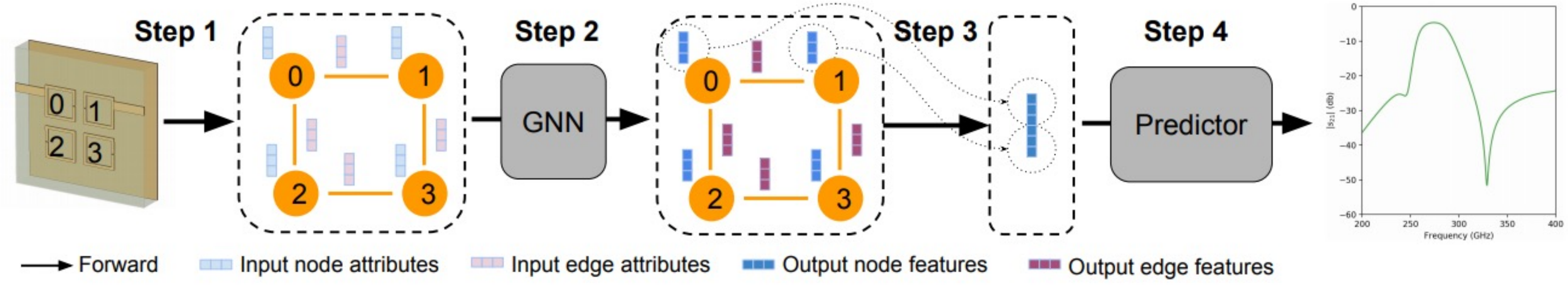
- KDD Cup: 大规模分子性质预测: https://ogb.stanford.edu/kddcup2021



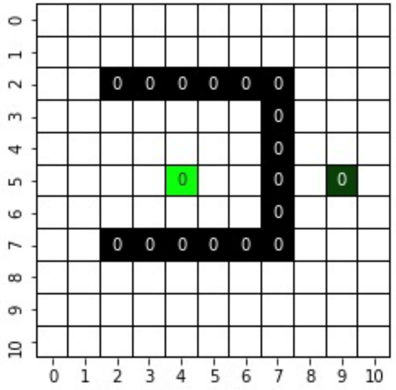


EDA 开发

- MIT于ICML2019发表论文，将电路映射到图结构，利用GNN仿真器件间的电磁耦合

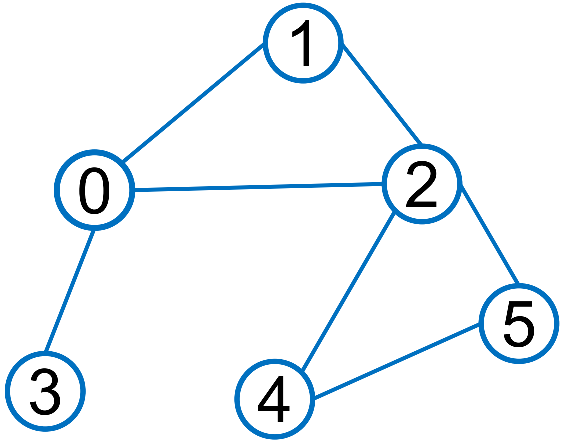


- 利用GNN的局部搜索性质，希望设计超过人设计的heuristic近似/精确算法



图(graph)的定义

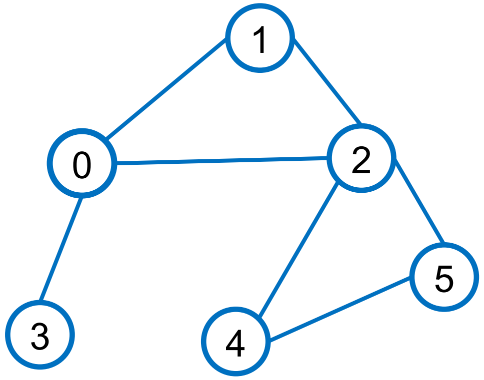
- 图$G = (V, E)$：
  - □ 节点集 V；节点数量 $n = |V|$；
  - □ 边集 E；边数量 $m = |E|$；
  - □ 邻居矩阵 $\boldsymbol{A}$；
  - □ 度矩阵 $\boldsymbol{D}$；
  - □ 归一化邻接矩阵：$\boldsymbol{P} = \boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}$
  - □ 归一化拉普拉斯矩阵：$\boldsymbol{L} = \boldsymbol{I} - \boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}$
  - □ 节点特征矩阵 $\boldsymbol{X} \in \mathcal{R}^{n \times f}$，$f$代表特征维度。



图(graph)的定义

- 图$G = (V, E)$

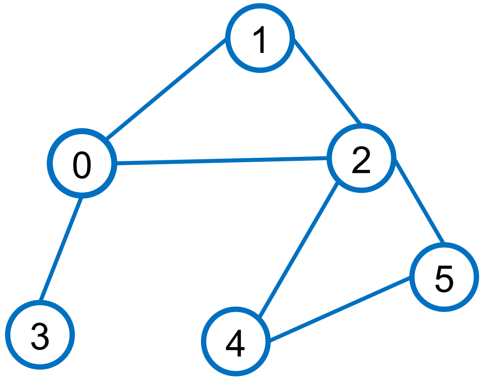
$$A = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$



图(graph)的定义

- 图$G = (V, E)$

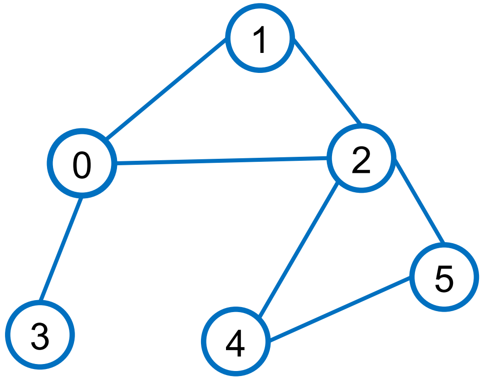
$$
\boldsymbol{D} = \begin{bmatrix} 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}
$$



图(graph)的定义

- 图$G = (V, E)$

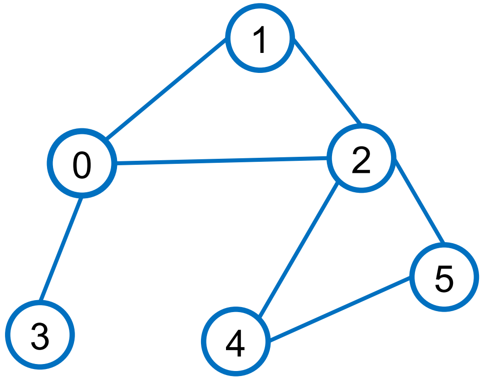
$$D^{-1/2} = \begin{bmatrix} \frac{1}{\sqrt{3}} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} \end{bmatrix}$$



图(graph)的定义

- 图$G = (V, E)$

$$\boldsymbol{P = D^{-1/2}AD^{-1/2}}$$



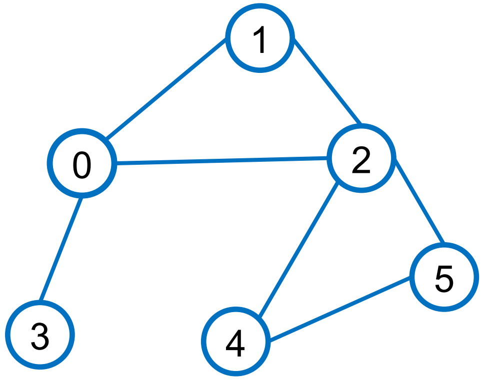
$$\boldsymbol{P} = \begin{bmatrix} 0 & \frac{1}{\sqrt{3}\cdot\sqrt{2}} & \frac{1}{\sqrt{3}\cdot\sqrt{4}} & \frac{1}{\sqrt{3}\cdot\sqrt{1}} & 0 & 0 \\ \frac{1}{\sqrt{3}\cdot\sqrt{2}} & 0 & \frac{1}{\sqrt{2}\cdot\sqrt{4}} & 0 & 0 & 0 \\ \frac{1}{\sqrt{3}\cdot\sqrt{4}} & \frac{1}{\sqrt{2}\cdot\sqrt{4}} & 0 & 0 & \frac{1}{\sqrt{4}\cdot\sqrt{2}} & \frac{1}{\sqrt{4}\cdot\sqrt{2}} \\ \frac{1}{\sqrt{3}\cdot\sqrt{1}} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{\sqrt{4}\cdot\sqrt{2}} & 0 & \frac{1}{\sqrt{2}\cdot\sqrt{2}} \\ 0 & 0 & 0 & \frac{1}{\sqrt{4}\cdot\sqrt{2}} & \frac{1}{\sqrt{2}\cdot\sqrt{2}} & 0 \end{bmatrix}$$

图(graph)的定义

$$L = I - D^{-1/2} A D^{-1/2}$$

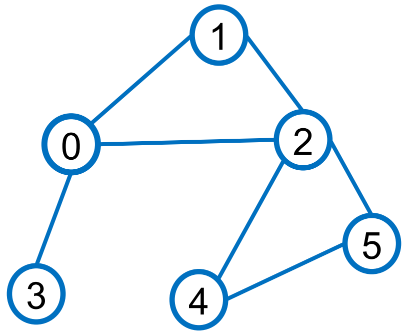
- 图$G = (V, E)$

$$L = \begin{bmatrix} 1 & \frac{-1}{\sqrt{3}\cdot\sqrt{2}} & \frac{-1}{\sqrt{3}\cdot\sqrt{4}} & \frac{-1}{\sqrt{3}\cdot\sqrt{1}} & 0 & 0 \\ \frac{-1}{\sqrt{3}\cdot\sqrt{2}} & 1 & \frac{-1}{\sqrt{2}\cdot\sqrt{4}} & 0 & 0 & 0 \\ \frac{-1}{\sqrt{3}\cdot\sqrt{4}} & \frac{-1}{\sqrt{2}\cdot\sqrt{4}} & 1 & 0 & \frac{-1}{\sqrt{4}\cdot\sqrt{2}} & \frac{-1}{\sqrt{4}\cdot\sqrt{2}} \\ \frac{-1}{\sqrt{3}\cdot\sqrt{1}} & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \frac{-1}{\sqrt{4}\cdot\sqrt{2}} & 1 & \frac{-1}{\sqrt{2}\cdot\sqrt{2}} \\ 0 & 0 & 0 & \frac{-1}{\sqrt{4}\cdot\sqrt{2}} & \frac{-1}{\sqrt{2}\cdot\sqrt{2}} & 1 \end{bmatrix}$$



图(graph)的定义

- 节点特征矩阵 $\boldsymbol{X} \in \mathcal{R}^{n \times f}$，$f$代表特征维度。

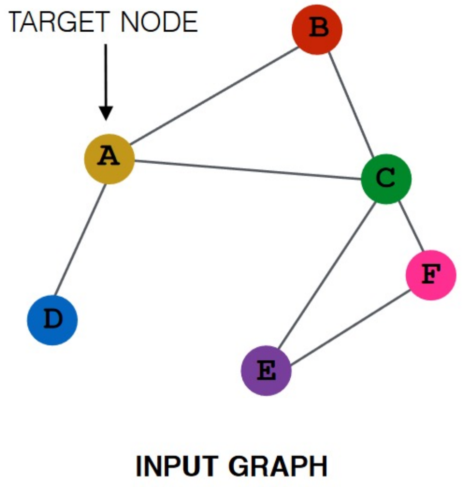


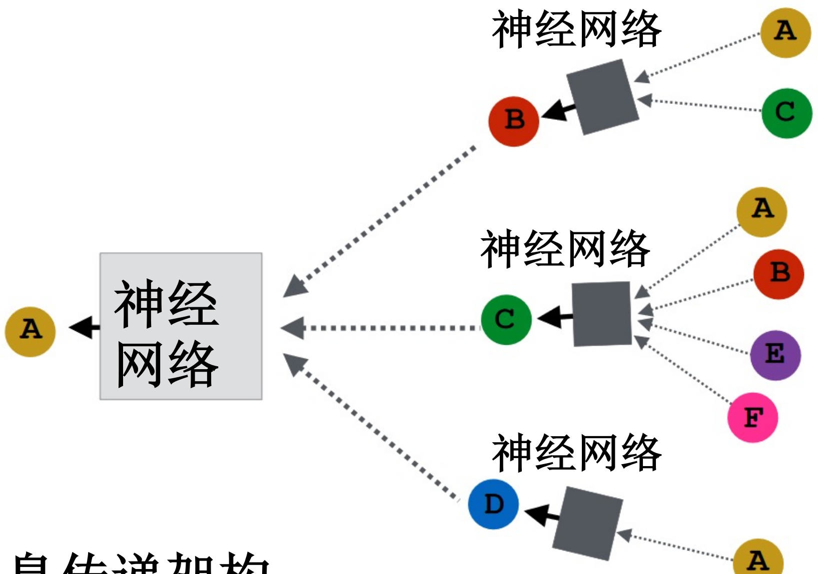
Node	Features1	Features2	Features3	Features4	Features5	Features6
$\mathbf{x}_0$	1	0	0	0	0	0
$\mathbf{x}_1$	0	1	0	0	0	0
$\mathbf{x}_2$	0	0	1	0	0	0
$\mathbf{x}_3$	0	0	0	1	0	0
$\mathbf{x}_4$	0	0	0	0	1	0
$\mathbf{x}_5$	0	0	0	0	0	1

图神经网络

- 图卷积神经网络 (GCN) [Kipf et al.,2017]
  - 在图结构上聚合邻居的特征信息，进行消息传递；
  - 借助神经网络训练消息传递中的权重；
  - 遵循消息传递架构：

$$\boldsymbol{H}^{(\ell+1)}=\sigma\left(\widetilde{\boldsymbol{P}}\boldsymbol{H}^{(\ell)}\boldsymbol{W}^{(\ell)}\right)$$





GCN消息传递架构

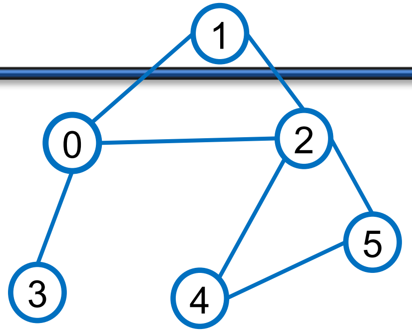
GCN消息消息传播框架

$$H^{(\ell+1)} = \sigma\left(\tilde{P} \cdot H^{(\ell)} \cdot W^{(\ell)}\right)$$

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

上一层的表示结果

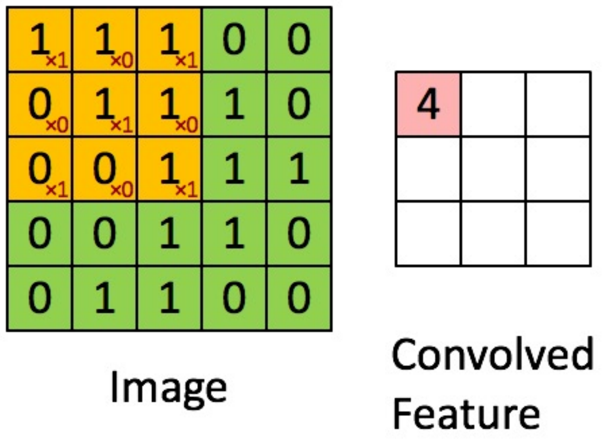
待学习权重矩阵

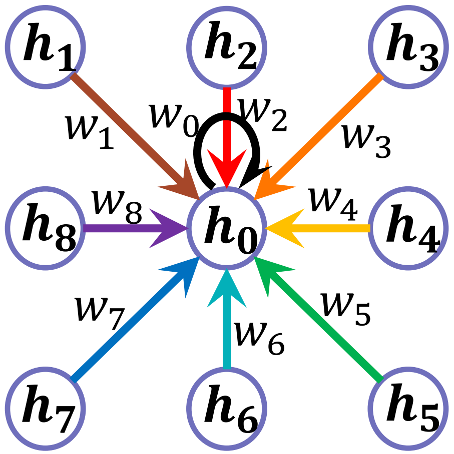


Node	Features1	Features2	Features3	Features4	Features5	Features6
x_0	1	0	0	0	0	0
x_1	0	1	0	0	0	0
x_2	0	0	1	0	0	0
x_3	0	0	0	1	0	0
x_4	0	0	0	0	1	0
x_5	0	0	0	0	0	1
Sum	**0**	**1**	**1**	**1**	**0**	**0**
Self-loop	**1**	**1**	**1**	**1**	**0**	**0**
Symmetry	**1/**$\sqrt{4}\,\sqrt{4}$	**1/** $\sqrt{4}\,\sqrt{3}$	**1/** $\sqrt{4}\,\sqrt{5}$	**1/** $\sqrt{4}\,\sqrt{2}$	**0**	**0**

GCN与CNN

- CNN也是一种图卷积神经网络(GCN)：
  - 聚合周围八个邻居节点和自身的特征信息





报告提纲

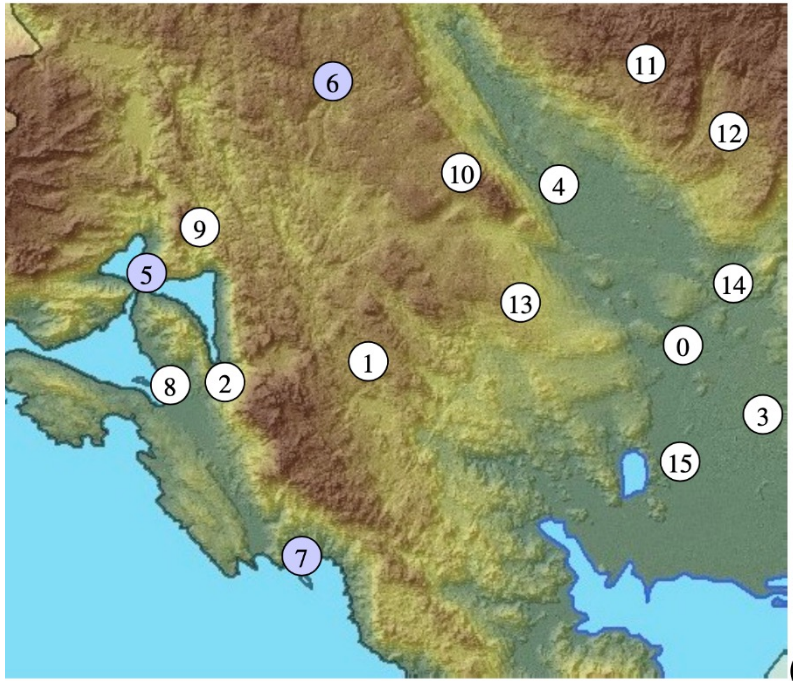
- 图神经网络的应用和概述
- 图神经网络的三个视角
  - 滤波器 → 学习任意的滤波器
  - 随机游走 → 基于重启随机游走的深度GNN
  - 优化函数 → 统一的GNN优化函数
- 展望与总结

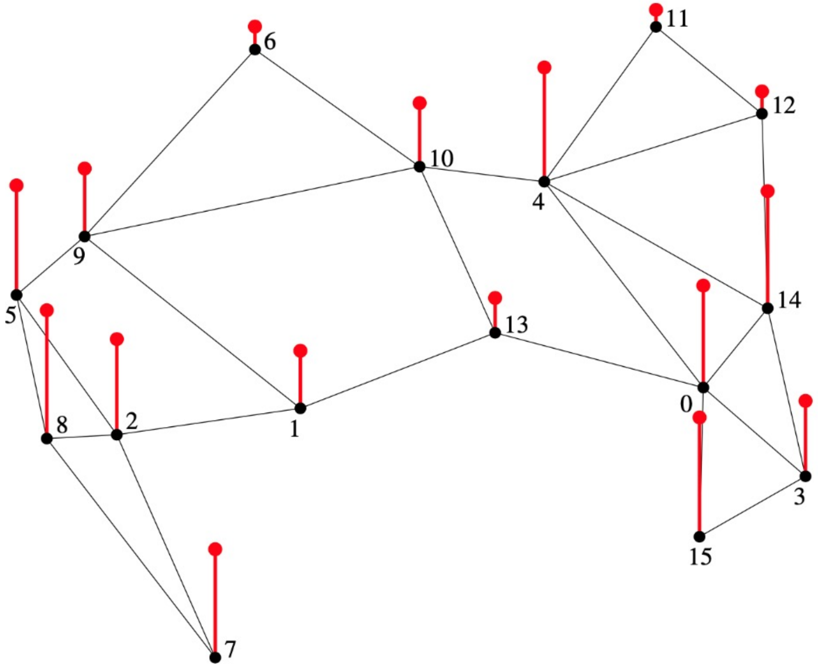
相关工作

论文	主要方法
Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering (ChebNet, Defferrard et al., 2016)	利用Chebyshev多项式近似滤波器
Semi-Supervised Classification with Graph Convolutional Networks (GCN, Kipf et al.,2017)	简化的二阶Chebyshev多项式作为滤波器
Adaptive Universal Generalized Pagerank Graph Neural Network (GPR-GNN, Chien et al., 2021)	直接学习多项式滤波器的系数近似滤波器
Interpreting and Unifying Graph Neural Networks with An Optimization Framework (GNN-LF/HF, Zhu et al.,2021)	从优化函数的角度设计低通/高通滤波器
BernNet: Learning Arbitrary Graph Spectral Filters via Bernstein Approximation (BernNet, He et al., 2021)	使用Bernstein多项式学习任意的滤波器（Ours）
Graph Neural Networks with Convolutional ARMA Filters (ARMA, Bianchi et al., 2021)	利用ARMA滤波器族学习滤波器

图信号

- 传感器测量的温度作为图信号，用向量$\boldsymbol{x} \in \mathcal{R}^n$表示

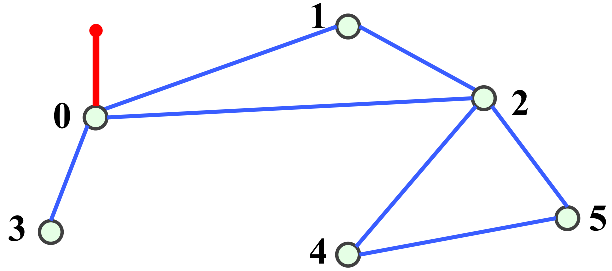




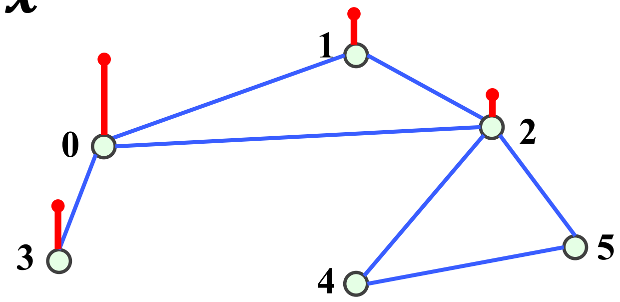
图信号

- 通过拉普拉斯矩阵$\boldsymbol{L}$对图信号进行操作

$$\boldsymbol{x}_1 = (2\boldsymbol{I} - \boldsymbol{L})\boldsymbol{x}$$

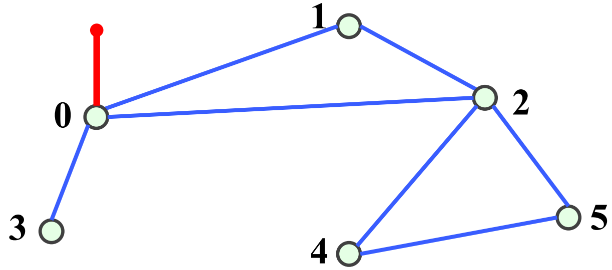


$\boldsymbol{x}$

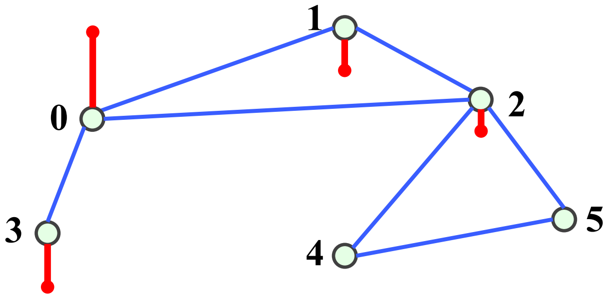


$\boldsymbol{x}_1$

$$\boldsymbol{x}_2 = \boldsymbol{L}\boldsymbol{x}$$



$\boldsymbol{x}$



$\boldsymbol{x}_2$

图傅里叶变换

- 拉普拉斯矩阵特征分解

$$\boldsymbol{L} = \boldsymbol{U\Lambda U}^T = \boldsymbol{U}\begin{pmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_n \end{pmatrix}\boldsymbol{U}^T$$

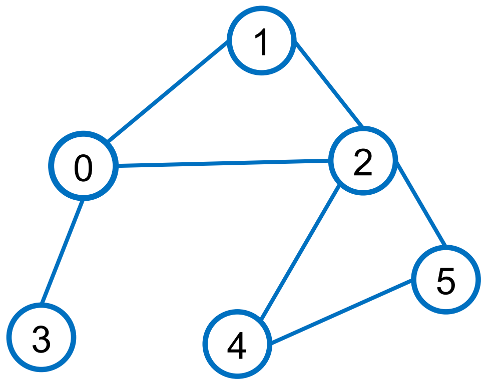
其中$\boldsymbol{U} = [\boldsymbol{u}_1, \dots, \boldsymbol{u}_n]$，$\boldsymbol{\Lambda} = \mathrm{diag}([\lambda_1, \dots, \lambda_n])$， $\boldsymbol{u}_i$ 和 λ_i for $i \in \{1,2,\dots,n\}$分别表示特征向量和特征值，且$\lambda_i \in [0,2]$。

- 图信号的傅里叶变换：$\hat{\boldsymbol{x}} = \boldsymbol{U}^T\boldsymbol{x}$
- 图信号的傅里叶逆变换：$\boldsymbol{x} = \boldsymbol{U}\hat{\boldsymbol{x}}$

图傅里叶变换

- 拉普拉斯矩阵特征分解

$$\boldsymbol{L} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T = \boldsymbol{U}\begin{pmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_n \end{pmatrix}\boldsymbol{U}^T$$



$$\underbrace{\begin{bmatrix} 1 & \frac{-1}{\sqrt{3}\cdot\sqrt{2}} & \frac{-1}{\sqrt{3}\cdot\sqrt{4}} & \frac{-1}{\sqrt{3}\cdot\sqrt{1}} & 0 & 0 \\ \frac{-1}{\sqrt{3}\cdot\sqrt{2}} & 1 & \frac{-1}{\sqrt{2}\cdot\sqrt{4}} & 0 & 0 & 0 \\ \frac{-1}{\sqrt{3}\cdot\sqrt{4}} & \frac{-1}{\sqrt{2}\cdot\sqrt{4}} & 1 & 0 & \frac{-1}{\sqrt{4}\cdot\sqrt{2}} & \frac{-1}{\sqrt{4}\cdot\sqrt{2}} \\ \frac{-1}{\sqrt{3}\cdot\sqrt{1}} & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \frac{-1}{\sqrt{4}\cdot\sqrt{2}} & 1 & \frac{-1}{\sqrt{2}\cdot\sqrt{2}} \\ 0 & 0 & 0 & \frac{-1}{\sqrt{4}\cdot\sqrt{2}} & \frac{-1}{\sqrt{2}\cdot\sqrt{2}} & 1 \end{bmatrix}}_{\boldsymbol{L}} \cdot \underbrace{\begin{bmatrix} \sqrt{3} \\ \sqrt{2} \\ \sqrt{4} \\ \sqrt{1} \\ \sqrt{2} \\ \sqrt{2} \end{bmatrix}}_{\boldsymbol{D}^{1/2}\mathbf{1}} = \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{\mathbf{0}}$$

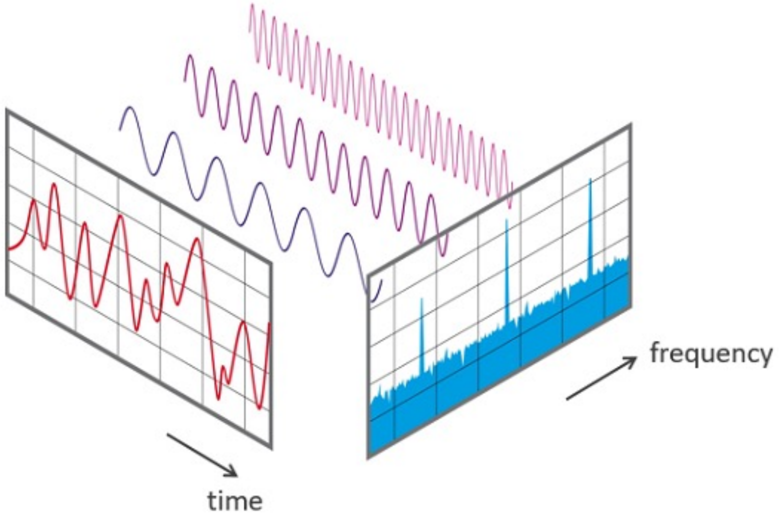
最小的特征值 $\lambda_1 = 0$

对应的特征向量 $\boldsymbol{u}_1 = \boldsymbol{D}^{1/2}\mathbf{1}$

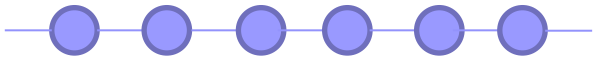
$$\begin{bmatrix} \sqrt{3} & 0 & 0 & 0 & 0 & 0 \\ 0 & \sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \sqrt{4} & 0 & 0 & 0 \\ 0 & 0 & 0 & \sqrt{1} & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \sqrt{2} \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} \sqrt{3} \\ \sqrt{2} \\ \sqrt{4} \\ \sqrt{1} \\ \sqrt{2} \\ \sqrt{2} \end{bmatrix}$$

图傅里叶变换

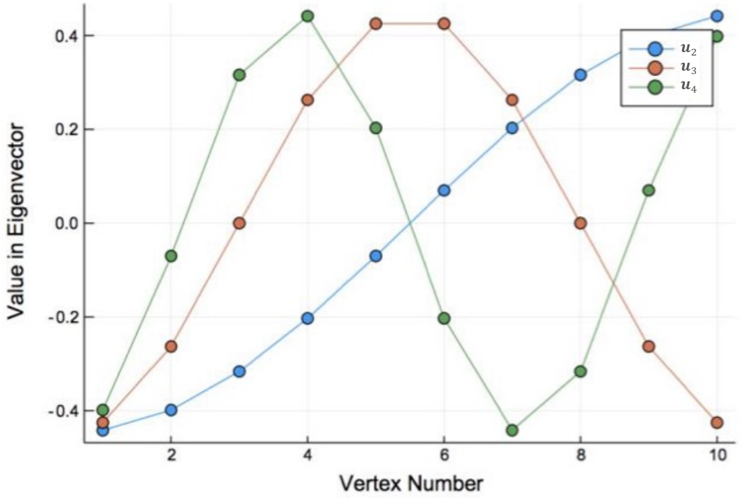
时域-频域傅里叶变换：



图傅里叶变换：



Path Graph



Eigenvectors of Path Graph

$$\lambda_1 = 0$$
$$\boldsymbol{u}_1 = \boldsymbol{D}^{1/2}\mathbf{1}$$

滤波器

- 对图信号$\boldsymbol{x}$的滤波操作定义为：

$$\boldsymbol{y} = h(\boldsymbol{L})\boldsymbol{x} = \boldsymbol{U}h(\boldsymbol{\Lambda})\boldsymbol{U}^T\boldsymbol{x}$$

$$= \boxed{\boldsymbol{U}}\ diag[h(\lambda_1), \dots, h(\lambda_n)]\ \boxed{\boldsymbol{U}^T\boldsymbol{x}}$$

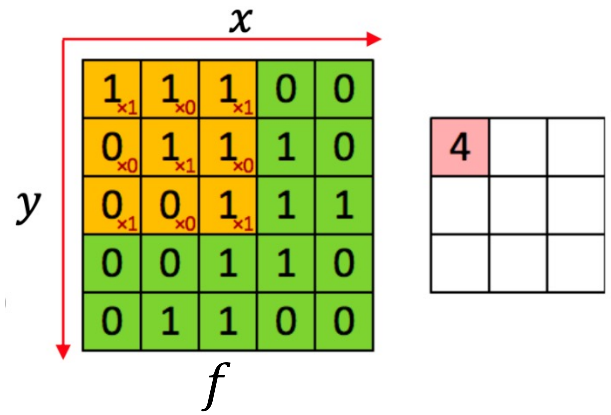
3.图傅里叶逆变换 ← 2.频谱域滤波操作 ← 1.图信号傅里叶变换

$\boldsymbol{U}h(\boldsymbol{\Lambda})\boldsymbol{U}^T\boldsymbol{x}$ $h(\boldsymbol{\Lambda})\boldsymbol{U}^T\boldsymbol{x}$ $\boldsymbol{U}^T\boldsymbol{x}$

- $h(\Lambda)/h(\lambda)$称为滤波器。

图卷积

- 传统意义上的离散卷积：$(f * g)(x,y) \stackrel{\text{def}}{=} \sum_{m,n} f(x-m, y-n)g(m,n)$



$g=$

$g(1,1)$ 1	$g(0,1)$ 0	$g(-1,1)$ 1
$g(1,0)$ 0	$g(0,0)$ 1	$g(-1,0)$ 0
$g(1,-1)$ 1	$g(0,-1)$ 0	$g(-1,-1)$ 1

  - □ g即为深度学习里常说的核(Kernel)，也可对应到信号处理中的滤波器(Filter)

- 图信号x和y的卷积：其傅立叶变换的Hadamard积的逆变换(卷积定理)

$$\boldsymbol{x} *_G \boldsymbol{y} = \boldsymbol{U}\left((\boldsymbol{U}^T\boldsymbol{x}) \odot (\boldsymbol{U}^T\boldsymbol{y})\right)$$

其中，$\odot$表示Hadamard积，$\boldsymbol{U}^T\boldsymbol{y}$称为谱域卷积滤波器。

图卷积

- 参数化$\boldsymbol{U}^T\boldsymbol{y}=[\theta_1,\ldots,\theta_n]^T$，记$h_\theta=\mathrm{diag}([\theta_1,\ldots,\theta_n])$作为对角矩阵表示谱域卷积滤波器

$$\boldsymbol{x} *_G \boldsymbol{y}=\boldsymbol{U}\left((\boldsymbol{U}^T\boldsymbol{x})\odot(\boldsymbol{U}^T\boldsymbol{y})\right)$$

$$\boldsymbol{x} *_G \boldsymbol{y}=\boxed{\boldsymbol{U}}h_\theta\boxed{\boldsymbol{U}^T\boldsymbol{x}}$$

3.图傅里叶逆变换 ← 2.谱域图卷积 ← 1.图信号傅里叶变换

$\boldsymbol{U}\boldsymbol{h}_\theta\boldsymbol{U}^T\boldsymbol{x}$ $\boldsymbol{h}_\theta\boldsymbol{U}^T\boldsymbol{x}$ $\boldsymbol{U}^T\boldsymbol{x}$

- 将h_θ作为$\boldsymbol{L}$特征值的函数$h(\boldsymbol{\Lambda})$，则得到了对图信号的滤波操作

滤波器

- 对图信号$\boldsymbol{x}$的滤波操作定义为：

$$\boldsymbol{y} = h(\boldsymbol{L})\boldsymbol{x} = \boldsymbol{U}h(\boldsymbol{\Lambda})\boldsymbol{U}^T\boldsymbol{x}$$

$$= \boxed{\boldsymbol{U}}\ diag[h(\lambda_1), \dots, h(\lambda_n)]\ \boxed{\boldsymbol{U}^T\boldsymbol{x}}$$

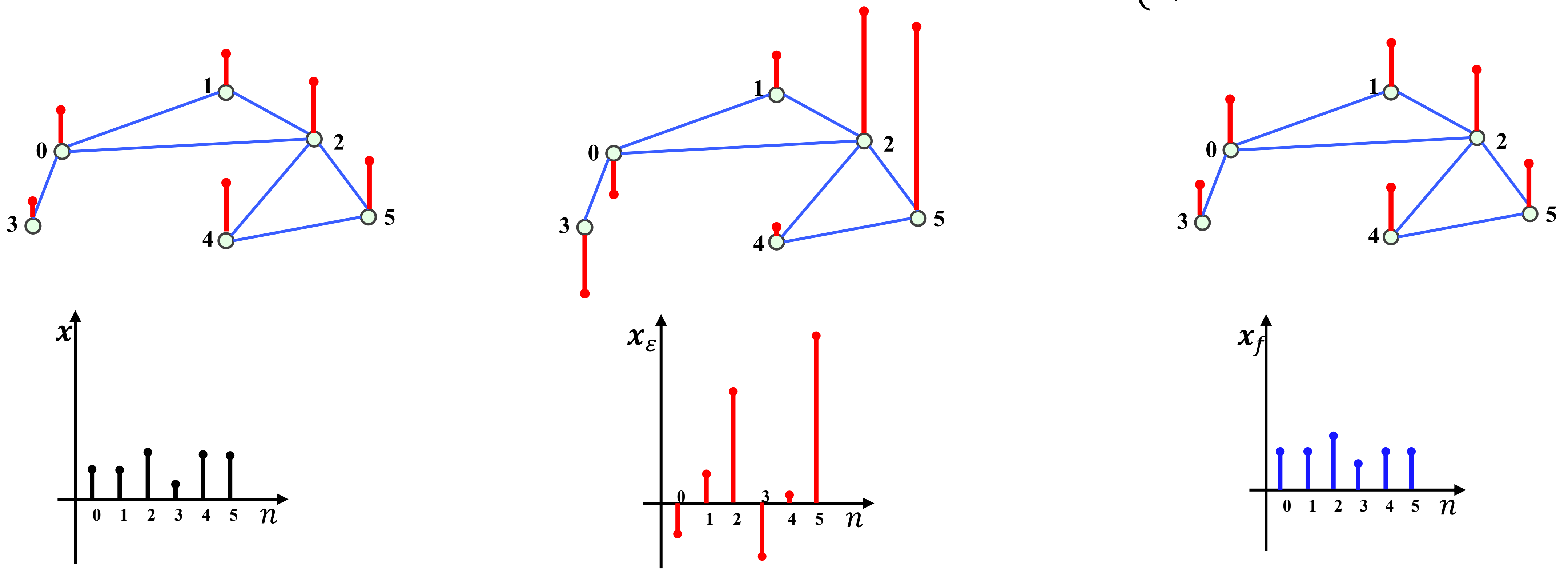
3.图傅里叶逆变换	⇐	2.频谱域滤波操作	⇐	1.图信号傅里叶变换
$\boldsymbol{U}h(\boldsymbol{\Lambda})\boldsymbol{U}^T\boldsymbol{x}$		$h(\boldsymbol{\Lambda})\boldsymbol{U}^T\boldsymbol{x}$		$\boldsymbol{U}^T\boldsymbol{x}$

- $h(\Lambda)/h(\lambda)$称为滤波器。

滤波器

- 基于拉普拉斯矩阵的低通滤波器

$$h(\lambda)=\begin{cases}1, & \text{for } \lambda<\lambda_c\\ 0, & \text{for other } \lambda.\end{cases}$$



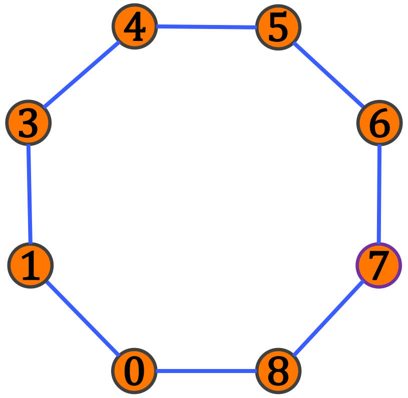
原始信号：
$\boldsymbol{x}=2\boldsymbol{u}_0+0.5\boldsymbol{u}_1$

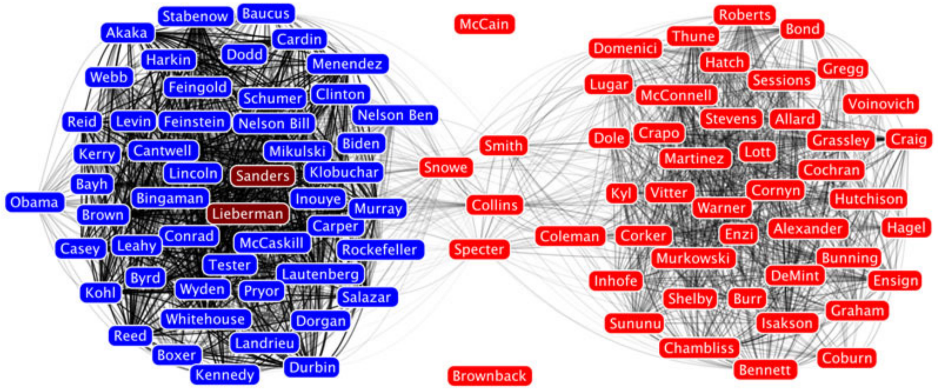
加了白高斯噪声的信号：
$\boldsymbol{x}_\varepsilon=\boldsymbol{x}+\boldsymbol{\varepsilon}$ ($SNR_{in}=-1.8\text{dB}$)

过滤后的信号：
$\lambda_c=0.35$ ($SNR_{out}=15.2\text{dB}$)

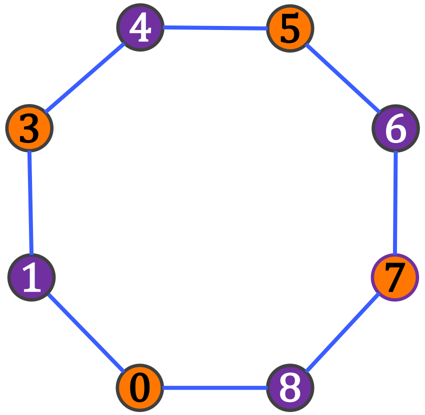
同/异配图

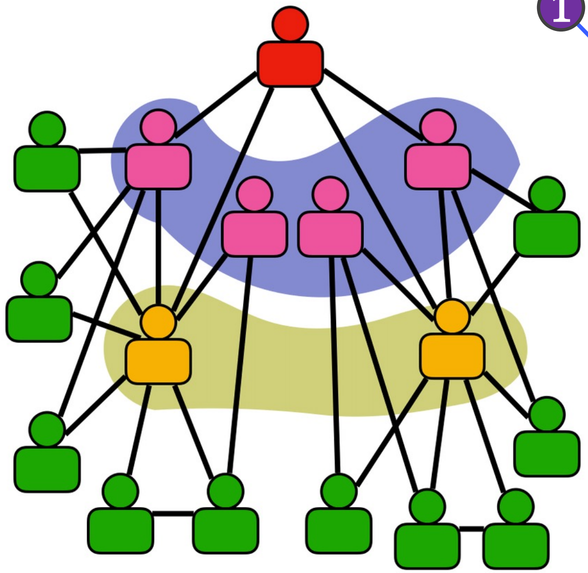
- 同配图(Homophily graph)：边相连的节点通常属于同一类(有相同标签)
- 异配图(Heterophily graph)：边相连的节点通常属于不同类(有不同标签)





Network of senators

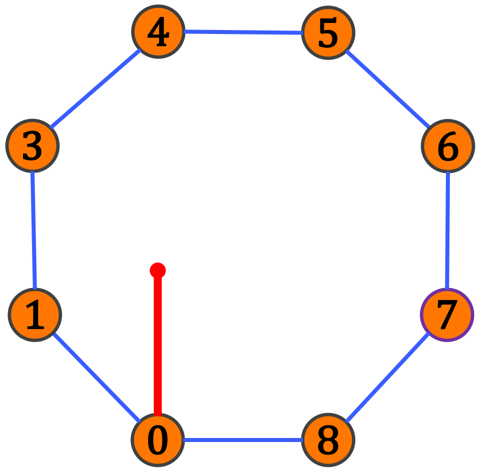




Network of managers

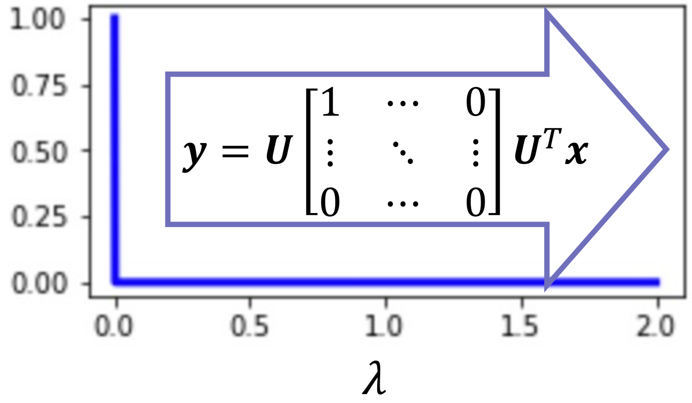
滤波器

低通滤波器→同配图；高通滤波器→异配图

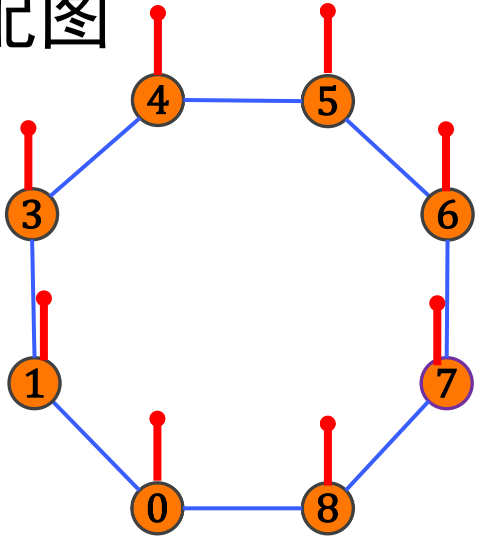


$x = [1.0,0,0,0,0,0,0,0,0]^T$

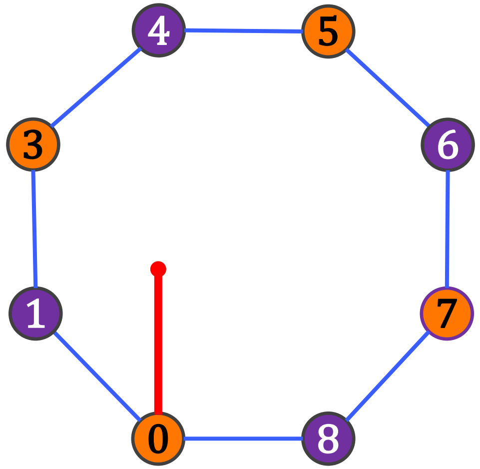
Impulse low-pass



λ

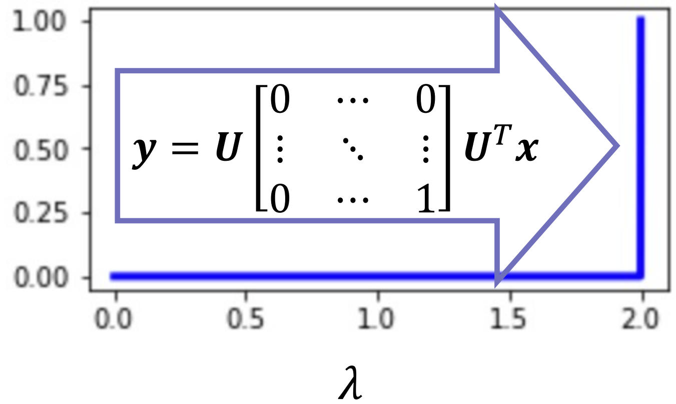


$y = [0.125, \ldots, 0.125]^T$

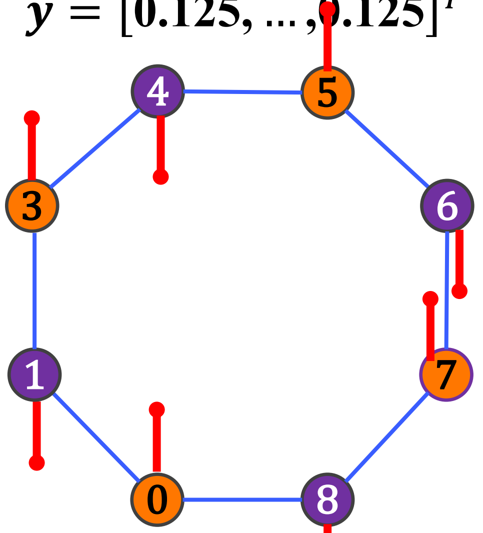


$x = [1.0,0,0,0,0,0,0,0,0]^T$

Impulse high-pass



λ



$y = [0.125, -0.125, \ldots, 0.125, -0.125]^T$

滤波器

- 设计不同的滤波器
  - 直接对滤波器$h(\lambda)$进行变换？

$$\boldsymbol{y} = \boldsymbol{U}\begin{pmatrix} h(\lambda_1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & h(\lambda_n) \end{pmatrix}\boldsymbol{U}^T\boldsymbol{x}$$

  - 特征值分解复杂度太高！
  - 当前流行的解决方案：利用多项式近似滤波器！

多项式滤波器

- 利用多项式近似滤波器

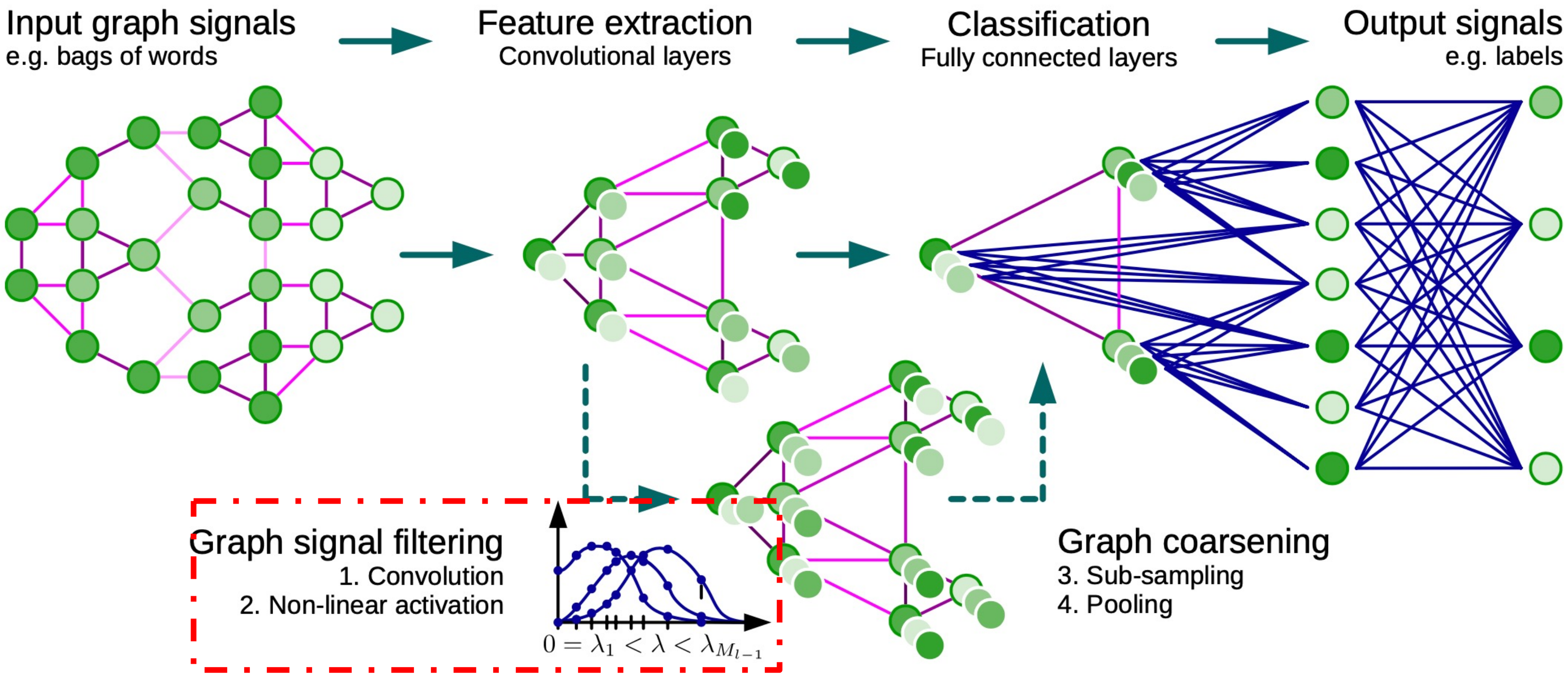
$$h(\lambda) \approx \sum_{k=0}^{K} w_k \lambda^k$$

- 对应的滤波操作

$$\boldsymbol{y} = \sum_{k=0}^{K} w_k \boldsymbol{L}^k \boldsymbol{x}$$

ChebNet

- ChebNet [Defferrard et al., 2016]用切比雪夫多项式近似滤波器
  - 模型结构：

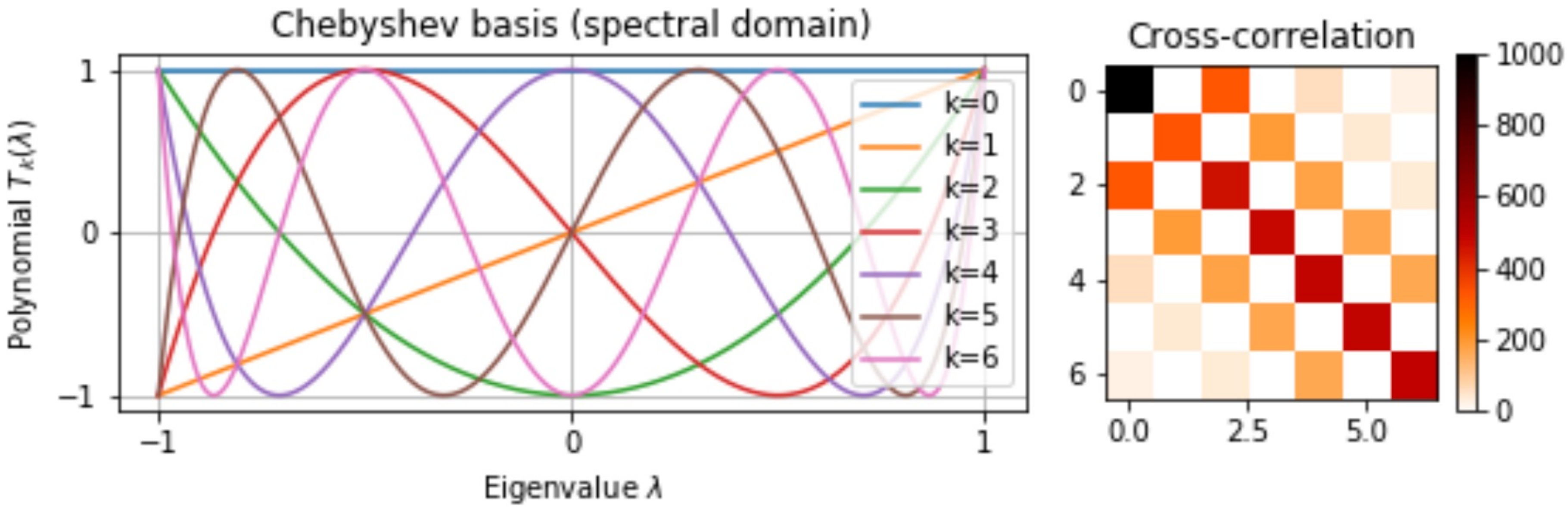


切比雪夫多项式近似

ChebNet

- Chebyshev多项式：k阶的Chebyshev多项式$T_k(x)$可以使用以下迭代式定义：

$$T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$$
$$\text{with } T_0(x) = 1, T_1(x) = x$$



$$T_0(x) = 1$$
$$T_1(x) = x$$
$$T_2(x) = 2x^2 - 1$$
$$T_3(x) = 4x^3 - 3x$$
$$T_4(x) = 8x^4 - 8x^2 + 1$$
$$T_5(x) = 16x^5 - 20x^3 + 5x$$
$$T_6(x) = 32x^6 - 48x^4 + 18x^2 - 1$$

ChebNet

- ChebNet [Defferrard et al., 2016]用切比雪夫多项式近似滤波器
  - 表达式：

$$H^{(\ell+1)} = \sigma\left(\sum_{k=0}^{K} T_k(\widehat{L})\, H^{(\ell)} W^{(\ell,k)}\right)$$

其中$\widehat{L} = 2L/\lambda_{max} - I$,$T_k(\widehat{L})$为切比雪夫多项式 $T_0(\widehat{L}) = I$, $T_1(\widehat{L}) = \widehat{L}, \dots, T_k(\widehat{L}) = 2T_{k-1}(\widehat{L}) - T_{k-2}(\widehat{L})$； W为可学习的权重，H为每一层的表示($H^{(0)} = X$)

ChebNet

- ChebNet [Defferrard et al., 2016]用切比雪夫多项式近似滤波器
  - 表达式：

$$\boldsymbol{H}^{(\ell+1)}=\sigma\left(\sum_{k=0}^{K} T_k(\widehat{\boldsymbol{L}})\,\boldsymbol{H}^{(\ell)}\boldsymbol{W}^{(\ell,k)}\right)$$

其中$\widehat{\boldsymbol{L}}=2\boldsymbol{L}/\lambda_{max}-\boldsymbol{I}$,$T_k(\widehat{\boldsymbol{L}})$为切比雪夫多项式 $T_0(\widehat{\boldsymbol{L}})=\boldsymbol{I}$, $T_1(\widehat{\boldsymbol{L}})=\widehat{\boldsymbol{L}},\ldots,T_k(\widehat{\boldsymbol{L}})=2T_{k-1}(\widehat{\boldsymbol{L}})-T_{k-2}(\widehat{\boldsymbol{L}})$； $\boldsymbol{W}$为可学习的权重，$\boldsymbol{H}$为每一层的表示($\boldsymbol{H}^{(0)}=\boldsymbol{X}$)

  - 对应的多项式滤波器操作 ($\lambda_{max}=2$)：

$$\boldsymbol{y}=\Big(\big(\theta_0\boldsymbol{I}+\theta_1(\boldsymbol{L}-\boldsymbol{I})+\theta_2(2\boldsymbol{L}-3\boldsymbol{I})+\cdots+\theta_k(2T_{k-1}(\widehat{\boldsymbol{L}})-T_{k-2}(\widehat{\boldsymbol{L}})\big)\Big)\boldsymbol{x}$$

$\theta_0,..,\theta_k$表示多项式参数

GCN

- GCN[Kipf et al.,2017]简化ChebNet，$\theta = \theta_0 = -\theta_1$
  - 则滤波器操作为：

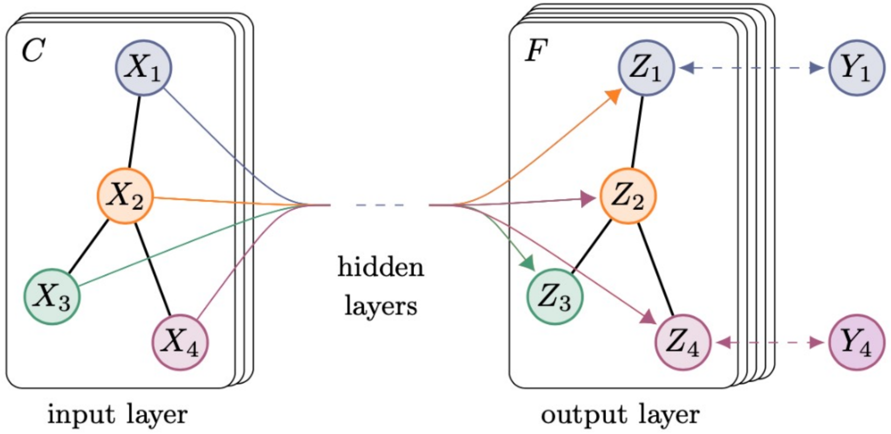
$$\begin{aligned} \boldsymbol{y} &= (\theta \boldsymbol{I} - \theta(\boldsymbol{L} - \boldsymbol{I}))\boldsymbol{x} \\ &= \theta(2\boldsymbol{I} - \boldsymbol{L})\boldsymbol{x} \\ &= \theta\left(\boxed{\boldsymbol{I} + \boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}}\right)\boldsymbol{x} \xrightarrow{\text{Renormalization trick}} \theta\left(\boxed{\widetilde{\boldsymbol{D}}^{-1/2}\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{D}}^{-1/2}}\right)\boldsymbol{x} \end{aligned}$$

GCN

- GCN[Kipf et al.,2017]简化ChebNet，$\theta = \theta_0 = -\theta_1$
  - 则滤波器操作为：

$$
\begin{aligned}
\boldsymbol{y} &= (\theta \boldsymbol{I} - \theta(\boldsymbol{L} - \boldsymbol{I}))\boldsymbol{x} \\
&= \theta(2\boldsymbol{I} - \boldsymbol{L})\boldsymbol{x} \\
&= \theta\left(\boxed{\boldsymbol{I} + \boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}}\right)\boldsymbol{x}
\end{aligned}
\quad \xrightarrow{\text{Renormalization trick}} \quad \theta\left(\boxed{\widetilde{\boldsymbol{D}}^{-1/2}\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{D}}^{-1/2}}\right)\boldsymbol{x}
$$

  - 对应的模型表达式(记$\widetilde{\boldsymbol{P}} = \widetilde{\boldsymbol{D}}^{-1/2}\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{D}}^{-1/2}$)



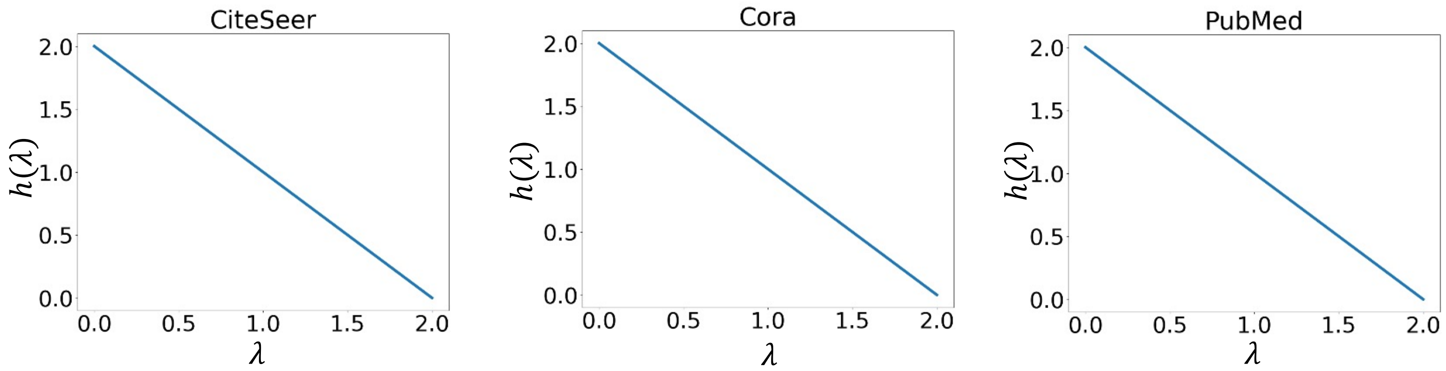
$$\boldsymbol{H}^{(\ell+1)} = \sigma\left(\widetilde{\boldsymbol{P}} \cdot \boldsymbol{H}^{(\ell)} \cdot \boldsymbol{W}^{(\ell)}\right)$$

$$\boldsymbol{Y} = \mathrm{softmax}\left(\widetilde{\boldsymbol{P}}\mathrm{ReLU}\left(\widetilde{\boldsymbol{P}}\boldsymbol{X}\boldsymbol{W}^{(0)}\right)\boldsymbol{W}^{(1)}\right)$$

GCN

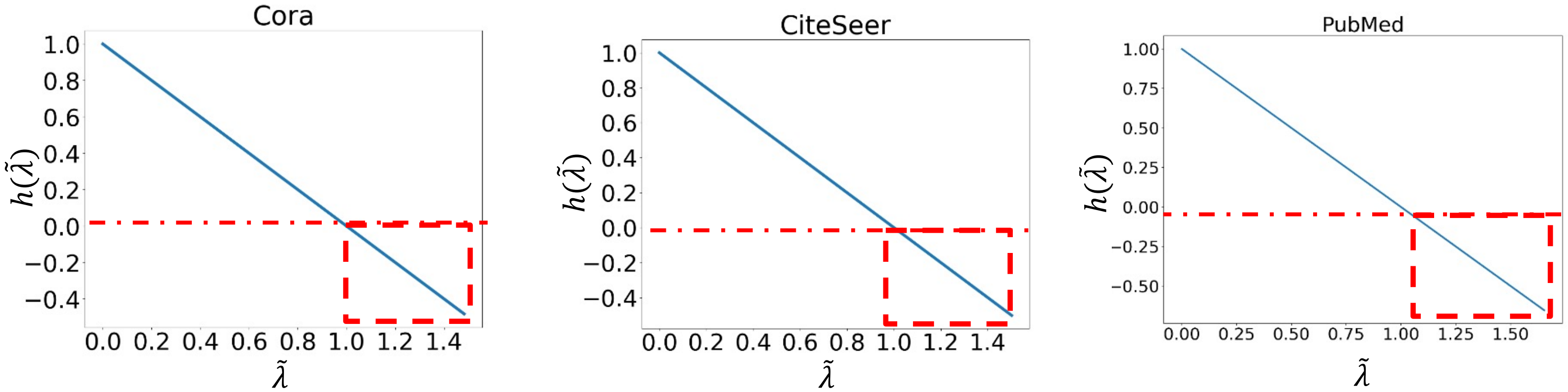
$$\boldsymbol{H}^{(\ell+1)} = \sigma\left(\widetilde{\boldsymbol{P}} \cdot \boldsymbol{H}^{(\ell)} \cdot \boldsymbol{W}^{(\ell)}\right)$$

- 单层GCN对应的滤波器（低通滤波器）
  - Before renormalization trick: $h(\boldsymbol{L}) = 2\boldsymbol{I} - \boldsymbol{L}$, $h(\lambda) = 2 - \lambda$



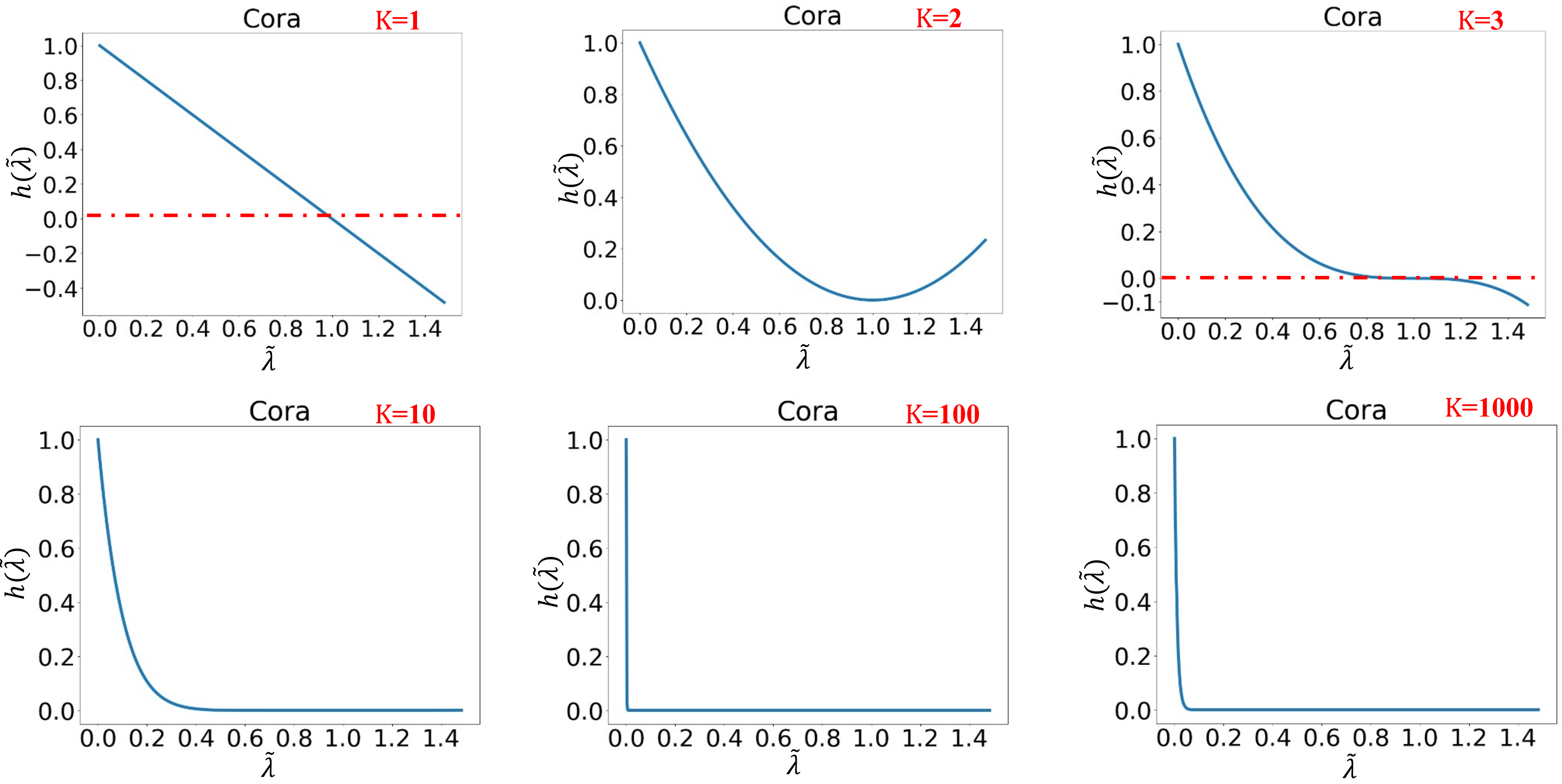
  - After renormalization trick: $h(\tilde{\boldsymbol{L}}) = \widetilde{\boldsymbol{P}} = \boldsymbol{I} - \tilde{\boldsymbol{L}}$, $h(\tilde{\lambda}) = 1 - \tilde{\lambda}$

$h(\tilde{\lambda}) < 0,$ for $\tilde{\lambda} > 1$

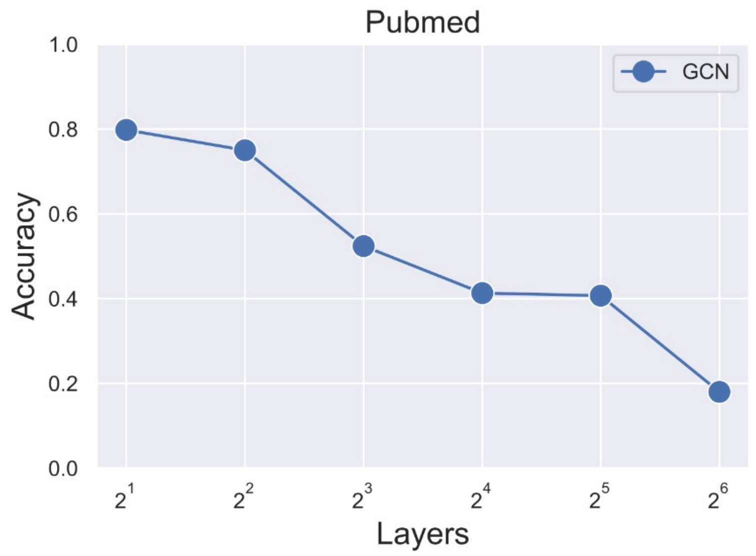


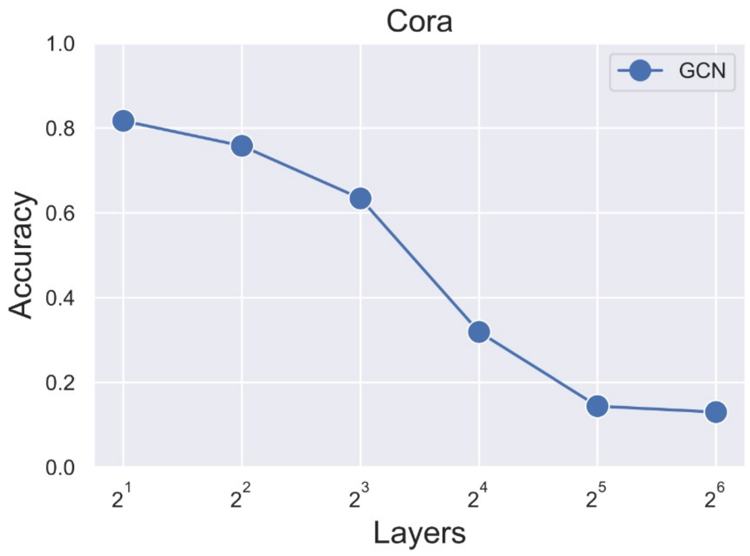
GCN

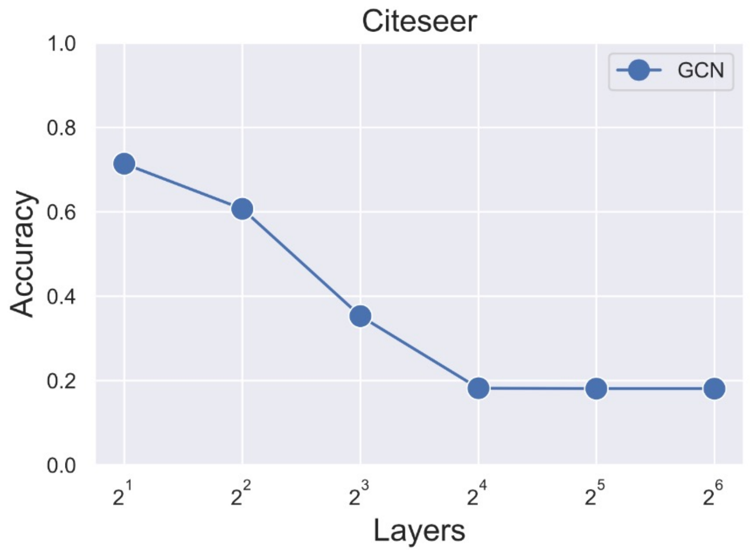
- K层GCN对应的滤波器 $h(\tilde{\lambda}) = (1-\tilde{\lambda})^K$ 当K增大导致**过平滑！**



GCN-过平滑

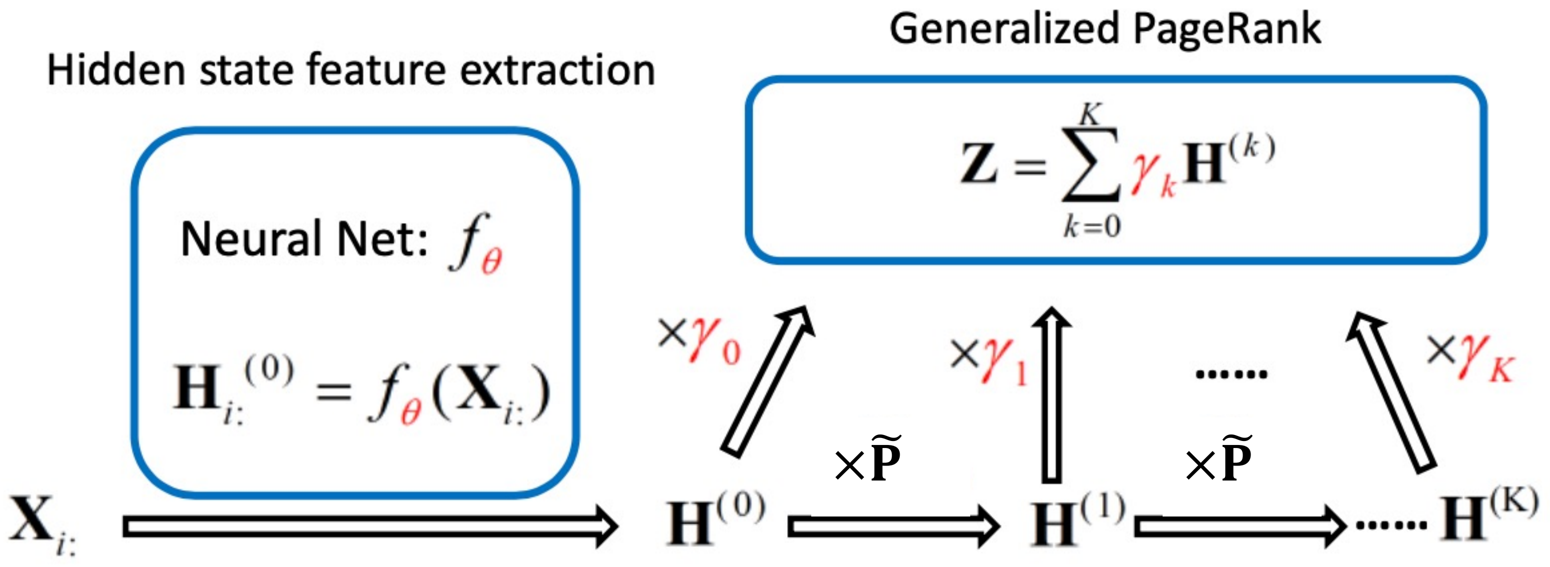

Pubmed
GCN
Accuracy
Layers
1.0
0.8
0.6
0.4
0.2
0.0
2^1 2^2 2^3 2^4 2^5 2^6


Cora
GCN
Accuracy
Layers
1.0
0.8
0.6
0.4
0.2
0.0
2^1 2^2 2^3 2^4 2^5 2^6


Citeseer
GCN
Accuracy
Layers
1.0
0.8
0.6
0.4
0.2
0.0
2^1 2^2 2^3 2^4 2^5 2^6

GPR-GNN

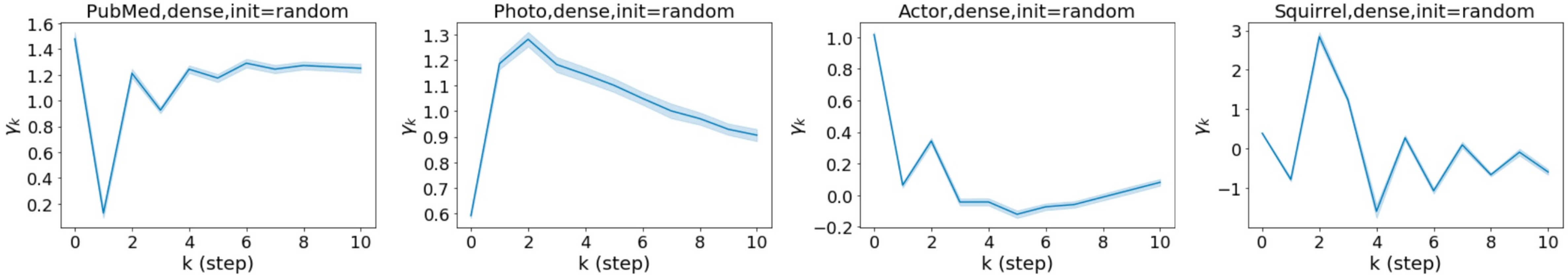
- GPR-GNN [Chien et al., 2021]学习多项式滤波器的系数：



- 表达式： $\boldsymbol{Z} = \sum_{k=0}^{K} \gamma_k \widetilde{\boldsymbol{P}}^k \boldsymbol{H}^{(0)}$
- 对应的滤波器操作： $\boldsymbol{y} = \sum_{k=0}^{K} \gamma_k \widetilde{\boldsymbol{P}}^k \boldsymbol{x}$

GPR-GNN

- GPR-GNN [Chien et al., 2021]： $\boldsymbol{Z}=\sum_{k=0}^{K}\gamma_k\widetilde{\boldsymbol{P}}^k\boldsymbol{H}^{(0)}$，在真实数据集上学习到的$\gamma_k$：

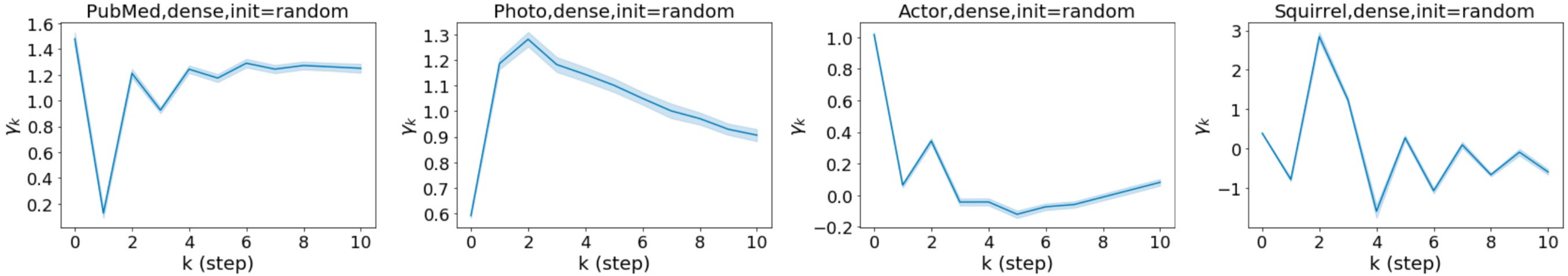


(a) PubMed (b) Photo (c) Actor (d) Squirrel

- 证明了 $\forall\gamma_k\geq 0$ 为low-pass滤波器， $\gamma_k=(-\alpha)^k,\alpha\in(0,1)$为high-pass滤波器。

GPR-GNN

- GPR-GNN [Chien et al., 2021]： $\boldsymbol{Z} = \sum_{k=0}^{K} \gamma_k \widetilde{\boldsymbol{P}}^k \boldsymbol{H}^{(0)}$，在真实数据集上学习到的$\gamma_k$：



(a) PubMed (b) Photo (c) Actor (d) Squirrel

- 证明了 $\forall \gamma_k \geq 0$ 为low-pass滤波器， $\gamma_k = (-\alpha)^k, \alpha \in (0,1)$为high-pass滤波器。
- 基于空域的多项式滤波器缺乏可解释性！不能解释学到的任意滤波器。

GPR-GNN

- 实验结果

Table 1: Benchmark dataset properties and statistics.

Dataset	Cora	Citeseer	PubMed	Computers	Photo	Chameleon	Squirrel	Actor	Texas	Cornell
Classes	7	6	5	10	8	5	5	5	5	5
Features	1433	3703	500	767	745	2325	2089	932	1703	1703
Nodes	2708	3327	19717	13752	7650	2277	5201	7600	183	183
Edges	5278	4552	44324	245861	119081	31371	198353	26659	279	277
$\mathcal{H}(G)$	0.825	0.718	0.792	0.802	0.849	0.247	0.217	0.215	0.057	0.301

	Cora	Citeseer	PubMed	Computers	Photo	Chameleon	Actor	Squirrel	Texas	Cornell
GPRGNN	**79.51±0.36**	67.63±0.38	**85.07±0.09**	**82.90±0.37**	**91.93±0.26**	**67.48±0.40**	**39.30±0.27**	**49.93±0.53**	**92.92±0.61**	**91.36±0.70**
APPNP	**79.41±0.38**	**68.59±0.30**	**85.02±0.09**	81.99±0.26	91.11±0.26	51.91±0.56	38.86±0.24	34.77±0.34	91.18±0.70	**91.80±0.63**
MLP	50.34±0.48	52.88±0.51	80.57±0.12	70.48±0.28	78.69±0.30	46.72±0.46	38.58±0.25	31.28±0.27	92.26±0.71	**91.36±0.70**
SGC	70.81±0.67	58.98±0.47	82.09±0.11	76.27±0.36	83.80±0.46	63.02±0.43	29.39±0.20	43.14±0.28	55.18±1.17	47.80±1.50
GCN	75.21±0.38	67.30±0.35	84.27±0.01	82.52±0.32	90.54±0.21	60.96±0.78	30.59±0.23	45.66±0.39	75.16±0.96	66.72±1.37
GAT	76.70±0.42	67.20±0.46	83.28±0.12	81.95±0.38	90.09±0.27	63.9±0.46	35.98±0.23	42.72±0.33	78.87±0.86	76.00±1.01
SAGE	70.89±0.54	61.52±0.44	81.30±0.10	**83.11±0.23**	90.51±0.25	62.15±0.42	36.37±0.21	41.26±0.26	79.03±1.20	71.41±1.24
JKNet	73.22±0.64	60.85±0.76	82.91±0.11	77.80±0.97	87.70±0.70	62.92±0.49	33.41±0.25	44.72±0.48	75.53±1.16	66.73±1.73
GCN-Cheby	71.39±0.51	65.67±0.38	83.83±0.12	82.41±0.28	90.09±0.28	59.96±0.51	38.02±0.23	40.67±0.31	86.08±0.96	85.33±1.04
GeomGCN	20.37±1.13	20.30±0.90	58.20±1.23	NA	NA	61.06±0.49	31.81±0.24	38.28±0.27	58.56±1.77	55.59±1.59

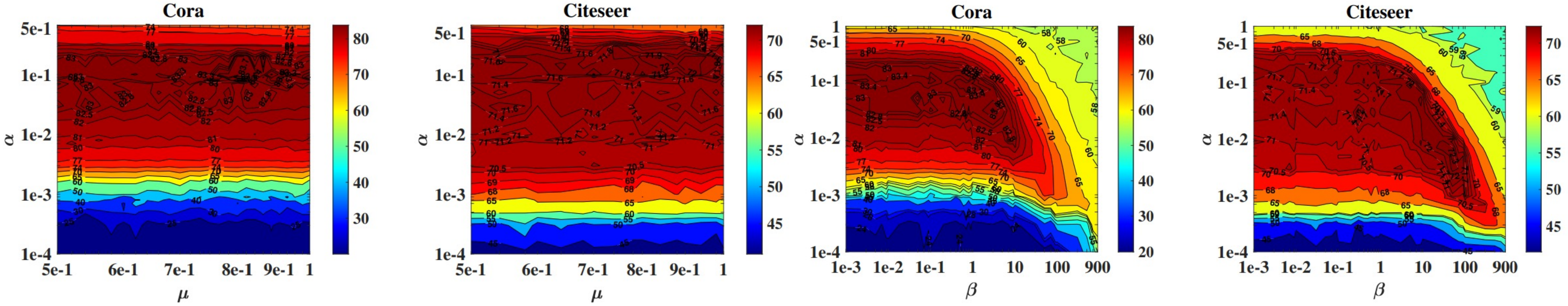
GNN-LF/HF

- GNN-LF/HF[Zhu et al.,2021]从优化函数的角度设计滤波器

$$\boldsymbol{Z}=\left((\mu+1/\alpha-1)\boldsymbol{I}+(2-\mu-1/\alpha)\widetilde{\boldsymbol{P}}\right)^{-1}\left(\mu\boldsymbol{I}+(1-\mu)\widetilde{\boldsymbol{P}}\right)\boldsymbol{H}$$ (GNN-LF对应低通滤波器)

$$\boldsymbol{Z}=\left((\beta+1/\alpha)\boldsymbol{I}+(1-\beta-1/\alpha)\widetilde{\boldsymbol{P}}\right)^{-1}\left(\boldsymbol{I}+\beta\widetilde{\boldsymbol{L}}\right)\boldsymbol{H}$$ (GNN−HF对应高通滤波器)

- 基于空域设计的多项式滤波器较复杂



- □ 参数过于复杂，难以确定合适的滤波器。

第三部分将进一步讲解

问题

- 通过前面的分析：
  - □ GCN使用的滤波器会出现负数，并且多层GCN会出现过平滑，其使用的滤波器是ill-posed；ChebNet可能学到负的滤波器；
  - □ GPR-GNN、GNN-LF/GNN-HF等从空域上通过学习或设计多项式的系数试图得到滤波器，不一定是最优选择。
- 因此：

是否存在多项式滤波器，可以学习/设计任意

Valid谱域滤波器？

BernNet: Learning Arbitrary Graph Spectral Filters via Bernstein Approximation (NeurIPS, 2021)

Mingguo He, Zhewei Wei, Zengfeng Huang, Hongteng Xu

研究背景

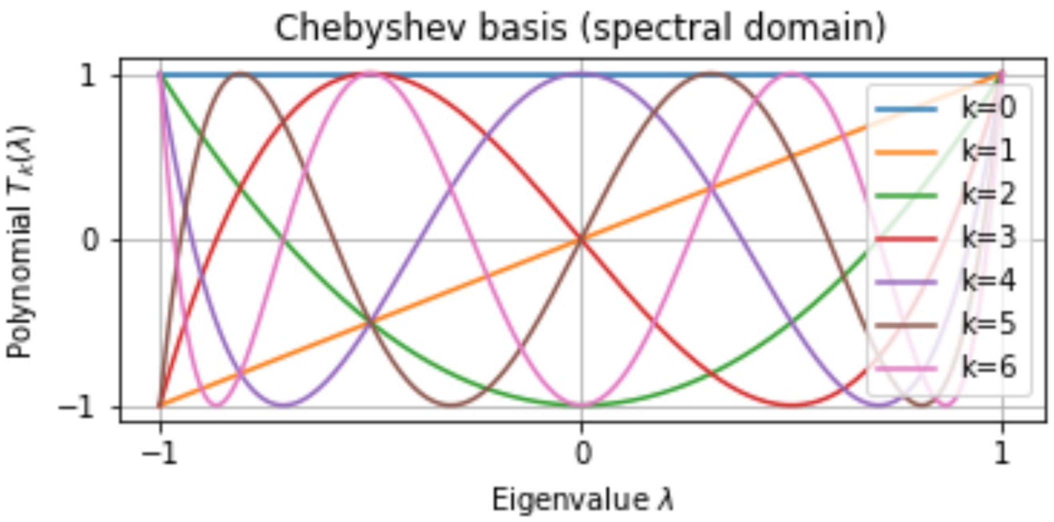
- Valid的多项式滤波器应该满足：

$$h(\lambda) = \sum_{k=0}^{K} w_k \lambda^k \geq 0, \forall \lambda \in [0,2]$$

  - □ 这个条件看起来很简单，但是现有一些GNN却不满足，如GCN。

- Chebyshev多项式近似不容易满足该条件：
  - □ 需要$h(\lambda) = \sum_{k=0}^{K} w_k T_k(\lambda) \geq 0$；
  - □ 不能简单地对w_k进行限制，

  因为$T_k(\lambda)$存在负值域区间



He M, Wei Z et al. BernNet: Learning Arbitrary Graph Spectral Filters via Bernstein Approximation. [NeurIPS’2021]

研究思路

- Valid的多项式滤波器应该满足：

$$h(\lambda)=\sum_{k=0}^{K} w_k \lambda^k \geq 0, \forall \lambda \in [0,2]$$

  - 这个条件看起来很简单，但是现有一些GNN却不满足，如GCN。

- 在[0,2]上非负的任意多项式可以写成Bernstein多项式的形式：

$$h(\lambda)=\sum_{k=0}^{K} \boxed{\theta_k} \boxed{\frac{1}{2^k}\binom{K}{k}(2-\lambda)^{K-k}\lambda^k}$$

Bernstein多项式系数，需满足$\theta_k \geq 0$

Bernstein基，可以近似任意多项式，任意基≥ 0

He M, Wei Z et al. BernNet: Learning Arbitrary Graph Spectral Filters via Bernstein Approximation. [NeurIPS'2021]

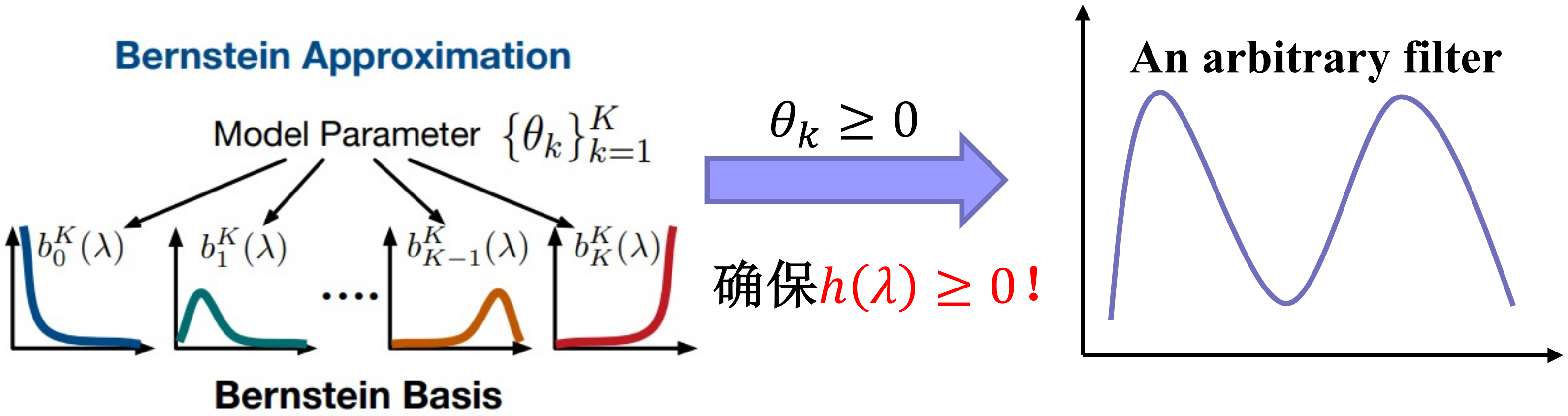
研究思路

- Valid的滤波器应该满足：

$$h(\lambda) \geq 0, \forall\lambda \in [0,2]$$

  - □ 这个条件看起来很简单，但是现有一些GNN却不满足，如GCN。

- 利用Bernstein多项式近似滤波器：

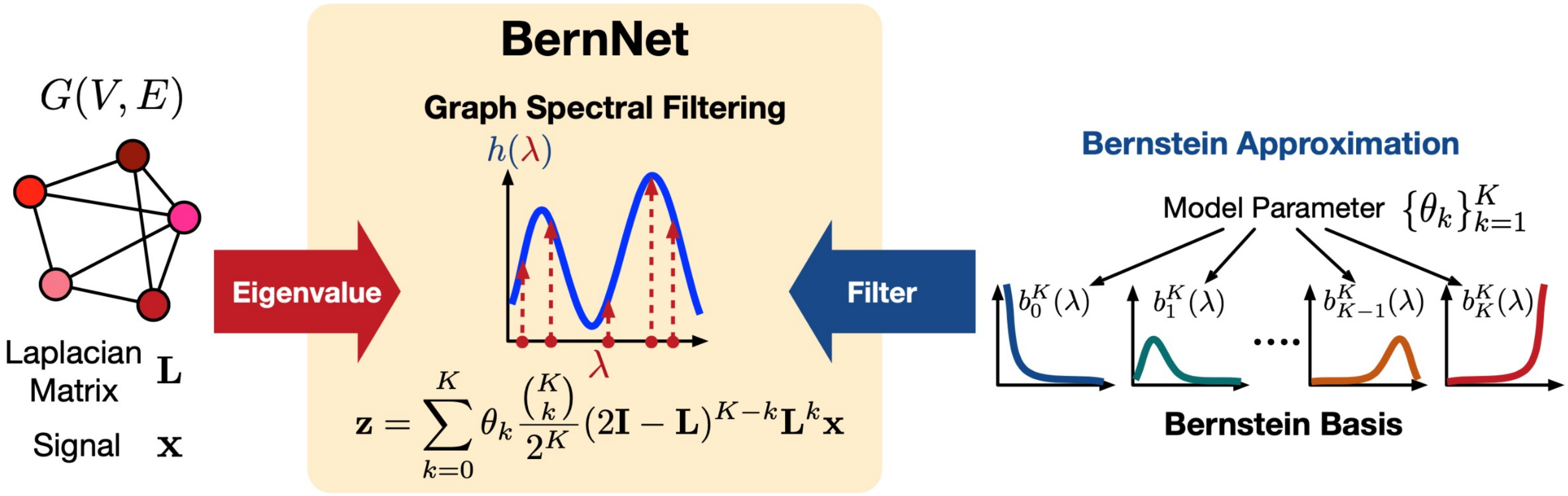


He M, Wei Z et al. BernNet: Learning Arbitrary Graph Spectral Filters via Bernstein Approximation. [NeurIPS’2021]

BernNet

- 利用Bernstein多项式近似滤波器

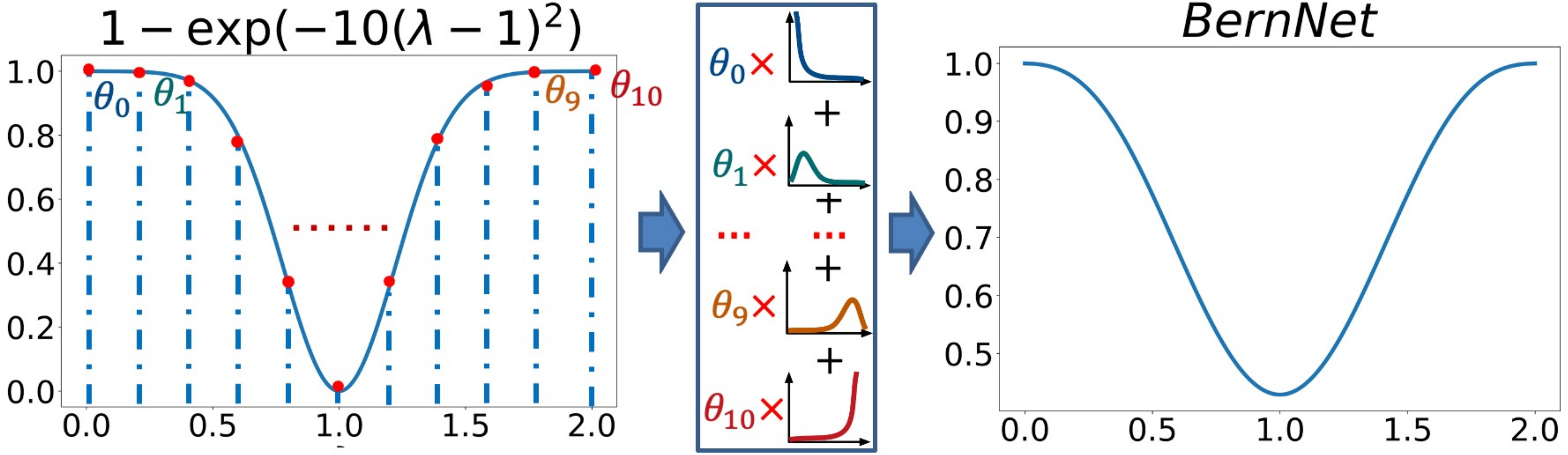
$$\boldsymbol{z} = \left(\sum_{k=0}^{K} \theta_k \frac{1}{2^k} \binom{K}{k} (2\boldsymbol{I} - \boldsymbol{L})^{K-k} \boldsymbol{L}^k \right) \boldsymbol{x}$$



He M, Wei Z et al. BernNet: Learning Arbitrary Graph Spectral Filters via Bernstein Approximation. [NeurIPS'2021]

Bernstein Approximation

- 系数$\theta_k = h(\frac{2k}{K})$对应了滤波器$h(\lambda)$在$\frac{2k}{K}$处的值，可解释性强！



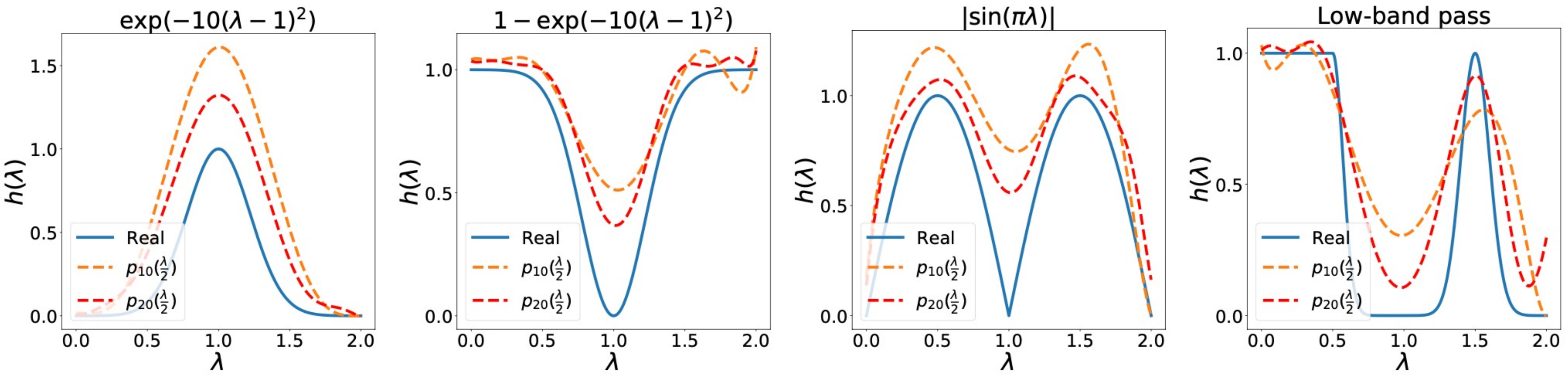
$$\boldsymbol{z} = \left(\sum_{k=0}^{K} \theta_k \frac{1}{2^k}\binom{K}{k}(2\boldsymbol{I}-\boldsymbol{L})^{K-k}\boldsymbol{L}^k\right)\boldsymbol{x} \rightarrow h(\boldsymbol{L})\boldsymbol{x}, k \rightarrow \infty$$

He M, Wei Z et al. BernNet: Learning Arbitrary Graph Spectral Filters via Bernstein Approximation. [NeurIPS'2021]

BernNet

- 学习任意的滤波器，随着K增大，效果越好

$$\boldsymbol{z} = \sum_{k=0}^{K} ReLU(\theta_k) \frac{1}{2^k} \binom{K}{k} (2\boldsymbol{I} - \boldsymbol{L})^{K-k} \boldsymbol{L}^k$$

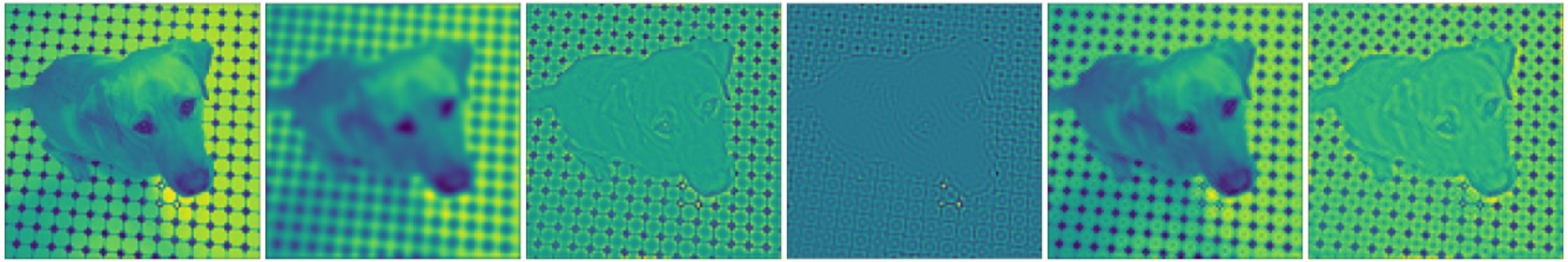


(a) Band-pass (b) Band-rejection (c) Comb (d) Low-band-pass

He M, Wei Z et al. BernNet: Learning Arbitrary Graph Spectral Filters via Bernstein Approximation. [NeurIPS'2021]

BernNet：实验结果

- 学习图像数据+人工过滤器



(a) Original (b) Low-pass (c) High-pass (d) Band-pass (e) Band-rejection (f) Comb

Figure 3: Input image and the filtering results.

- 学习滤波器（误差/ R^2得分）

	Low-pass	High-pass	Band-pass	Band-rejection	Comb
	$\exp(-10\lambda^2)$	$1-\exp(-10\lambda^2)$	$\exp(-10(\lambda-1)^2)$	$1-\exp(-10(\lambda-1)^2)$	$\lvert\sin(\pi\lambda)\rvert$
GCN	3.4799(.9872)	67.6635(.2364)	25.8755(.1148)	21.0747(.9438)	50.5120(.2977)
GAT	2.3574(.9905)	21.9618(.7529)	14.4326(.4823)	12.6384(.9652)	23.1813(.6957)
GPR-GNN	0.4169(.9984)	0.0943(.9986)	3.5121(.8551)	3.7917(.9905)	4.6549(.9311)
ARMA	1.8478(.9932)	1.8632(.9793)	7.6922(.7098)	8.2732(.9782)	15.1214(.7975)
ChebyNet	0.8220(.9973)	0.7867(.9903)	2.2722(.9104)	2.5296(.9934)	4.0735(.9447)
BernNet	**0.0314(.9999)**	**0.0113(.9999)**	**0.0411(.9984)**	**0.9313(.9973)**	**0.9982(.9868)**

He M, Wei Z et al. BernNet: Learning Arbitrary Graph Spectral Filters via Bernstein Approximation. [NeurIPS’2021]

BernNet：实验结果

- 真实数据集节点分类任务

Table 3: Dataset statistics.

	Cora	CiteSeer	PubMed	Computers	Photo	Chameleon	Squirrel	Actor	Texas	Cornell
Nodes	2708	3327	19717	13752	7650	2277	5201	7600	183	183
Edges	5278	4552	44324	245861	119081	31371	198353	26659	279	277
Features	1433	3703	500	767	745	2325	2089	932	1703	1703
Classes	7	6	5	10	8	5	5	5	5	5

- 真实数据集节点分类（精确度）

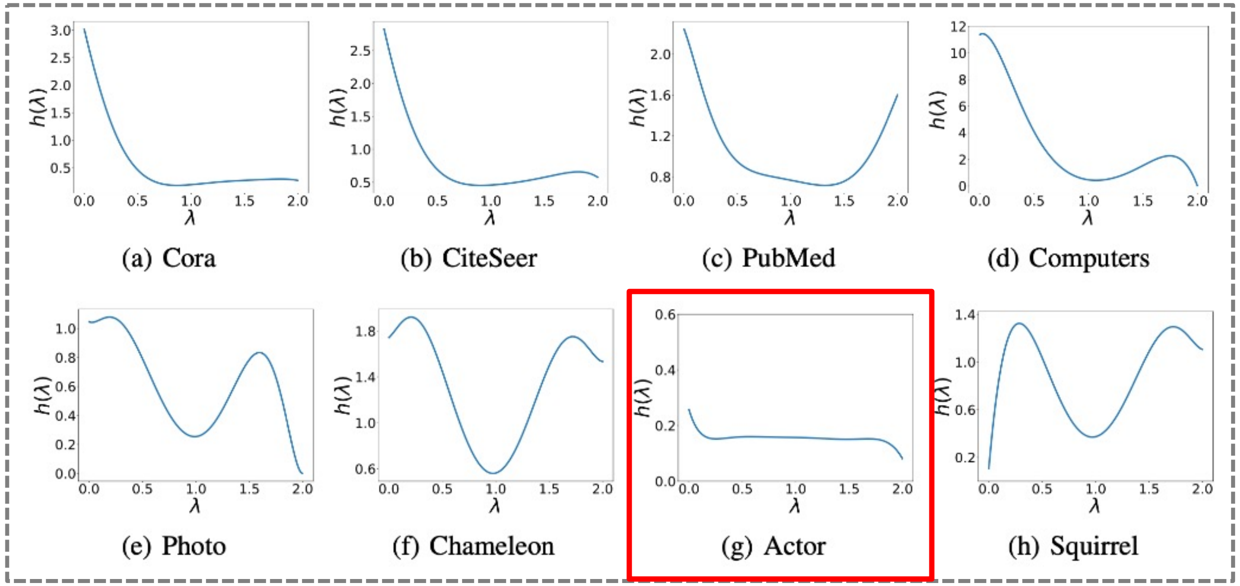
	GCN	GAT	APPNP	MLP	ChebyNet	GPR-GNN	BernNet
Cora	$87.14_{\pm 1.01}$	$88.03_{\pm 0.79}$	$88.14_{\pm 0.73}$	$76.96_{\pm 0.95}$	$86.67_{\pm 0.82}$	$\mathbf{88.57_{\pm 0.69}}$	$88.52_{\pm 0.95}$
CiteSeer	$79.86_{\pm 0.67}$	$\mathbf{80.52_{\pm 0.71}}$	$80.47_{\pm 0.74}$	$76.58_{\pm 0.88}$	$79.11_{\pm 0.75}$	$80.12_{\pm 0.83}$	$80.09_{\pm 0.79}$
PubMed	$86.74_{\pm 0.27}$	$87.04_{\pm 0.24}$	$88.12_{\pm 0.31}$	$85.94_{\pm 0.22}$	$87.95_{\pm 0.28}$	$88.46_{\pm 0.33}$	$\mathbf{88.48_{\pm 0.41}}$
Computers	$83.32_{\pm 0.33}$	$83.32_{\pm 0.39}$	$85.32_{\pm 0.37}$	$82.85_{\pm 0.38}$	$87.54_{\pm 0.43}$	$86.85_{\pm 0.25}$	$\mathbf{87.64_{\pm 0.44}}$
Photo	$88.26_{\pm 0.73}$	$90.94_{\pm 0.68}$	$88.51_{\pm 0.31}$	$84.72_{\pm 0.34}$	$93.77_{\pm 0.32}$	$\mathbf{93.85_{\pm 0.28}}$	$93.63_{\pm 0.35}$
Chameleon	$59.61_{\pm 2.21}$	$63.13_{\pm 1.93}$	$51.84_{\pm 1.82}$	$46.85_{\pm 1.51}$	$59.28_{\pm 1.25}$	$67.28_{\pm 1.09}$	$\mathbf{68.29_{\pm 1.58}}$
Actor	$33.23_{\pm 1.16}$	$33.93_{\pm 2.47}$	$39.66_{\pm 0.55}$	$40.19_{\pm 0.56}$	$37.61_{\pm 0.89}$	$39.92_{\pm 0.67}$	$\mathbf{41.79_{\pm 1.01}}$
Squirrel	$46.78_{\pm 0.87}$	$44.49_{\pm 0.88}$	$34.71_{\pm 0.57}$	$31.03_{\pm 1.18}$	$40.55_{\pm 0.42}$	$50.15_{\pm 1.92}$	$\mathbf{51.35_{\pm 0.73}}$
Texas	$77.38_{\pm 3.28}$	$80.82_{\pm 2.13}$	$90.98_{\pm 1.64}$	$91.45_{\pm 1.14}$	$86.22_{\pm 2.45}$	$92.95_{\pm 1.31}$	$\mathbf{93.12_{\pm 0.65}}$
Cornell	$65.90_{\pm 4.43}$	$78.21_{\pm 2.95}$	$91.81_{\pm 1.96}$	$90.82_{\pm 1.63}$	$83.93_{\pm 2.13}$	$91.37_{\pm 1.81}$	$\mathbf{92.13_{\pm 1.64}}$

He M, Wei Z et al. BernNet: Learning Arbitrary Graph Spectral Filters via Bernstein Approximation. [NeurIPS'2021]

BernNet：实验结果

- 学习滤波器

	GCN	GAT	APPNP	MLP	ChebyNet	GPR-GNN	BernNet
Cora	87.14±1.01	88.03±0.79	88.14±0.73	76.96±0.95	86.67±0.82	**88.57±0.69**	88.52±0.95
CiteSeer	79.86±0.67	**80.52±0.71**	80.47±0.74	76.58±0.88	79.11±0.75	80.12±0.83	80.09±0.79
PubMed	86.74±0.27	87.04±0.24	88.12±0.31	85.94±0.22	87.95±0.28	88.46±0.33	**88.48±0.41**
Computers	83.32±0.33	83.32±0.39	85.32±0.37	82.85±0.38	87.54±0.43	86.85±0.25	**87.64±0.44**
Photo	88.26±0.73	90.94±0.68	88.51±0.31	84.72±0.34	93.77±0.32	**93.85±0.28**	93.63±0.35
Chameleon	59.61±2.21	63.13±1.93	51.84±1.82	46.85±1.51	59.28±1.25	67.28±1.09	**68.29±1.58**
Actor	33.23±1.16	33.93±2.47	39.66±0.55	40.19±0.56	37.61±0.89	39.92±0.67	**41.79±1.01**
Squirrel	46.78±0.87	44.49±0.88	34.71±0.57	31.03±1.18	40.55±0.42	50.15±1.92	**51.35±0.73**
Texas	77.38±3.28	80.82±2.13	90.98±1.64	91.45±1.14	86.22±2.45	92.95±1.31	**93.12±0.65**
Cornell	65.90±4.43	78.21±2.95	91.81±1.96	90.82±1.63	83.93±2.13	91.37±1.81	**92.13±1.64**

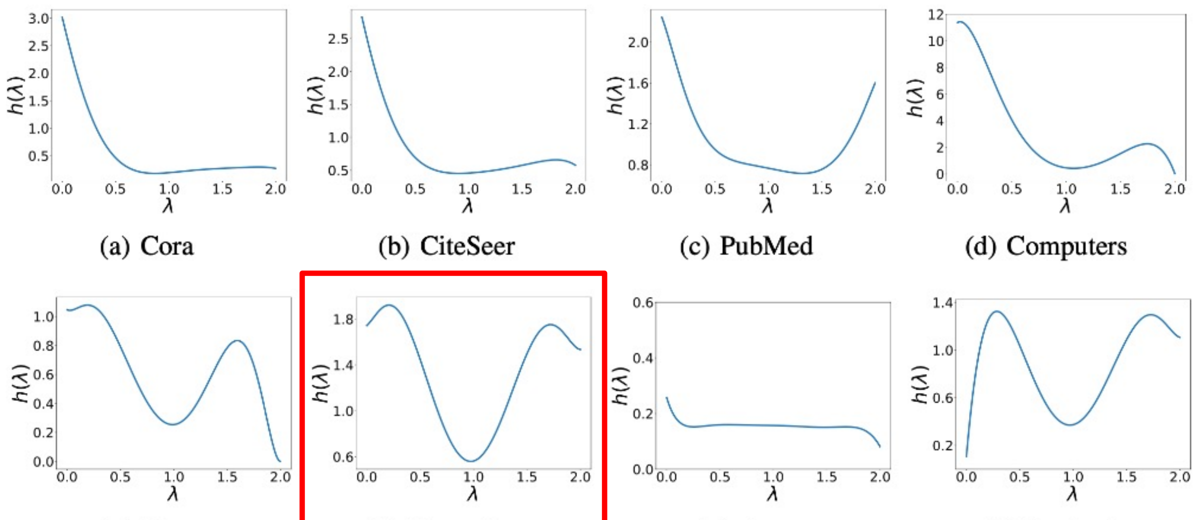


He M, Wei Z et al. BernNet: Learning Arbitrary Graph Spectral Filters via Bernstein Approximation. [NeurIPS'2021]

BernNet：实验结果

- 学习滤波器

	GCN	GAT	APPNP	MLP	ChebyNet	GPR-GNN	BernNet
Cora	87.14$_{\pm 1.01}$	88.03$_{\pm 0.79}$	88.14$_{\pm 0.73}$	76.96$_{\pm 0.95}$	86.67$_{\pm 0.82}$	**88.57$_{\pm 0.69}$**	88.52$_{\pm 0.95}$
CiteSeer	79.86$_{\pm 0.67}$	**80.52$_{\pm 0.71}$**	80.47$_{\pm 0.74}$	76.58$_{\pm 0.88}$	79.11$_{\pm 0.75}$	80.12$_{\pm 0.83}$	80.09$_{\pm 0.79}$
PubMed	86.74$_{\pm 0.27}$	87.04$_{\pm 0.24}$	88.12$_{\pm 0.31}$	85.94$_{\pm 0.22}$	87.95$_{\pm 0.28}$	88.46$_{\pm 0.33}$	**88.48$_{\pm 0.41}$**
Computers	83.32$_{\pm 0.33}$	83.32$_{\pm 0.39}$	85.32$_{\pm 0.37}$	82.85$_{\pm 0.38}$	87.54$_{\pm 0.43}$	86.85$_{\pm 0.25}$	**87.64$_{\pm 0.44}$**
Photo	88.26$_{\pm 0.73}$	90.94$_{\pm 0.68}$	88.51$_{\pm 0.31}$	84.72$_{\pm 0.34}$	93.77$_{\pm 0.32}$	**93.85$_{\pm 0.28}$**	93.63$_{\pm 0.35}$
Chameleon	59.61$_{\pm 2.21}$	63.13$_{\pm 1.93}$	51.84$_{\pm 1.82}$	46.85$_{\pm 1.51}$	59.28$_{\pm 1.25}$	67.28$_{\pm 1.09}$	**68.29$_{\pm 1.58}$**
Actor	33.23$_{\pm 1.16}$	33.93$_{\pm 2.47}$	39.66$_{\pm 0.55}$	40.19$_{\pm 0.56}$	37.61$_{\pm 0.89}$	39.92$_{\pm 0.67}$	**41.79$_{\pm 1.01}$**
Squirrel	46.78$_{\pm 0.87}$	44.49$_{\pm 0.88}$	34.71$_{\pm 0.57}$	31.03$_{\pm 1.18}$	40.55$_{\pm 0.42}$	50.15$_{\pm 1.92}$	**51.35$_{\pm 0.73}$**
Texas	77.38$_{\pm 3.28}$	80.82$_{\pm 2.13}$	90.98$_{\pm 1.64}$	91.45$_{\pm 1.14}$	86.22$_{\pm 2.45}$	92.95$_{\pm 1.31}$	**93.12$_{\pm 0.65}$**
Cornell	65.90$_{\pm 4.43}$	78.21$_{\pm 2.95}$	91.81$_{\pm 1.96}$	90.82$_{\pm 1.63}$	83.93$_{\pm 2.13}$	91.37$_{\pm 1.81}$	**92.13$_{\pm 1.64}$**



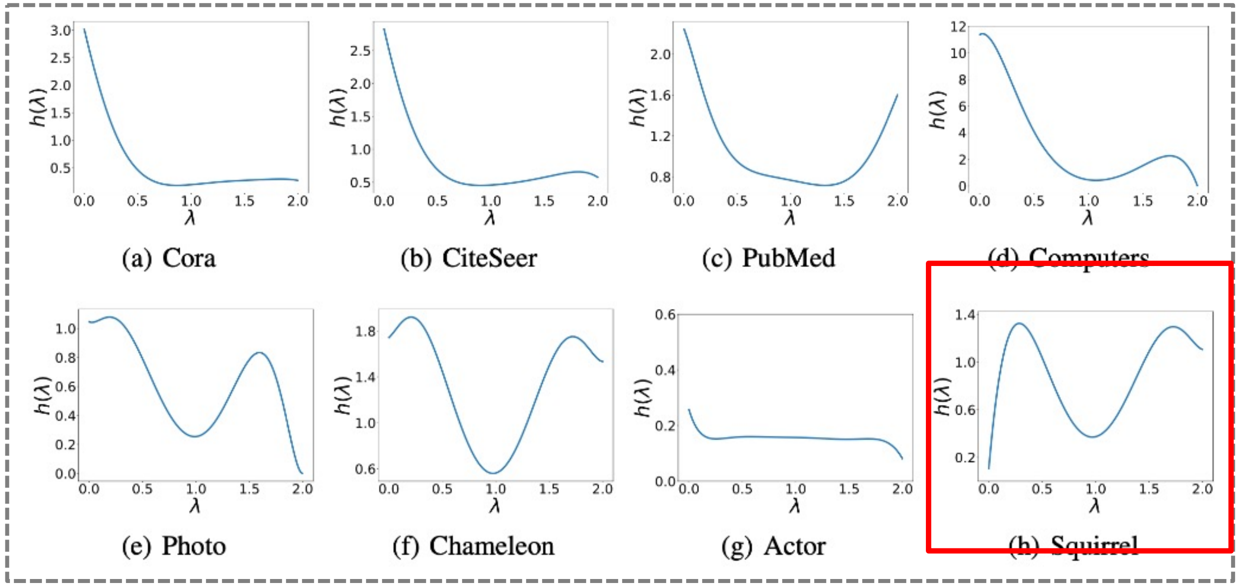
(a) Cora (b) CiteSeer (c) PubMed (d) Computers
(e) Photo (f) Chameleon (g) Actor (h) Squirrel

He M, Wei Z et al. BernNet: Learning Arbitrary Graph Spectral Filters via Bernstein Approximation. [NeurIPS’2021]

BernNet：实验结果

- 学习滤波器

	GCN	GAT	APPNP	MLP	ChebyNet	GPR-GNN	BernNet
Cora	87.14$_{\pm 1.01}$	88.03$_{\pm 0.79}$	88.14$_{\pm 0.73}$	76.96$_{\pm 0.95}$	86.67$_{\pm 0.82}$	**88.57$_{\pm 0.69}$**	88.52$_{\pm 0.95}$
CiteSeer	79.86$_{\pm 0.67}$	**80.52$_{\pm 0.71}$**	80.47$_{\pm 0.74}$	76.58$_{\pm 0.88}$	79.11$_{\pm 0.75}$	80.12$_{\pm 0.83}$	80.09$_{\pm 0.79}$
PubMed	86.74$_{\pm 0.27}$	87.04$_{\pm 0.24}$	88.12$_{\pm 0.31}$	85.94$_{\pm 0.22}$	87.95$_{\pm 0.28}$	88.46$_{\pm 0.33}$	**88.48$_{\pm 0.41}$**
Computers	83.32$_{\pm 0.33}$	83.32$_{\pm 0.39}$	85.32$_{\pm 0.37}$	82.85$_{\pm 0.38}$	87.54$_{\pm 0.43}$	86.85$_{\pm 0.25}$	**87.64$_{\pm 0.44}$**
Photo	88.26$_{\pm 0.73}$	90.94$_{\pm 0.68}$	88.51$_{\pm 0.31}$	84.72$_{\pm 0.34}$	93.77$_{\pm 0.32}$	**93.85$_{\pm 0.28}$**	93.63$_{\pm 0.35}$
Chameleon	59.61$_{\pm 2.21}$	63.13$_{\pm 1.93}$	51.84$_{\pm 1.82}$	46.85$_{\pm 1.51}$	59.28$_{\pm 1.25}$	67.28$_{\pm 1.09}$	**68.29$_{\pm 1.58}$**
Actor	33.23$_{\pm 1.16}$	33.93$_{\pm 2.47}$	39.66$_{\pm 0.55}$	40.19$_{\pm 0.56}$	37.61$_{\pm 0.89}$	39.92$_{\pm 0.67}$	**41.79$_{\pm 1.01}$**
Squirrel	46.78$_{\pm 0.87}$	44.49$_{\pm 0.88}$	34.71$_{\pm 0.57}$	31.03$_{\pm 1.18}$	40.55$_{\pm 0.42}$	50.15$_{\pm 1.92}$	**51.35$_{\pm 0.73}$**
Texas	77.38$_{\pm 3.28}$	80.82$_{\pm 2.13}$	90.98$_{\pm 1.64}$	91.45$_{\pm 1.14}$	86.22$_{\pm 2.45}$	92.95$_{\pm 1.31}$	**93.12$_{\pm 0.65}$**
Cornell	65.90$_{\pm 4.43}$	78.21$_{\pm 2.95}$	91.81$_{\pm 1.96}$	90.82$_{\pm 1.63}$	83.93$_{\pm 2.13}$	91.37$_{\pm 1.81}$	**92.13$_{\pm 1.64}$**



(a) Cora (b) CiteSeer (c) PubMed (d) Computers

(e) Photo (f) Chameleon (g) Actor (h) Squirrel

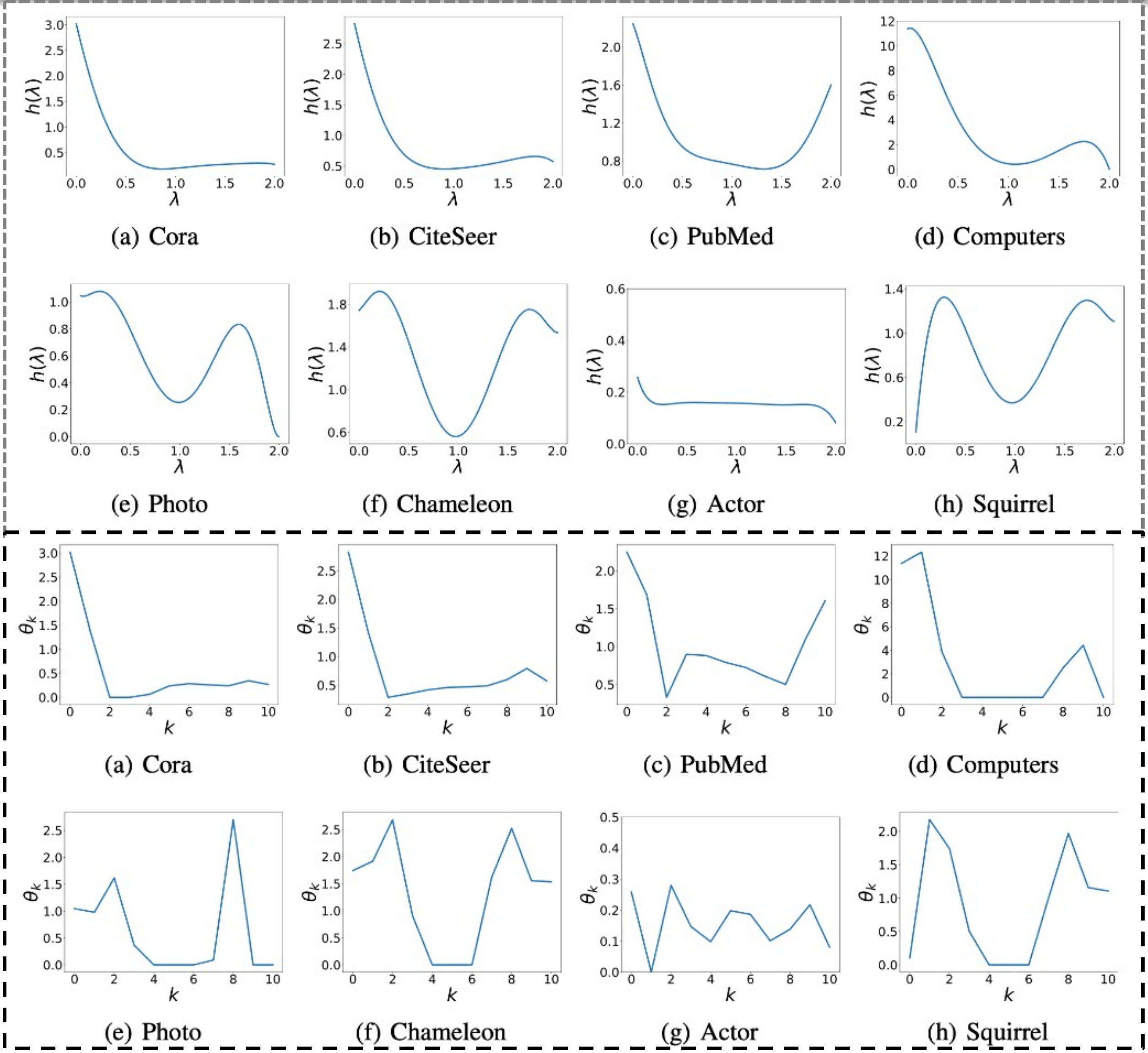
He M, Wei Z et al. BernNet: Learning Arbitrary Graph Spectral Filters via Bernstein Approximation. [NeurIPS’2021]

BernNet：实验结果

- 学习滤波器

滤波器

BernNet系数θ_k



He M, Wei Z et al. BernNet: Learning Arbitrary Graph Spectral Filters via Bernstein Approximation. [NeurIPS'2021]

报告提纲

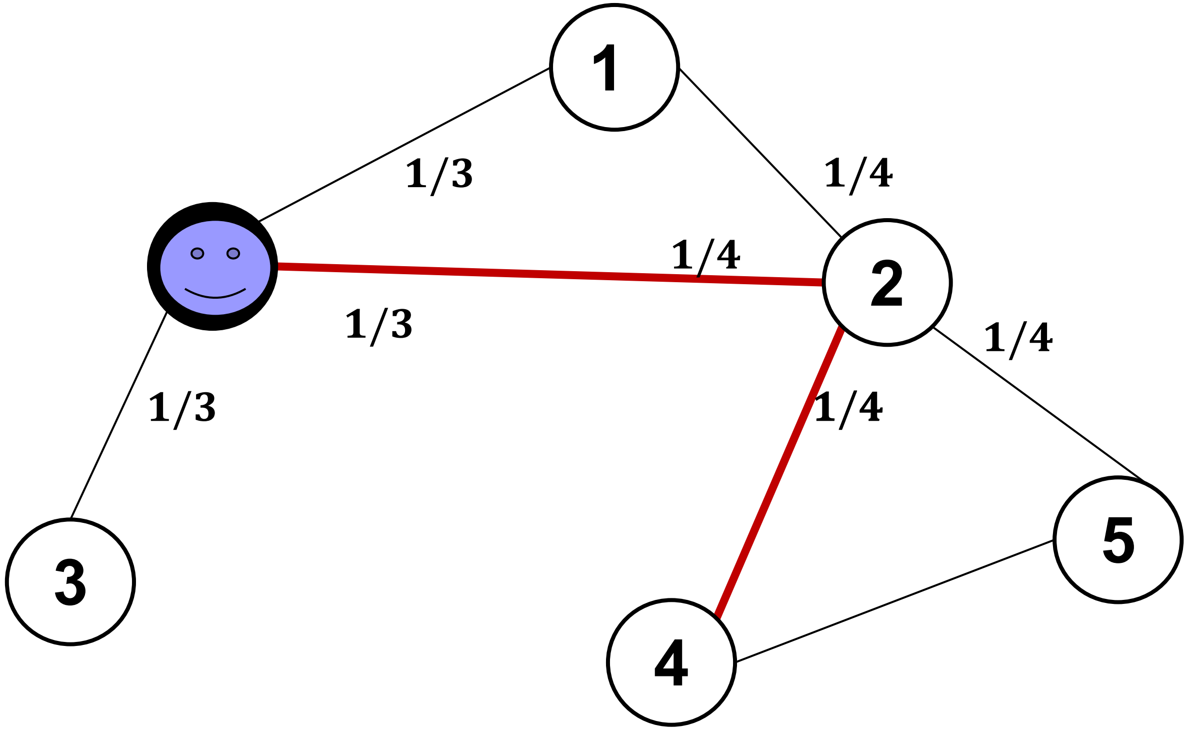
- 图神经网络的应用和概述
- 图神经网络的三个视角
  - □ 滤波器 ⇒ 学习任意的滤波器
  - □ 随机游走 ⇒ 基于重启随机游走的深度GNN
  - □ 优化函数 ⇒ 统一的GNN优化函数
- 展望与总结

相关工作

论文	主要方法
Representation Learning On Graphs With Jumping Knowledge Networks (JKNet, Xu et al., 2018)	第一个深度GNN模型，分析了GNN与随机游走的关系
Graph Neural Networks Exponentially Lose Expressive Power For Node Classification (Oono & Suzuki, 2020)	理论上证明了K层GCN的节点特征会收敛到一个子空间
Simple and Deep Graph Convolutional Networks (GCNII, Chen et al., 2020)	利用Initial residual和Identity mapping实现的深度GNN模型（Ours）
Towards Deeper Graph Neural Networks with Differentiable Group Normalization (DGN, Zhou et al, 2020)	通过Differentiable Group Normalization 解决GNN的过平滑
Training Graph Neural Networks with 1000 Layers (RevGNN, Li et al., 2021)	利用Reversible connection训练超过1000层的GNN模型

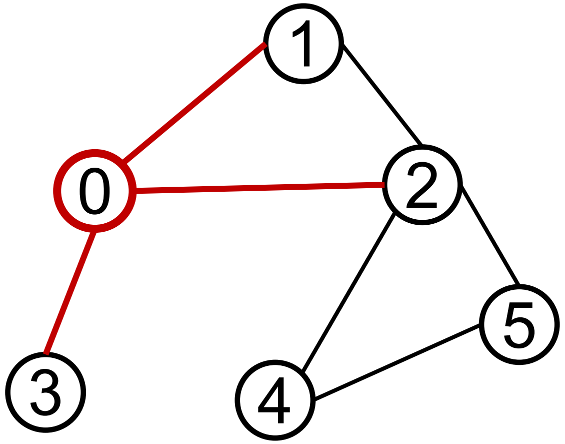
随机游走

- 每步游走时随机走向当前节点的邻居节点
- 概率转移矩阵$P = AD^{-1}$



随机游走

- 随机游走的矩阵形式：
  - 概率转移矩阵： $P = AD^{-1}$
  - 一步游走： $\pi^{(\ell+1)} = P \cdot \pi^{(\ell)}$



$\pi^{(\ell+1)} = P \cdot \pi^{(\ell)}$

$\pi^{(\ell+1)}$
0
1/3
1/3
1/3
0
0

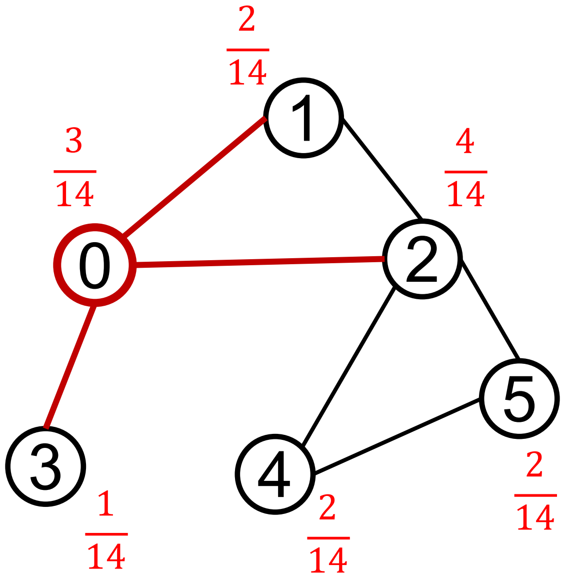
	1/2	1/4	1		
1/3		1/4			
1/3	1/2			1/2	1/2
1/3					
		1/4			1/2
		1/4		1/2	

$\pi^{(\ell)}$
1
0
0
0
0
0

随机游走稳态

- 随机游走的稳态分布：
  - □ 大部分随机游走有稳态分布π，$\boldsymbol{\pi} = \boldsymbol{P} \cdot \boldsymbol{\pi} = \boldsymbol{AD}^{-1} \cdot \boldsymbol{\pi}$
  - □ 无向图上随机游走的稳态分布：$\boldsymbol{\pi}(v) = \dfrac{d_v}{2m}$

稳态分布与起始节点无关



$\boldsymbol{\pi} \quad = \quad \boldsymbol{P} \quad \cdot \quad \boldsymbol{\pi}$

3/14
2/14
4/14
1/14
2/14
2/14

	1/2	1/4	1		
1/3		1/4			
1/3	1/2			1/2	1/2
1/3					
		1/4			1/2
		1/4		1/2	

3/14
2/14
4/14
1/14
2/14
2/14

随机游走稳态与Cheeger不等式

- Cheeger不等式

$$\left|\boldsymbol{\pi}_u^{(K)}(v) - \frac{d_v}{2m}\right| \le \sqrt{\frac{d_v}{d_u}}\left(1 - \frac{\lambda_{gap}^2}{2}\right)^K$$

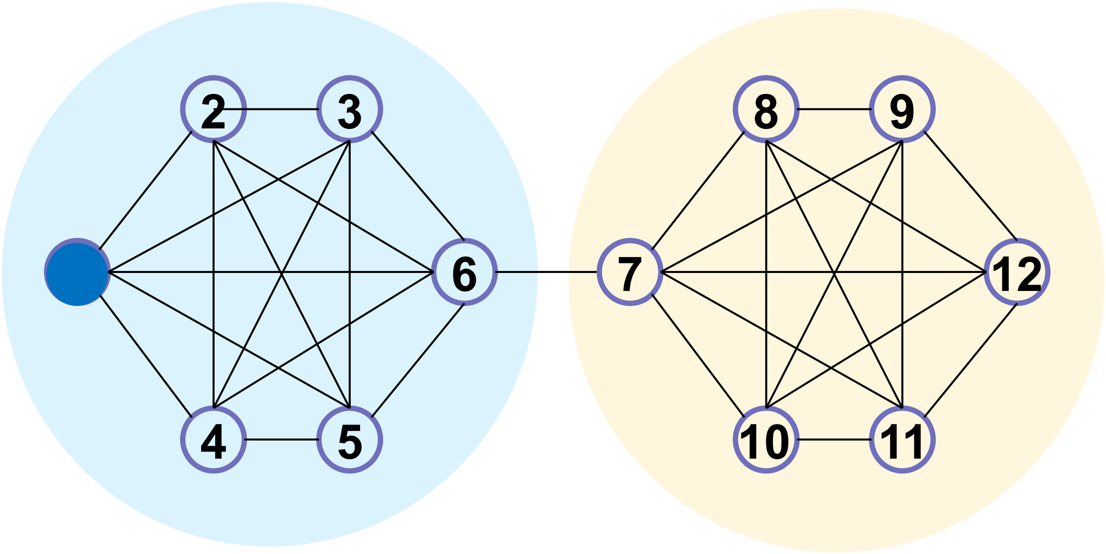
- 其中$\boldsymbol{\pi}_u^{(K)} = (\boldsymbol{A}\boldsymbol{D}^{-1})^K \boldsymbol{\pi}_u^{(0)}$从节点$u$开始的$K$步随机游走，$\lambda_{gap}$为spectral gap，即拉普拉斯矩阵最小非零特征值。

随机游走稳态与Cheeger不等式

- 稳态分布: $\pi(v) = \frac{d_v}{2\cdot|E|}$

$$\left[\frac{5}{62},\frac{5}{62},\frac{5}{62},\frac{5}{62},\frac{5}{62},\frac{6}{62},\frac{6}{62},\frac{5}{62},\frac{5}{62},\frac{5}{62},\frac{5}{62},\frac{5}{62}\right]$$

- Cheeger不等式：刻画收敛速度与社区发现

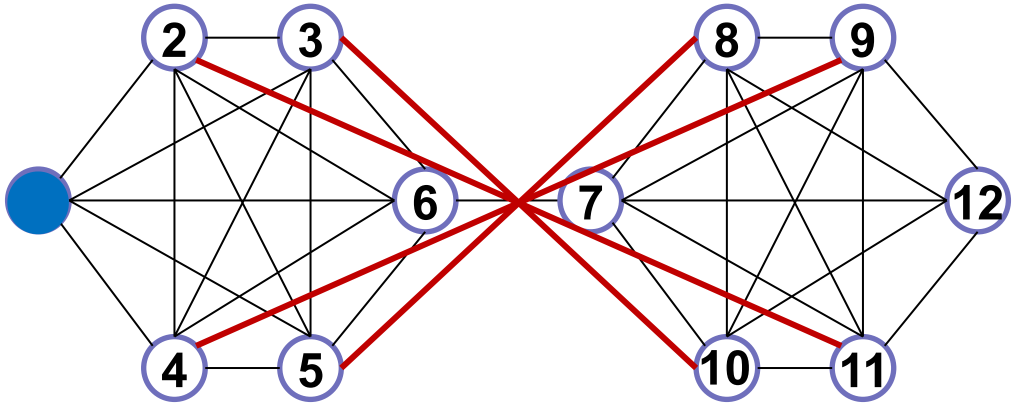


随机游走稳态与Cheeger不等式

- 稳态分布: $\pi(v)=\frac{d_v}{2\cdot|E|}$

$$\left[\frac{5}{70},\frac{6}{70},\frac{6}{70},\frac{6}{70},\frac{6}{70},\frac{6}{70},\frac{6}{70},\frac{6}{70},\frac{6}{70},\frac{6}{70},\frac{6}{70},\frac{5}{70}\right]$$

- Cheeger不等式：刻画收敛速度与社区发现



随机游走稳态与Cheeger不等式

- Cheeger不等式

$$\left|\boldsymbol{\pi}_u^{(K)}(v)-\frac{d_v}{2m}\right| \le \sqrt{\frac{d_v}{d_u}}\left(1-\frac{\lambda_{gap}^2}{2}\right)^K$$

  - □ 其中$\boldsymbol{\pi}_u^{(K)}=(\boldsymbol{AD}^{-1})^K\boldsymbol{\pi}_u^{(0)}$从节点$u$开始的$K$步随机游走，$\lambda_{gap}$为spectral gap，即拉普拉斯矩阵最小非零特征值。
- 随机游走趋于稳态的速度和节点的度数有关！

GCN与随机游走

- 深层图卷积神经网络会遭遇**过平滑问题**

$$H^{(\ell+1)} = \tilde{\boldsymbol{P}} \ldots \cancel{\sigma}(\tilde{\boldsymbol{P}}\cancel{\sigma}(\tilde{\boldsymbol{P}}\boldsymbol{X}\boldsymbol{W}^{(0)})\boldsymbol{W}^{(1)}) \ldots \boldsymbol{W}^{(\ell)}$$

$$\Rightarrow \boldsymbol{H}^{(K)} = \tilde{\boldsymbol{P}}^{K}\boldsymbol{X}\boldsymbol{W}$$

If $\tilde{\boldsymbol{P}} = \tilde{\boldsymbol{D}}^{-1/2}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}}^{-1/2}$, then

$$\tilde{\boldsymbol{P}}^{K} = \tilde{\boldsymbol{D}}^{-1/2}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}}^{-1}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}}^{-1} \ldots \tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}}^{-1}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}}^{-1/2}$$

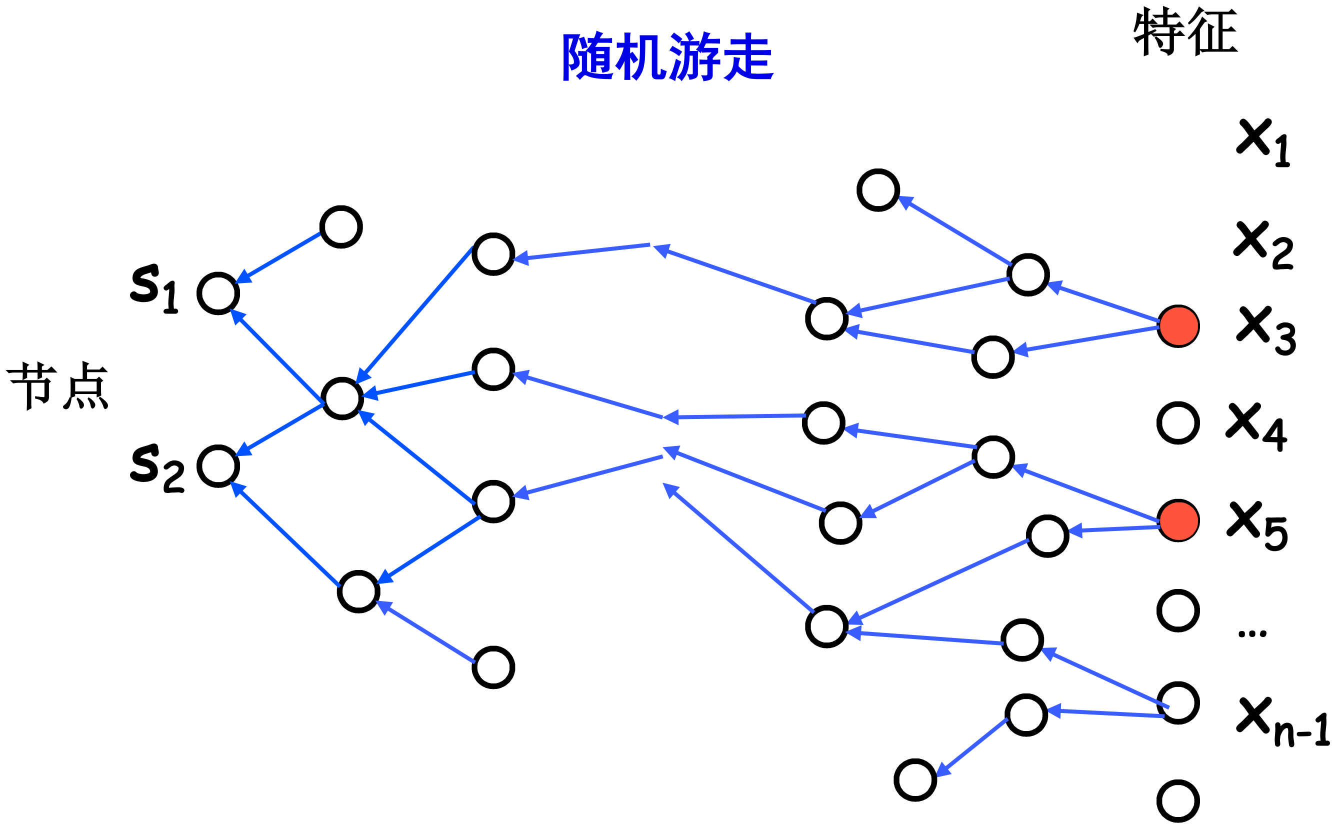
$$= \tilde{\boldsymbol{D}}^{-1/2}(\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}}^{-1}) \cdot (\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}}^{-1}) \ldots (\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}}^{-1})\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}}^{-1/2} \cdot \tilde{D}^{-1/2} \cdot \tilde{D}^{1/2}$$

$$= \tilde{\boldsymbol{D}}^{-1/2}(\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}}^{-1})^{K}\tilde{\boldsymbol{D}}^{1/2}$$

K步随机游走概率分布，最终会收敛到稳态

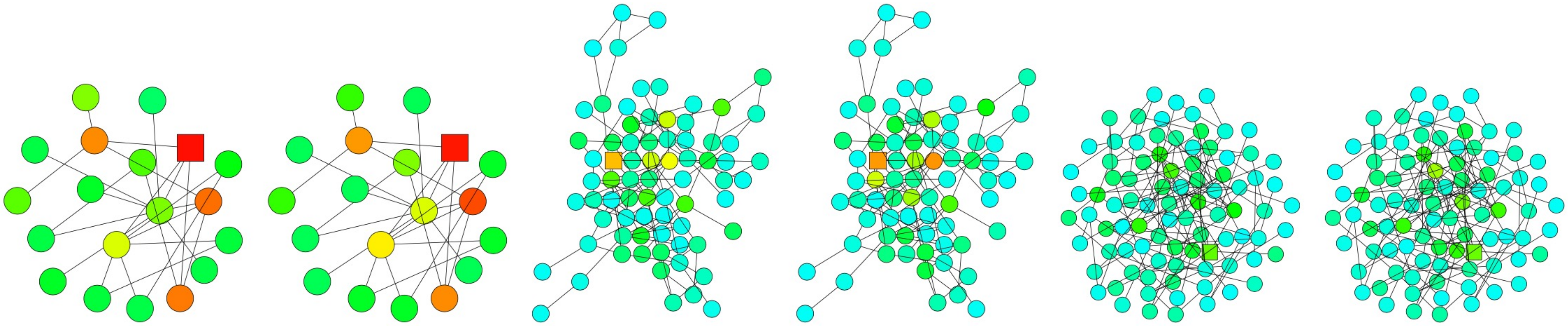
GCN与随机游走

- K层GCN消息传递 = 从特征(分布) $\boldsymbol{x}$到节点s的K步随机游走。



JKNet

- K层GCN消息传递 = 从源节点到目标节点的K步随机游走
- 当$K\to\infty$，极限分布与初始的节点表示无关，而只与图结构相关



(a) 2 layer GCN (b) 2 step r.w. (c) 4 layer GCN (d) 4 step r.w. (e) 6 layer GCN (f) 6 step r.w.

GCN与随机游走

- Residual Connection? $\boldsymbol{H}^{(\ell+1)}=\sigma\left((\widetilde{\boldsymbol{P}}+\boldsymbol{I})\cdot\boldsymbol{H}^{(\ell)}\cdot\boldsymbol{W}^{(\ell)}\right)$

$$\boldsymbol{H}^{(\ell+1)}=(\widetilde{\boldsymbol{P}}+\boldsymbol{I})\ldots\cancel{\sigma}\left((\widetilde{\boldsymbol{P}}+\boldsymbol{I})\cancel{\sigma}\left((\widetilde{\boldsymbol{P}}+\boldsymbol{I})\boldsymbol{X}\boldsymbol{W}^{(0)}\right)\boldsymbol{W}^{(1)}\right)\ldots\boldsymbol{W}^{(\ell)}$$

$$\Rightarrow \boldsymbol{H}^{(K)}=(\widetilde{\boldsymbol{P}}+\boldsymbol{I})^{K}\boldsymbol{X}\boldsymbol{W}$$

If $\widetilde{\boldsymbol{P}}=\widetilde{\boldsymbol{D}}^{-1/2}\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{D}}^{-1/2}$, $\boldsymbol{I}=\widetilde{\boldsymbol{D}}^{-1/2}\widetilde{\boldsymbol{D}}^{1/2}$, then

$$(\widetilde{\boldsymbol{P}}+\boldsymbol{I})^{K}=(\widetilde{\boldsymbol{D}}^{-1/2}\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{D}}^{-1/2}+\widetilde{\boldsymbol{D}}^{-1/2}\widetilde{\boldsymbol{D}}^{1/2})^{K}$$

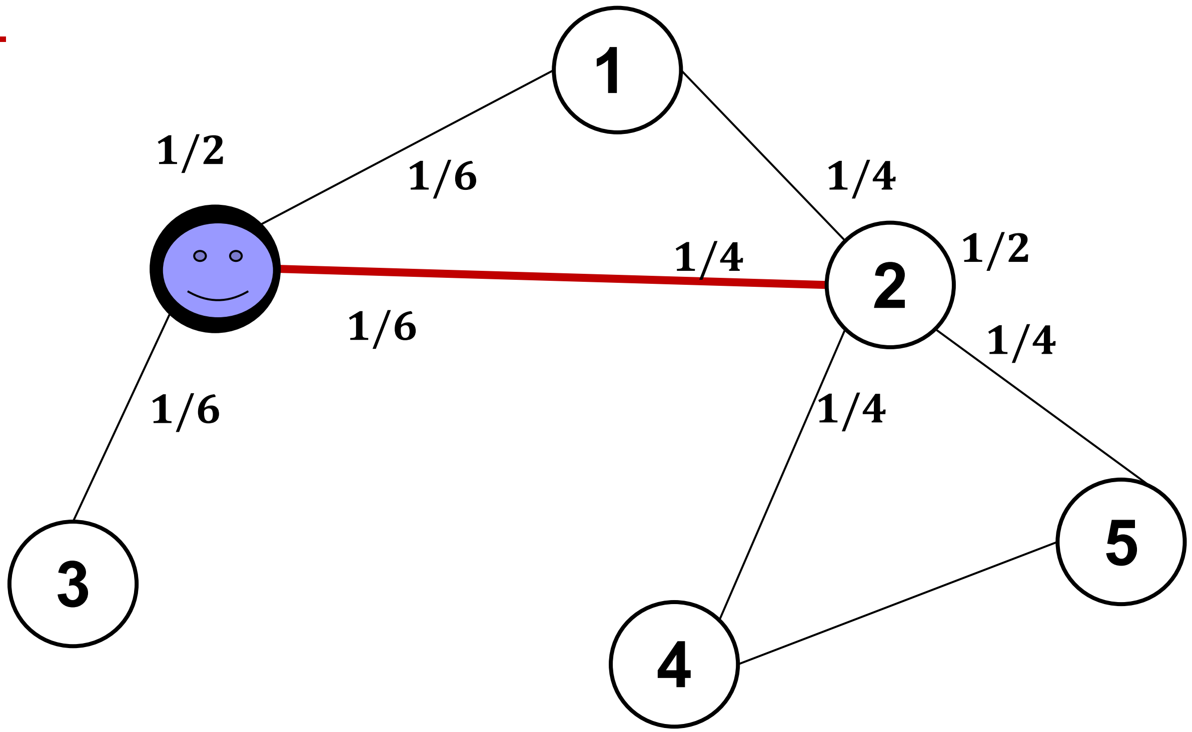
$$=\left(\widetilde{\boldsymbol{D}}^{-1/2}\left(\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{D}}^{-1}+\boldsymbol{I}\right)\widetilde{\boldsymbol{D}}^{1/2}\right)^{K}$$

$$=\widetilde{\boldsymbol{D}}^{-1/2}\left(\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{D}}^{-1}+\boldsymbol{I}\right)^{K}\widetilde{\boldsymbol{D}}^{1/2}$$

↳ K步**Lazy随机游走**概率分布，最终会收敛到稳态

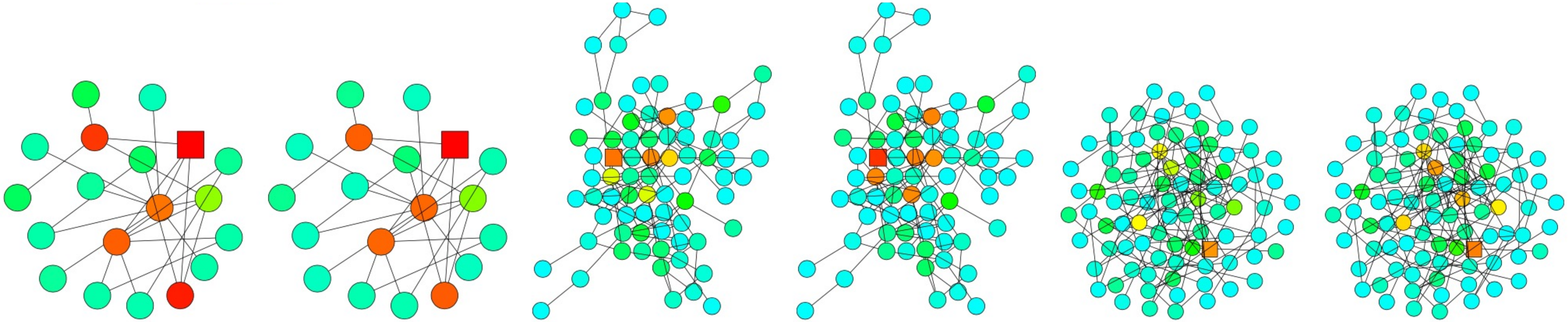
GCN与随机游走

- 添加Residual Connection的GCN=Lazy随机游走[Wang et al., 2019]
- 每步游走时有50%的概率停留在当前节点上，50%的概率邻居节点
- 概率转移矩阵$\boldsymbol{P} = \frac{\boldsymbol{A}\boldsymbol{D}^{-1}+\boldsymbol{I}}{2}$



JKNet

- 添加Residual Connections 的GCN=Lazy随机游走

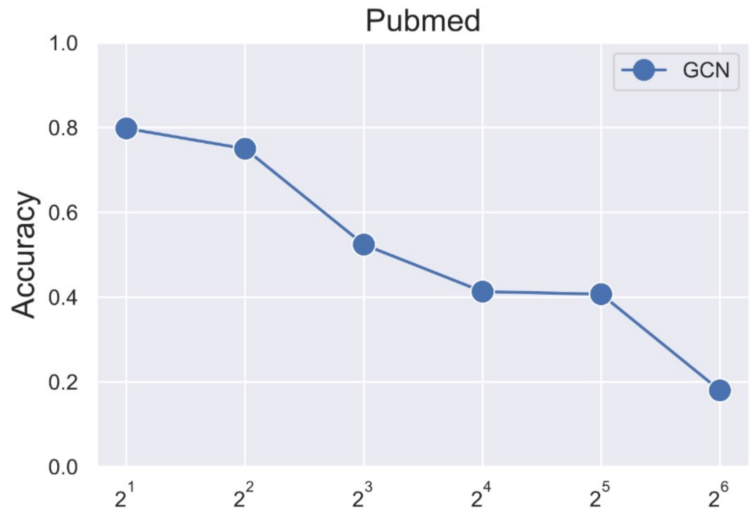


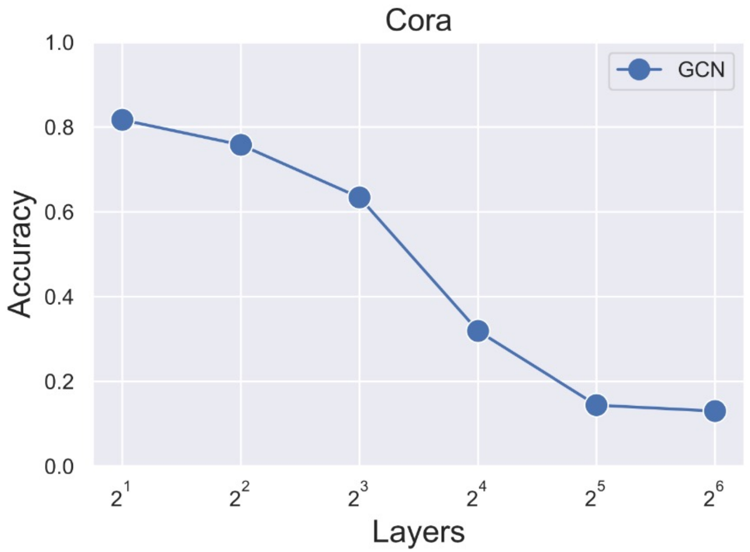
(a) 2 layer Res (b) 2 step lazy r.w. (c) 4 layer Res (d) 4 step lazy r.w. (e) 6 layer Res (f) 6 step lazy r.w.

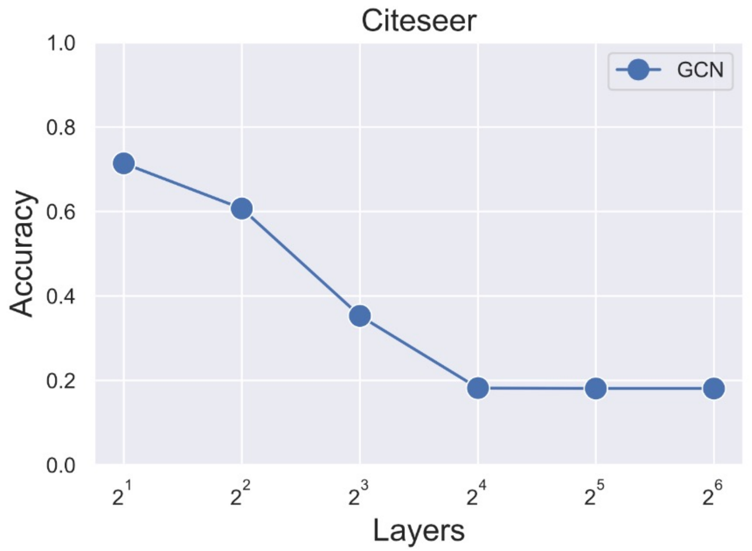
GCNs with residual connections and random walk distributions with lazy factor 0.4

GCN过平滑与随机游走稳态

- 过平滑的本质
  - 从特征出发的随机游走收敛到稳态，忘记了初始特征！







GCNII：实验结果

- Cheeger不等式

$$\left|\boldsymbol{\pi}_u^{(K)}(v)-\frac{d_v}{2m}\right|\le\sqrt{\frac{d_v}{d_u}}\left(1-\frac{\lambda_{gap}^2}{2}\right)^K$$

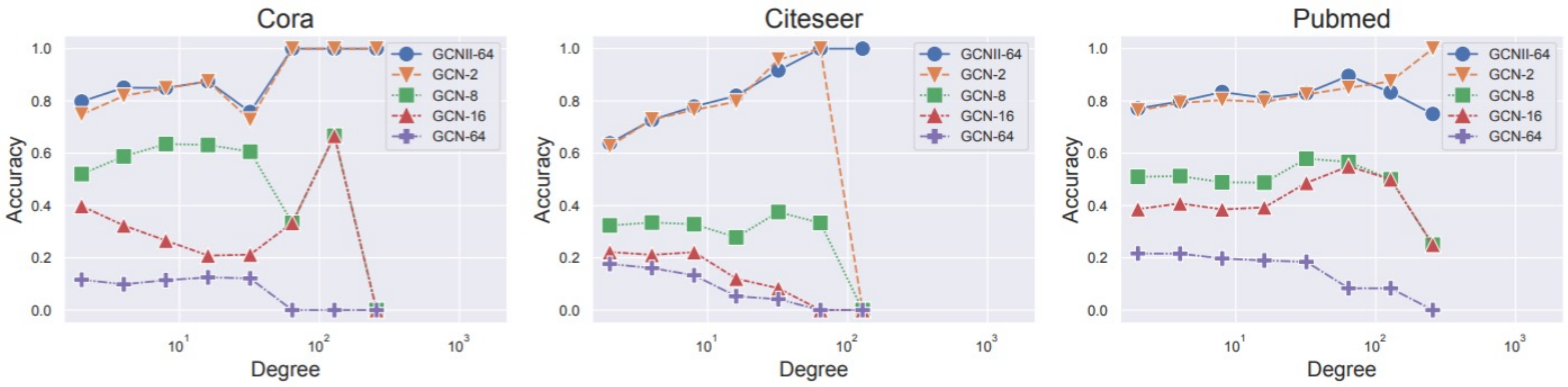


Figure 1. Semi-supervised node classification accuracy v.s. degree.

Chen M, Wei Z et al. Simple and Deep Graph Convolutional Networks. [ICML’2020]

Exponentially Lose Expressive Power

- [Oono & Suzuki, 2020]从理论上证明了K层GCN的节点特征会收敛到一个子空间并导致信息丢失。
- 收敛速度取决于$s^{(l)}$(权重矩阵$W^{(l)}$， $l = 1, \ldots, L$的最大奇异值)

Exponentially Lose Expressive Power

- [Oono & Suzuki, 2020]ReLU会加快GCN的收敛速度！

- 收敛速度取决于$s^{(l)}$(权重矩阵$W^{(l)}$，$l = 1, \dots, L$的最大奇异值)
- 有如下定理:

Theorem [Oono. and Suzuki, 20]

For the l –th layer $f^{(l)}$ of GCN and $\mathbf{x} \in \mathbb{R}^{N*}$ we have ,

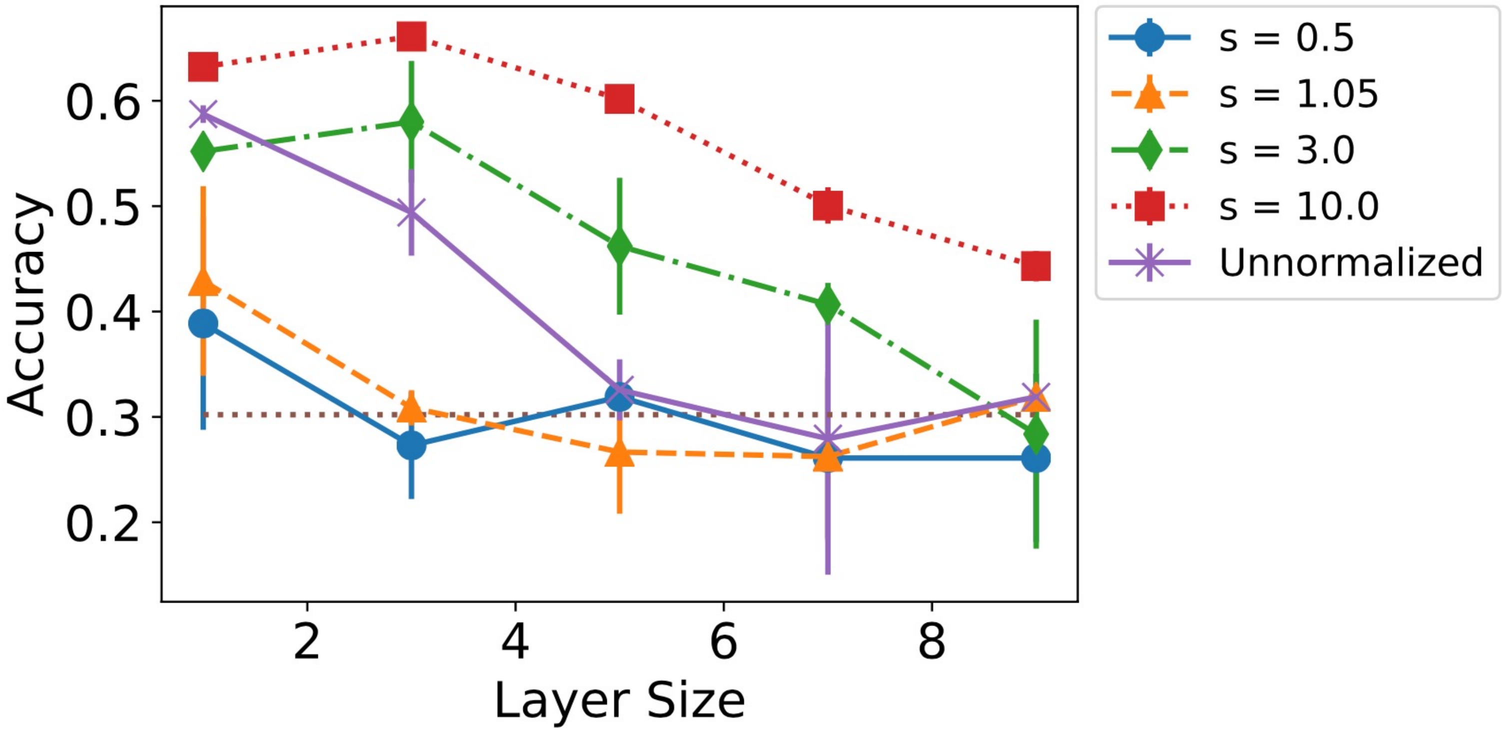
$$d_{\mathcal{M}}(f^{(l)}\boldsymbol{x}) \leq s^{(l)} \mu d_{\mathcal{M}}(\boldsymbol{x})$$

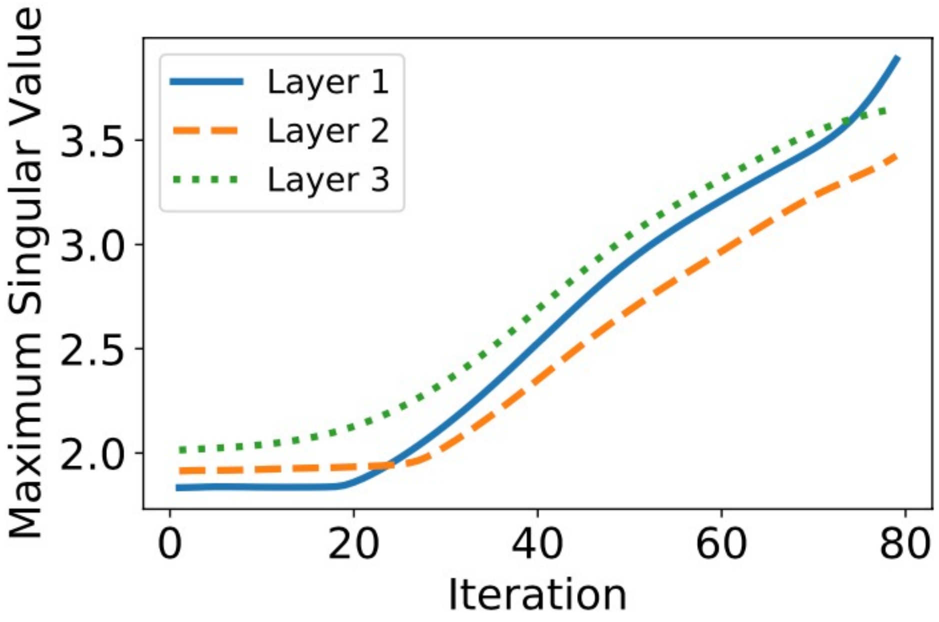
$d_{\mathcal{M}}$: L2 distance to $\mathcal{M}$

- 其中$\mathcal{M}$表示“Information-less” 信号空间，$\mu = \max_{i=2,\dots,n} |1 - \lambda_i|$ (λ_i为$\tilde{\boldsymbol{L}}$特征值)

Exponentially Lose Expressive Power

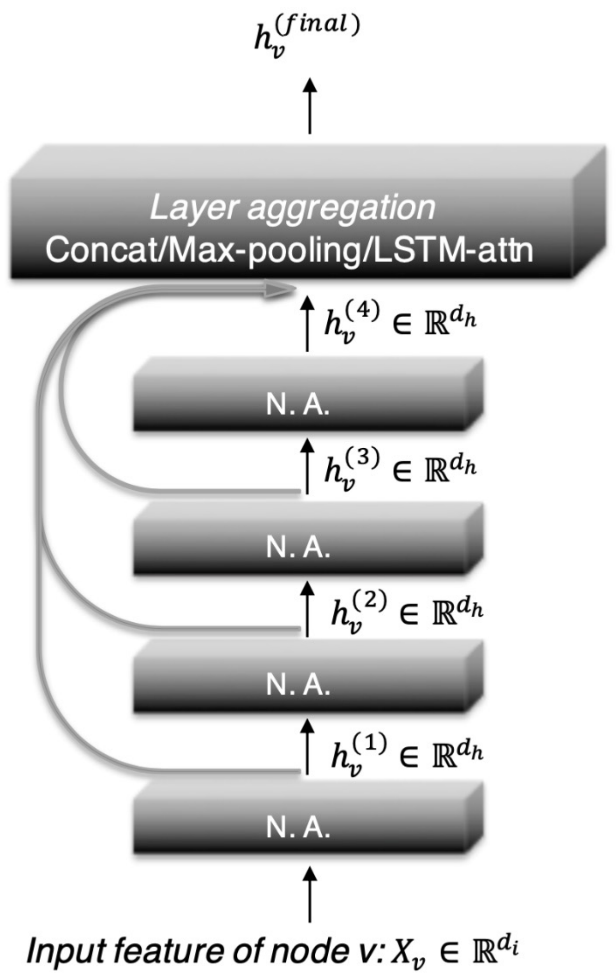
- GCN在Noisy Cora上的进行节点分类

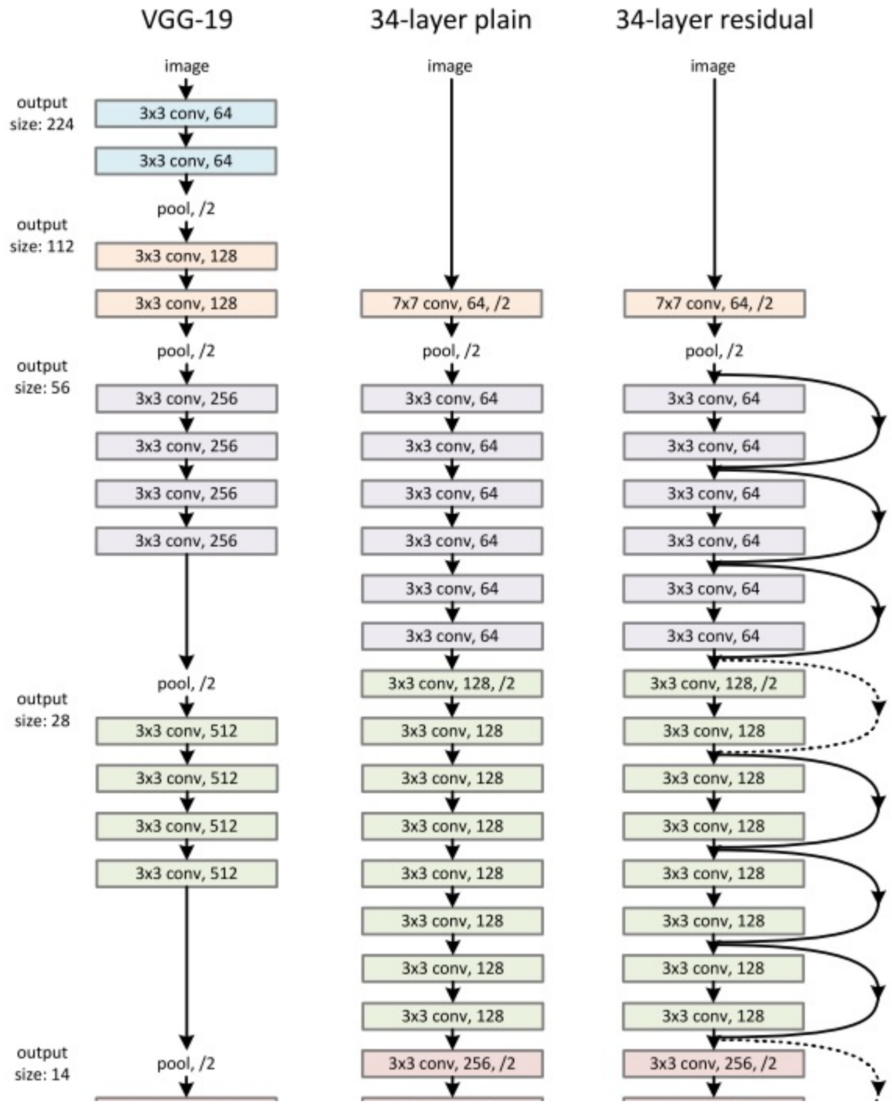




JKNet

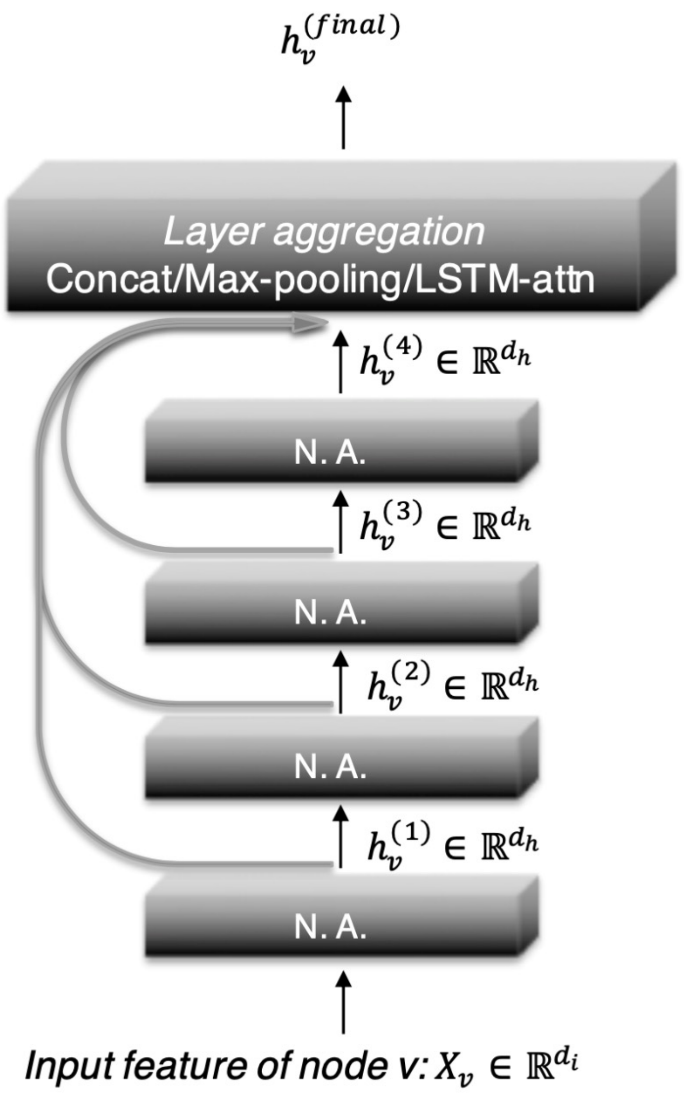
- Jumping Knowledge Network





JKNet

- Jumping Knowledge Network
- 最终表示的聚合方式
  - Concat
  - Max-pooling
  - LSTM-attention



JKNet

- Jumping Knowledge Network
- 实验结果

Dataset	Nodes	Edges	Classes	Features
Citeseer	3,327	4,732	6	3,703
Cora	2,708	5,429	7	1,433
Reddit	232,965	avg deg 492	50	300
PPI	56,944	818,716	121	50

Model	Citeseer	Model	Cora
GCN (2)	77.3 (1.3)	GCN (2)	88.2 (0.7)
GAT (2)	76.2 (0.8)	GAT (3)	87.7 (0.3)
JK-MaxPool (1)	77.7 (0.5)	JK-Maxpool (6)	**89.6** (0.5)
JK-Concat (1)	**78.3** (0.8)	JK-Concat (6)	89.1 (1.1)
JK-LSTM (2)	74.7 (0.9)	JK-LSTM (1)	85.8 (1.0)

Simple and Deep
Graph Convolutional Networks
(ICML, 2020)

Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, Yaliang Li

GCNII：算法思路

GCNII: Graph Convolutional Network via Initial residual and Identity mapping

- 原始GCN

$$\boldsymbol{H}^{(\ell+1)} = \sigma\left(\widetilde{\boldsymbol{P}}\boldsymbol{H}^{(\ell)}\boldsymbol{W}^{(\ell)}\right)$$

Chen M, Wei Z et al. Simple and Deep Graph Convolutional Networks. [ICML’2020]

GCNII：算法思路

GCNII: Graph Convolutional Network via Initial residual and Identity mapping

- 原始GCN

$$\boldsymbol{H}^{(\ell+1)} = \sigma\left(\widetilde{\boldsymbol{P}}\boldsymbol{H}^{(\ell)}\boldsymbol{W}^{(\ell)}\right)$$

- GCN+**初始值(Initial residual)**

$$\boldsymbol{H}^{(\ell+1)} = \sigma\left(\left((1-\alpha_\ell)\widetilde{\boldsymbol{P}}\boldsymbol{H}^{(\ell)} + \alpha_\ell \boldsymbol{H}^{(0)}\right)\boldsymbol{W}^{(\ell)}\right)$$

Chen M, Wei Z et al. Simple and Deep Graph Convolutional Networks. [ICML'2020]

GCNII：算法思路

GCNII: Graph Convolutional Network via Initial residual and Identity mapping

- 原始GCN

$$\boldsymbol{H}^{(\ell+1)}=\sigma\left(\widetilde{\boldsymbol{P}}\boldsymbol{H}^{(\ell)}\boldsymbol{W}^{(\ell)}\right)$$

- GCN+**初始值(Initial residual)**

$$\boldsymbol{H}^{(\ell+1)}=\sigma\left(\left((1-\alpha_\ell)\widetilde{\boldsymbol{P}}\boldsymbol{H}^{(\ell)}+\alpha_\ell\boldsymbol{H}^{(0)}\right)\boldsymbol{W}^{(\ell)}\right)$$

- **GCNII**

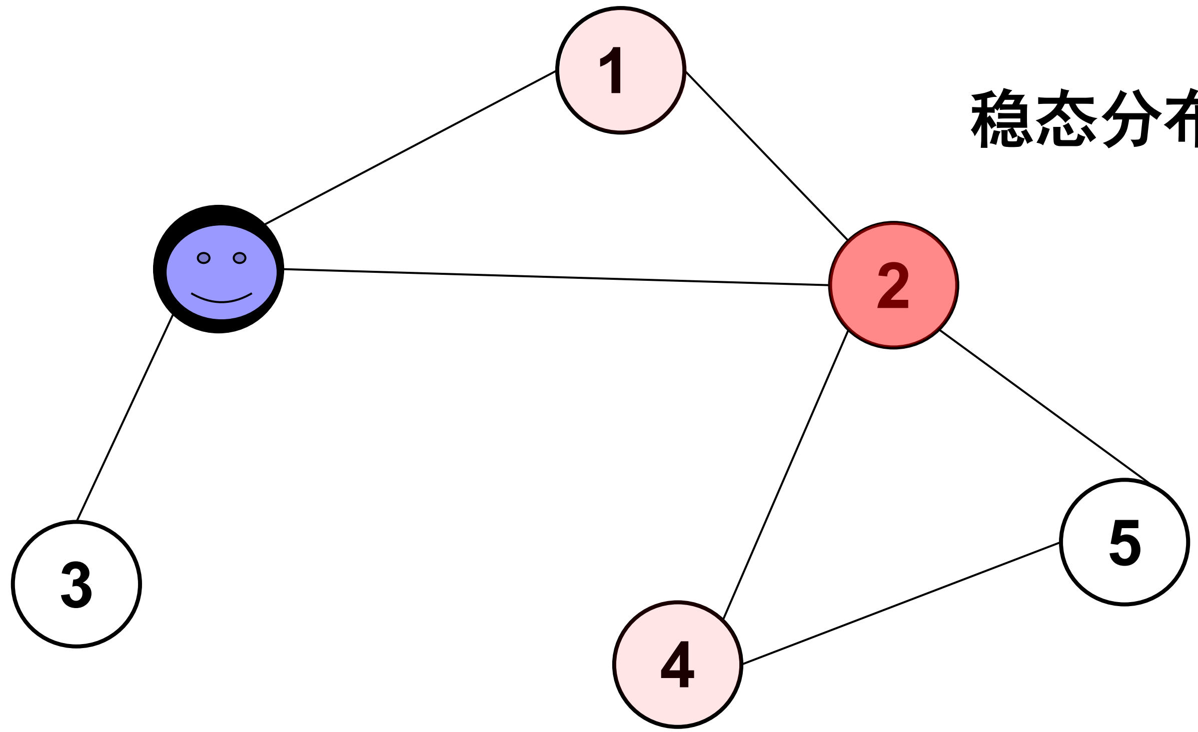
GCN+**初始值(Initial residual)**+**单位映射(Identity mapping)**

$$\boldsymbol{H}^{(\ell+1)}=\sigma\left(\left((1-\alpha_\ell)\widetilde{\boldsymbol{P}}\boldsymbol{H}^{(\ell)}+\alpha_\ell\boldsymbol{H}^{(0)}\right)\left((\mathbf{1}-\beta_\ell)\boldsymbol{I}_n+\beta_\ell\boldsymbol{W}^{(\ell)}\right)\right)$$

Chen M, Wei Z et al. Simple and Deep Graph Convolutional Networks. [ICML’2020]

带重启的随机游走

- 在每步游走中:
  - 以α的概率返回初始状态;
  - 以$(1-\alpha)$的概率随机走向当前节点的任一邻居

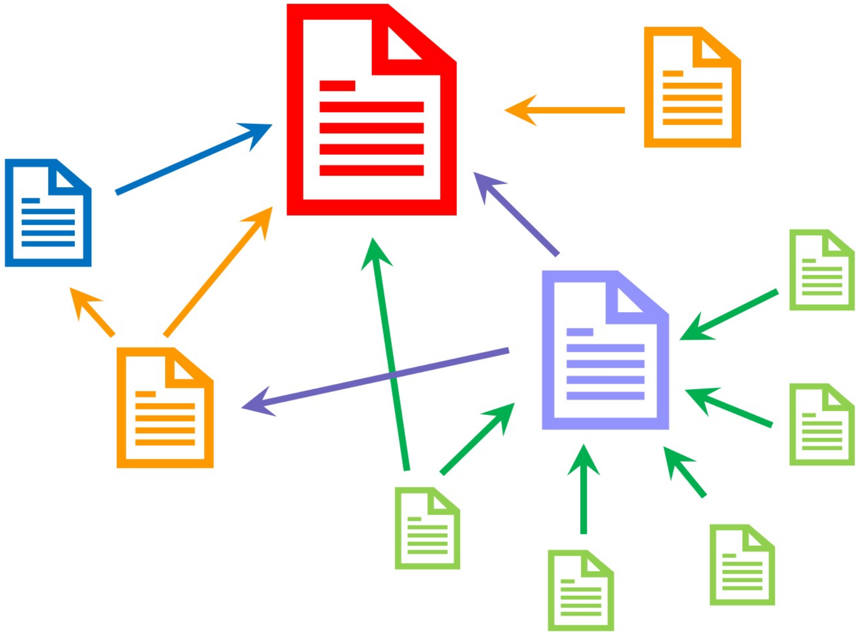


稳态分布与起始节点**有关**

Chen M, Wei Z et al. Simple and Deep Graph Convolutional Networks. [ICML'2020]

PageRank与PPR

- PageRank定义式:$\boldsymbol{\pi} = (1-\alpha)\boldsymbol{\pi}\boldsymbol{P} + \alpha \cdot \boldsymbol{e}$
  - $\boldsymbol{\pi}$：PageRank向量. $\boldsymbol{e}$：起始向量.
  - $\boldsymbol{P}$：概率转移矩阵. α：衰减系数.



PageRank: $\boldsymbol{e} = \left(\frac{1}{n}, \frac{1}{n}, \dots, \frac{1}{n}\right)$

衡量图上节点的全局重要性.

Personalized PageRank:

$$\boldsymbol{e} = (0,0,\dots,0,\boxed{1},0\dots,0)$$

第s项为1

衡量图上节点关于源节点s的相对重要性

Chen M, Wei Z et al. Simple and Deep Graph Convolutional Networks. [ICML'2020]

GCNII：算法思路

- GCN（固定系数）

$$\boldsymbol{H}^{(K)}=\widetilde{\boldsymbol{P}}^{K}\boldsymbol{x} \longrightarrow \text{随机游走}$$

- GCNII（支持选择系数）

$$\boldsymbol{H}^{(K)}=\left(\sum_{i=0}^{K}\theta_i\,\widetilde{\boldsymbol{P}}^{i}\right)\boldsymbol{x} \xrightarrow{\theta_i} \text{PPR}$$

APPNP (Klicpera et al., 2019)

PPR: $\theta_i=\alpha(1-\alpha)^i$

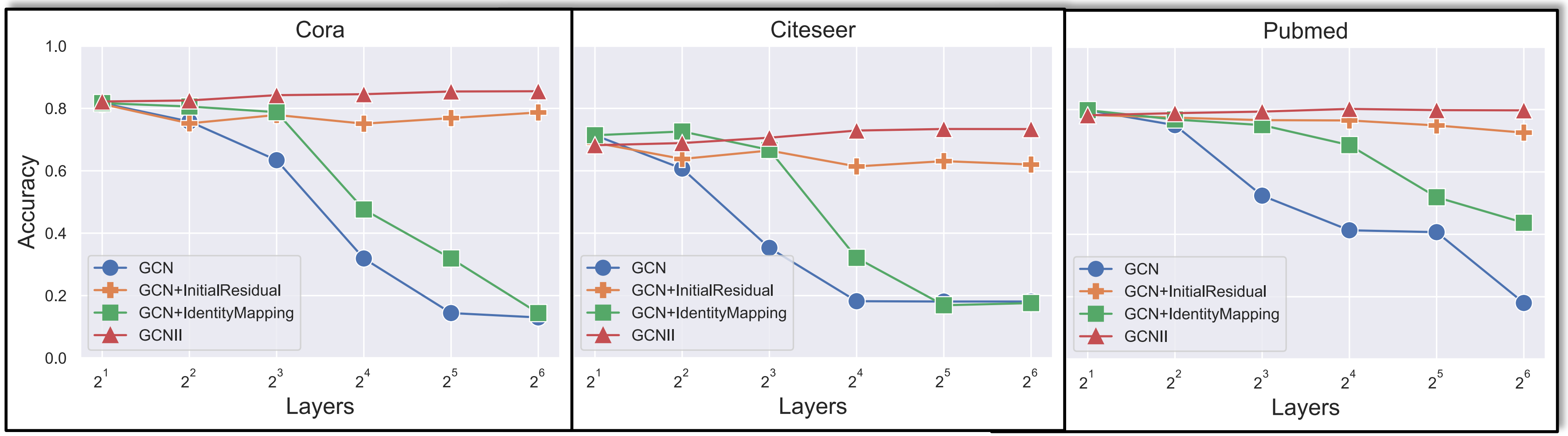
HKPR

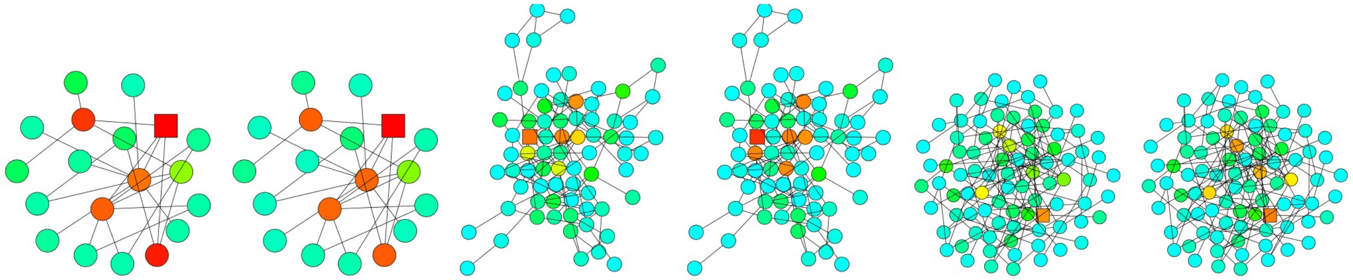
……

Chen M, Wei Z et al. Simple and Deep Graph Convolutional Networks. [ICML'2020]

GCNII：实验结果

- GCN + residual connection [Kipf et al., 2017]





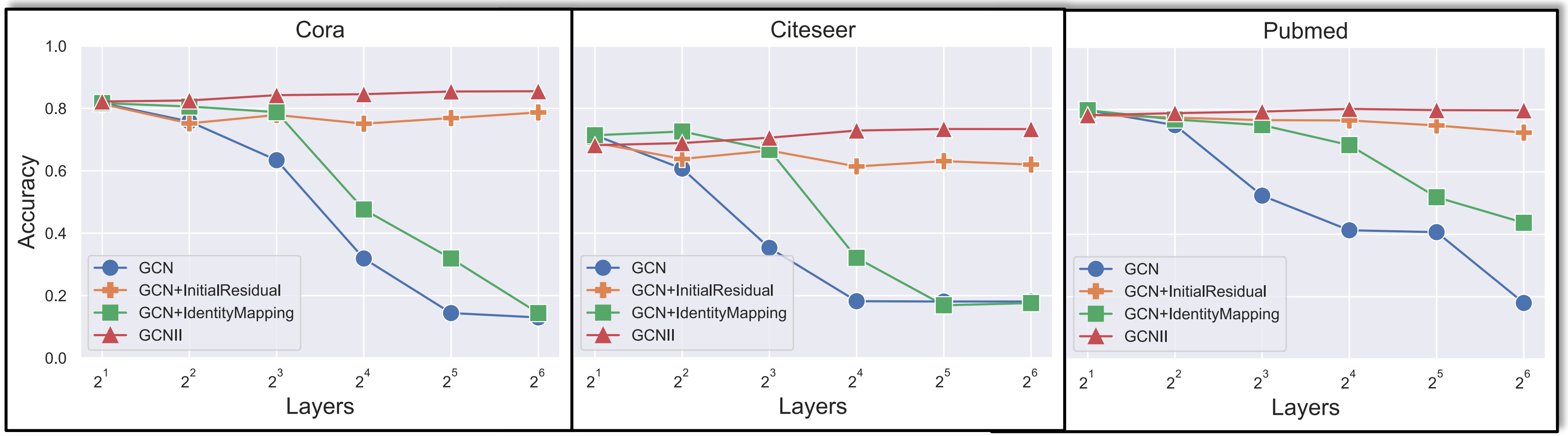
(a) 2 layer Res (b) 2 step lazy r.w. (c) 4 layer Res (d) 4 step lazy r.w. (e) 6 layer Res (f) 6 step lazy r.w.

添加Residual Connections 的GCN=Lazy随机游走

Chen M, Wei Z et al. Simple and Deep Graph Convolutional Networks. [ICML'2020]

GCNII：实验结果

- GCN + residual connection [Kipf et al., 2017]



$$\boldsymbol{H}^{(\ell+1)} = \sigma\left(\left((1-\alpha_\ell)\widetilde{\boldsymbol{P}}\boldsymbol{H}^{(\ell)} + \alpha_\ell \boldsymbol{H}^{(0)}\right)\left((1-\beta_\ell)\boldsymbol{I}_n + \beta_\ell \boldsymbol{W}^{(\ell)}\right)\right)$$

Identity Matters in Deep Learning

Moritz Hardt * Tengyu Ma†

July 23, 2018

Abstract

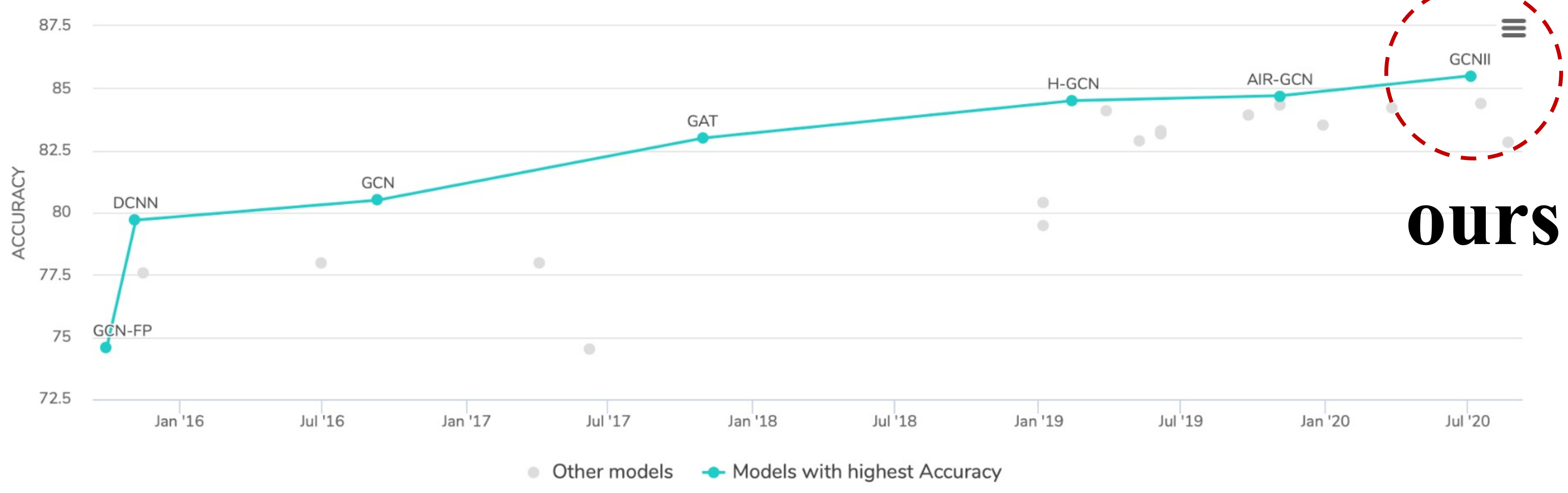
An emerging design principle in deep learning is that each layer of a deep artificial neural network should be able to easily express the identity transformation. This idea not only motivated various normalization techniques, such as *batch*

性能表现不会比浅层模型差(APPNP)

Chen M, Wei Z et al. Simple and Deep Graph Convolutional Networks. [ICML’2020]

GCNII：实验结果

Node Classification on Cora with Public Split: fixed 20 nodes per class

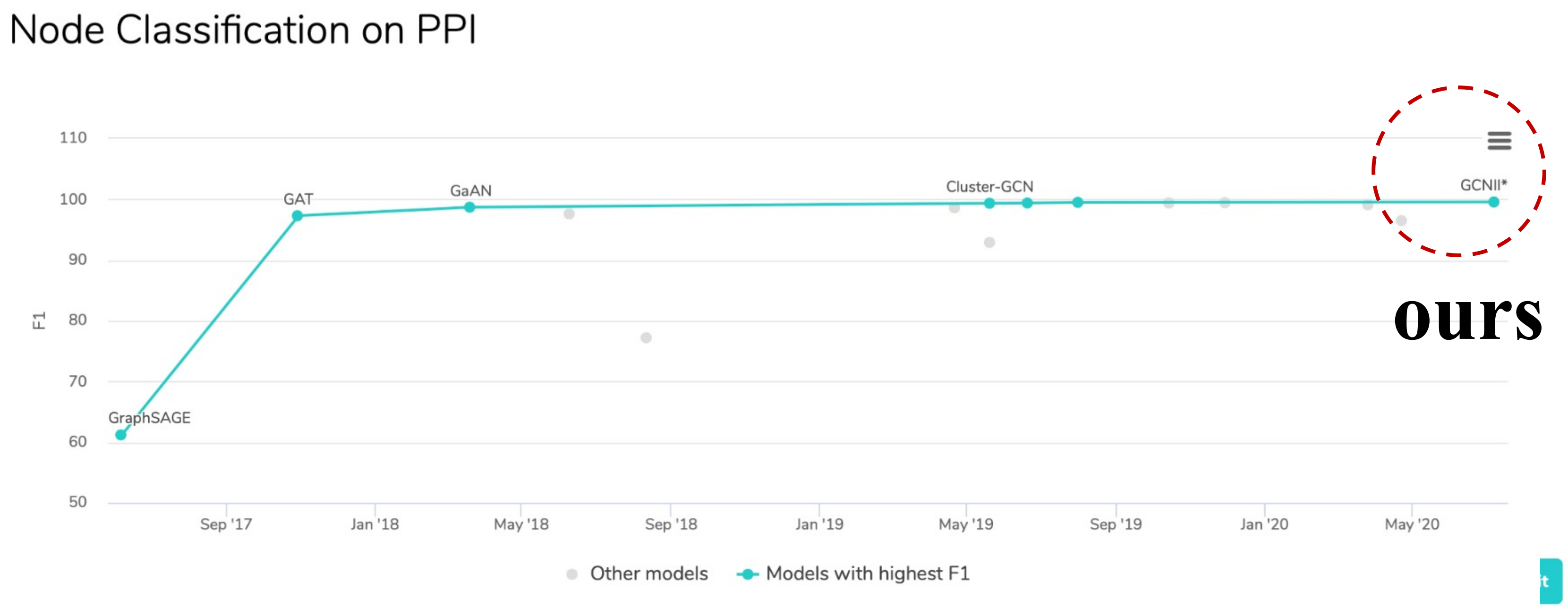


ours

RANK	MODEL	ACCURACY	PAPER	CODE	RESULT	YEAR
1	GCNII	85.5%	Simple and Deep Graph Convolutional Networks			2020
2	AIR-GCN	84.7%	GraphAIR: Graph Representation Learning with Neighborhood Aggregation and Interaction			2019
3	H-GCN	84.5%	Hierarchical Graph Convolutional Networks for Semi-supervised Node Classification			2019

Chen M, Wei Z et al. Simple and Deep Graph Convolutional Networks. [ICML'2020]

GCNII：实验结果



RANK	MODEL	F1	MICRO-F1	PAPER	CODE	RESULT	YEAR
1	GCNII*	99.56		Simple and Deep Graph Convolutional Networks			2020
2	GraphSAINT	99.50		GraphSAINT: Graph Sampling Based Inductive Learning Method			2019
3	SGAS	99.46		SGAS: Sequential Greedy Architecture Search			2019

Chen M, Wei Z et al. Simple and Deep Graph Convolutional Networks. [ICML'2020]

报告提纲

- 图神经网络的应用和概述
- 图神经网络的三个视角
  - □ 滤波器 ➡ 学习任意的滤波器
  - □ 随机游走 ➡ 基于重启随机游走的深度GNN
  - □ 优化函数 ➡ 统一的GNN优化函数
- 展望与总结

现有工作

论文	主要方法
A Note on Over-Smoothing for Graph Neural Networks (Cai et al., 2020)	从优化函数的视角解释了GCN会过平滑的原因
Interpreting and Unifying Graph Neural Networks with An Optimization Framework (GNN-LF/HF, Zhu et al., 2021)	从优化函数的视角解释和统一了现有GNN模型，并提出新模型
Graph Neural Networks Inspired by Classical Iterative Algorithms (TWIRLS, Yang et al., 2021)	从优化函数的视角解释激活函数、GAT，提出新模型（Ours）
Scaling Up Graph Neural Networks Via Graph Coarsening (Huang et al., 2021)	从优化函数的视角研究可扩展GNN
Topology Attack and Defense for Graph Neural Networks: An Optimization Perspective (Xu et al., 2019)	从优化函数的视角研究GNN的攻击和防御

优化函数

- 从优化函数的视角理解GCN模型[Cai et al.,2020]

$$\min_{\boldsymbol{z}} f(\boldsymbol{z}) = \boldsymbol{z}^T \tilde{\boldsymbol{L}}\, \boldsymbol{z}$$

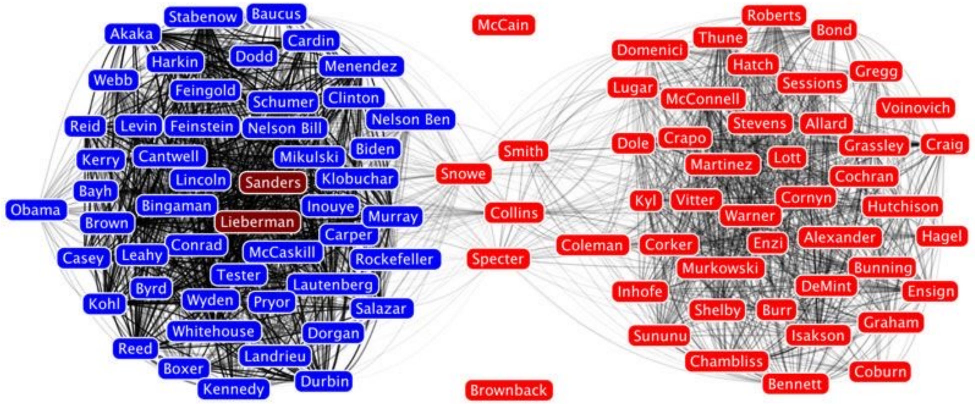
狄利克雷能量(Dirichlet energy)

$$\boldsymbol{z}^T \tilde{\boldsymbol{L}} \boldsymbol{z} = \sum_{(i,j)\in E} A_{ij} \cdot \left(\frac{z_i}{\sqrt{1+d_i}} - \frac{z_j}{\sqrt{1+d_j}} \right)^2$$

同配性假设 (Homophily Assumption):

Birds of a feather flock together.

邻居节点的特征具有相关性



优化函数

- 从优化函数的视角理解GCN模型[Cai et al.,2020]

$$\min_{\boldsymbol{z}} f(\boldsymbol{z}) = \boldsymbol{z}^T \tilde{\boldsymbol{L}}\, \boldsymbol{z}$$

狄利克雷能量(Dirichlet energy)

$$\boldsymbol{z}^T \tilde{\boldsymbol{L}} \boldsymbol{z} = \sum_{(i,j)\in E} A_{ij} \cdot \left(\frac{z_i}{\sqrt{1+d_i}} - \frac{z_j}{\sqrt{1+d_j}} \right)^2$$

- □ 近端梯度下降：$\frac{\partial f(z)}{\partial z} = 0 \Rightarrow \boldsymbol{L}\boldsymbol{z} = \boldsymbol{0} \Rightarrow \boldsymbol{z}^{(k+1)} = (\boldsymbol{I} - \tilde{\boldsymbol{L}})\boldsymbol{z}^{(k)}$；
- □ 令$\boldsymbol{z}^{(0)} = \boldsymbol{x}\boldsymbol{w}$，则 $\boldsymbol{z}^{(k+1)} = \left(\tilde{\boldsymbol{D}}^{-1/2}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}}^{-1/2}\right)\boldsymbol{z}^{(k)}$
- □ GCN迭代优化$f(\boldsymbol{z})$直到$\min_{\boldsymbol{z}} f(\boldsymbol{z}) = \boldsymbol{0}$

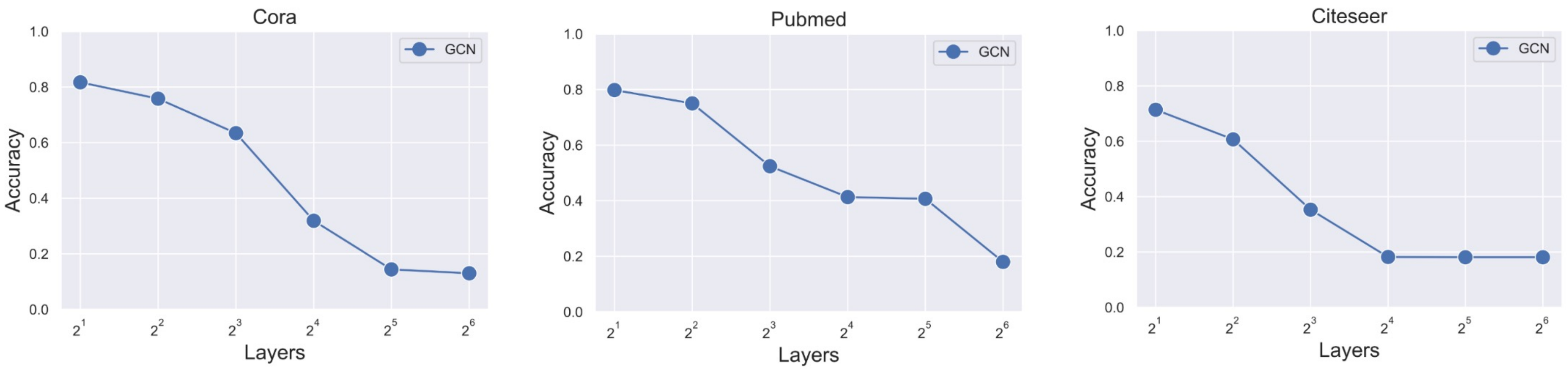
优化函数

- 从优化函数的视角理解GCN模型[Cai et al.,2020]

$$\min_{\boldsymbol{z}} f(\boldsymbol{z}) = \boldsymbol{z}^T \tilde{\boldsymbol{L}}\, \boldsymbol{z}$$

狄利克雷能量(Dirichlet energy)

$$\boldsymbol{z}^T \tilde{\boldsymbol{L}} \boldsymbol{z} = \sum_{(i,j)\in E} A_{ij} \cdot \left(\frac{z_i}{\sqrt{1+d_i}} - \frac{z_j}{\sqrt{1+d_j}} \right)^2$$

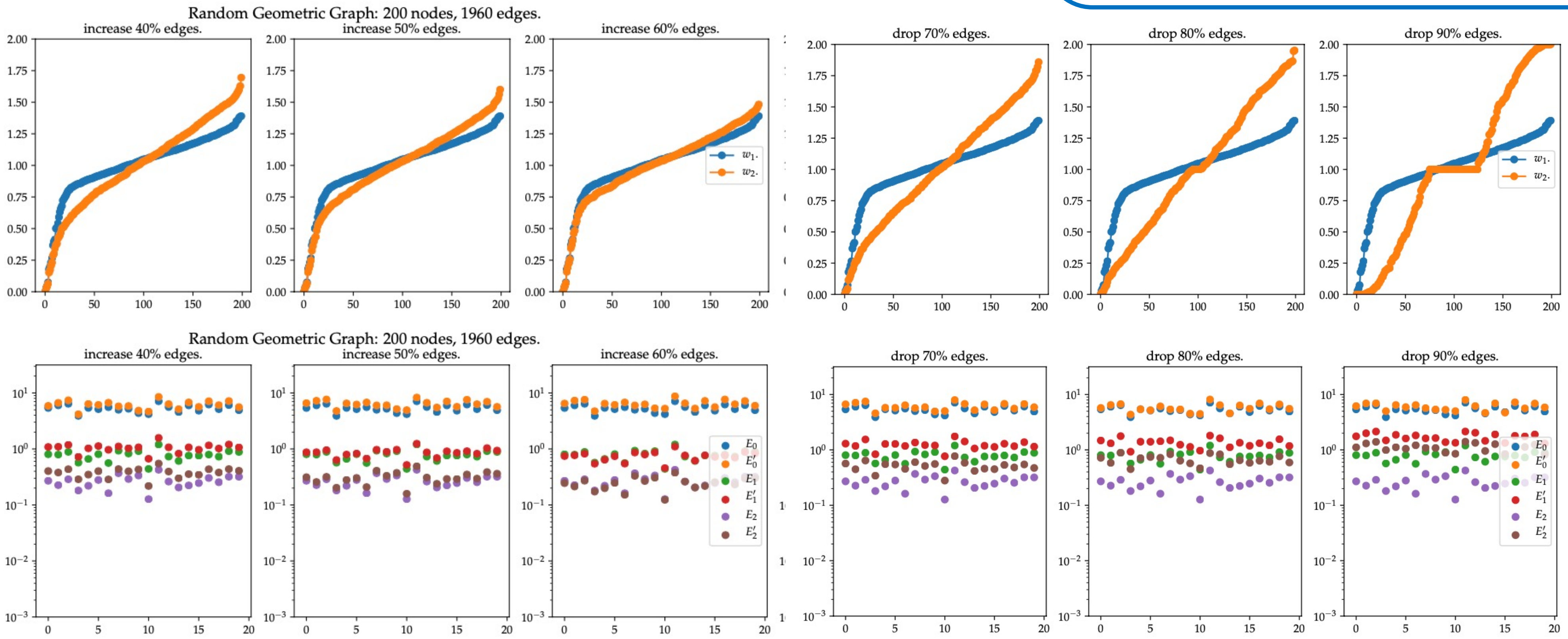


优化函数

- 实验分析[Cai et al.,2020]
  - 删除边和增加边的权重可以增加狄利克雷能量

狄利克雷能量(Dirichlet energy)

$$z^T\tilde{L}z=\sum_{(i,j)\in E}A_{ij}\cdot\left(\frac{z_i}{\sqrt{1+d_i}}-\frac{z_j}{\sqrt{1+d_j}}\right)^2$$



优化函数

- 从优化函数的视角理解GNN模型[Zhu et al.,2004, Zhu et al.,2021]

$$\min_{\boldsymbol{z}} f(\boldsymbol{z}) = \boxed{(1-\alpha)\boldsymbol{z}^T\tilde{\boldsymbol{L}}\,\boldsymbol{z}} + \boxed{\alpha\|\boldsymbol{z}-\boldsymbol{x}\|_2^2}$$

狄利克雷能量(Dirichlet energy)

$$\boldsymbol{z}^T\tilde{\boldsymbol{L}}\boldsymbol{z} = \sum_{(i,j)\in E} A_{ij}\cdot\left(\frac{z_i}{\sqrt{1+d_i}} - \frac{z_j}{\sqrt{1+d_j}}\right)^2$$

ℓ_2-正则项，约束最终嵌入表示$\boldsymbol{z}$和初始特征$\boldsymbol{x}$的相似程度

优化函数

- 从优化函数的视角理解GNN模型[Zhu et al.,2004, Zhu et al.,2021]

$$\min_{\boldsymbol{z}} f(\boldsymbol{z}) = (1-\alpha)\boldsymbol{z}^T \tilde{\boldsymbol{L}}\, \boldsymbol{z} + \alpha\|\boldsymbol{z}-\boldsymbol{x}\|_2^2$$

- 解释现有GNN模型：APPNP[Klicpera et.al. 2019]/GCNII
  - $\frac{\partial f(z)}{\partial z} = 2(1-\alpha)\tilde{\boldsymbol{L}}\boldsymbol{z} + 2\alpha(\boldsymbol{z}-\boldsymbol{x})$；
  - 令$\frac{\partial f(z)}{\partial z} = 0 \Longrightarrow \boldsymbol{z} = (1-\alpha)\tilde{\boldsymbol{P}}\boldsymbol{z} + \alpha\boldsymbol{x}$；

优化函数

- 从优化函数的视角理解GNN模型[Zhu et al.,2004, Zhu et al.,2021]

$$\min_{\boldsymbol{z}} f(\boldsymbol{z}) = (1-\alpha)\boldsymbol{z}^T \tilde{\boldsymbol{L}}\, \boldsymbol{z} + \alpha \|\boldsymbol{z}-\boldsymbol{x}\|_2^2$$

Model	Characteristic	Propagation Mechanism	Corresponding Objective
GCN/SGC [14]	K-layer graph convolutions	$\mathbf{Z} = \hat{\tilde{\mathbf{A}}}^K \mathbf{X}\mathbf{W}^*$	$\mathcal{O} = \min_{\mathbf{Z}} \{tr(\mathbf{Z}^T\tilde{\mathbf{L}}\mathbf{Z})\}, \mathbf{Z}^{(0)} = \mathbf{X}\mathbf{W}^*$
GC Operation [14]	1-layer graph convolution	$\mathbf{Z} = \hat{\tilde{\mathbf{A}}}\mathbf{X}\mathbf{W}$	$\mathcal{O} = \min_{\mathbf{Z}} \{\|\mathbf{Z}-\mathbf{H}\|_F^2 + tr(\mathbf{Z}^T\tilde{\mathbf{L}}\mathbf{Z})\}, \mathbf{H} = \mathbf{X}\mathbf{W}$, (*first-order*)
PPNP/APPNP [15]	Personalized pagerank	$\mathbf{H} = f_\theta(\mathbf{X}), \begin{cases} \textbf{PPNP:} \quad \mathbf{Z} = \alpha(\mathbf{I}-(1-\alpha)\hat{\tilde{\mathbf{A}}})^{-1}\mathbf{H} \\ \textbf{APPNP:} \quad \mathbf{Z} = \left\langle (1-\alpha)\hat{\tilde{\mathbf{A}}}\mathbf{Z}^{(k)} + \alpha\mathbf{H} \right\rangle_K, \mathbf{Z}^{(0)} = \mathbf{H} \end{cases}$	$\mathcal{O} = \min_{\mathbf{Z}} \{\|\mathbf{Z}-\mathbf{H}\|_F^2 + (1/\alpha - 1)tr(\mathbf{Z}^T\tilde{\mathbf{L}}\mathbf{Z})\}$
JKNet [39]	Jumping to the last layer	$\mathbf{Z} = \sum_{k=1}^{K} \alpha_k \hat{\tilde{\mathbf{A}}}^k \mathbf{X}\mathbf{W}^*$	$\mathcal{O} = \min_{\mathbf{Z}} \{\|\mathbf{Z}-\hat{\tilde{\mathbf{A}}}\mathbf{H}\|_F^2 + \xi tr(\mathbf{Z}^T\tilde{\mathbf{L}}\mathbf{Z})\}, \mathbf{H} = \mathbf{X}\mathbf{W}^*$
DAGNN [18]	Adaptively incorporating different layers	$\mathbf{H} = f_\theta(\mathbf{X}), \quad \mathbf{Z} = \sum_{k=0}^{K} s_k \hat{\tilde{\mathbf{A}}}^k \mathbf{H}$	$\mathcal{O} = \min_{\mathbf{Z}} \{\|\mathbf{Z}-\mathbf{H}\|_F^2 + \xi tr(\mathbf{Z}^T\tilde{\mathbf{L}}\mathbf{Z})\}$
GNN-LF (ours)	Flexible low-pass filtering kernel	$\mathbf{H} = f_\theta(\mathbf{X}), \begin{cases} \textbf{closed:} \quad \mathbf{Z} = \{\{\mu + 1/\alpha - 1\}\mathbf{I} + \{2-\mu-1/\alpha\}\hat{\tilde{\mathbf{A}}}\}^{-1}\{\mu\mathbf{I} + (1-\mu)\hat{\tilde{\mathbf{A}}}\}\mathbf{H} \\ \textbf{iter:} \quad \mathbf{Z} = \left\langle \frac{1+\alpha\mu-2\alpha}{1+\alpha\mu-\alpha}\hat{\tilde{\mathbf{A}}}\mathbf{Z}^{(k)} + \frac{\alpha\mu}{1+\alpha\mu-\alpha}\mathbf{H} + \frac{\alpha-\alpha\mu}{1+\alpha\mu-\alpha}\hat{\tilde{\mathbf{A}}}\mathbf{H} \right\rangle_K, \\ \quad \mathbf{Z}^{(0)} = \frac{\mu}{1+\alpha\mu-\alpha}\mathbf{H} + \frac{1-\mu}{1+\alpha\mu-\alpha}\hat{\tilde{\mathbf{A}}}\mathbf{H} \end{cases}$	$\mathcal{O} = \min_{\mathbf{Z}} \{\|\{\mu\mathbf{I} + (1-\mu)\hat{\tilde{\mathbf{A}}}\}^{1/2}(\mathbf{Z}-\mathbf{H})\|_F^2 + (1/\alpha - 1)tr(\mathbf{Z}^T\tilde{\mathbf{L}}\mathbf{Z})\}$
GNN-HF (ours)	Flexible high-pass filtering kernel	$\mathbf{H} = f_\theta(\mathbf{X}), \begin{cases} \textbf{closed:} \quad \mathbf{Z} = \{(\beta + 1/\alpha)\mathbf{I} + (1-\beta-1/\alpha)\hat{\tilde{\mathbf{A}}}\}^{-1}\{\mathbf{I} + \beta\tilde{\mathbf{L}}\}\mathbf{H} \\ \textbf{iter:} \quad \mathbf{Z} = \left\langle \frac{\alpha\beta-\alpha+1}{\alpha\beta+1}\hat{\tilde{\mathbf{A}}}\mathbf{Z}^{(k)} + \frac{\alpha}{\alpha\beta+1}\mathbf{H} + \frac{\alpha\beta}{\alpha\beta+1}\tilde{\mathbf{L}}\mathbf{H} \right\rangle_K, \\ \quad \mathbf{Z}^{(0)} = \frac{1}{\alpha\beta+1}\mathbf{H} + \frac{\beta}{\alpha\beta+1}\tilde{\mathbf{L}}\mathbf{H} \end{cases}$	$\mathcal{O} = \min_{\mathbf{Z}} \{\|\{\mathbf{I} + \beta\tilde{\mathbf{L}}\}^{1/2}(\mathbf{Z}-\mathbf{H})\|_F^2 + (1/\alpha - 1)tr(\mathbf{Z}^T\tilde{\mathbf{L}}\mathbf{Z})\}$

优化函数

- 从优化函数设计新的GNN模型[Zhu et al.,2021] GNN-LF

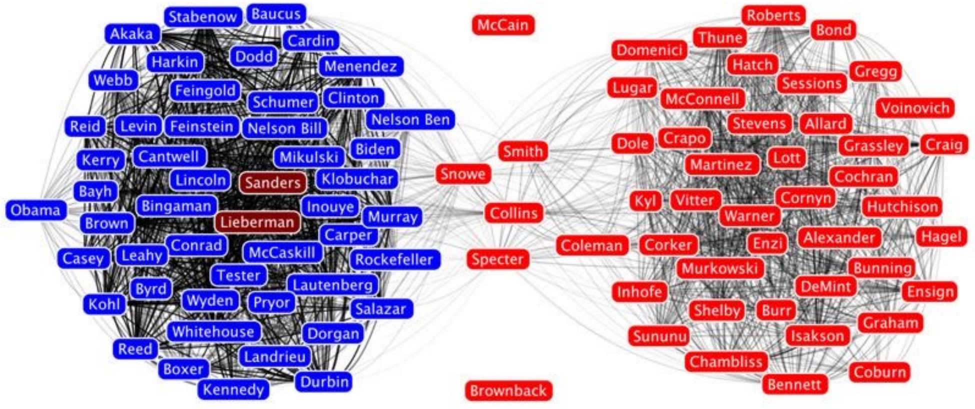
$$\min_{\boldsymbol{z}} f(\boldsymbol{z}) = (1-\alpha)\boldsymbol{z}^T \tilde{\boldsymbol{L}} \boldsymbol{z} + \alpha \left\| \boxed{\left(\mu \boldsymbol{I} + (1-\mu)\widetilde{\boldsymbol{P}}\right)^{1/2}} (\boldsymbol{z}-\boldsymbol{x}) \right\|_2^2$$

$\mu \in [1/2,1)$低通滤波器， $\boldsymbol{z} = \left((\mu + 1/\alpha - 1)\boldsymbol{I} + (2-\mu-1/\alpha)\widetilde{\boldsymbol{P}}\right)^{-1}\left(\mu\boldsymbol{I} + (1-\mu)\widetilde{\boldsymbol{P}}\right)\boldsymbol{x}$

同配性假设 (Homophily Assumption):

Birds of a feather flock together.

邻居节点的特征具有相关性



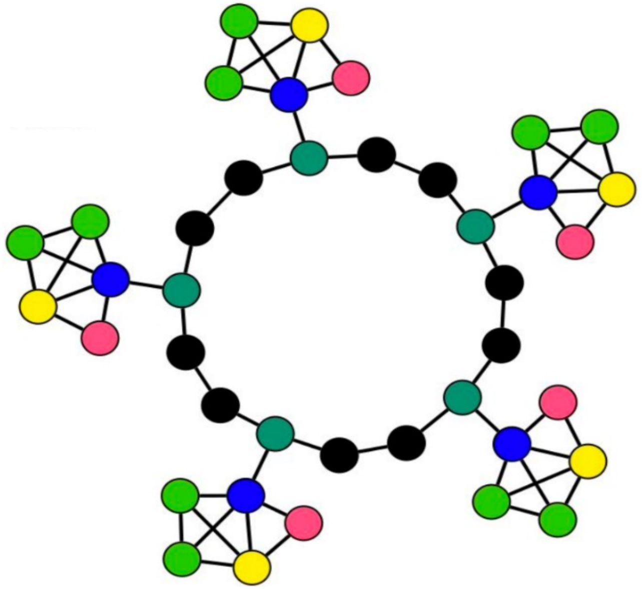
优化函数

- 从优化函数设计新的GNN模型[Zhu et al.,2021] GNN-HF

$$\min_{\boldsymbol{z}} f(\boldsymbol{z}) = (1-\alpha)\boldsymbol{z}^T\tilde{\boldsymbol{L}}\boldsymbol{z} + \alpha\left\|\boxed{(\boldsymbol{I}+\beta\tilde{\boldsymbol{L}})^{1/2}}(\boldsymbol{z}-\boldsymbol{x})\right\|_2^2$$

$\beta \in (0,+\infty)$高通滤波器， $\boldsymbol{Z} = \left((\beta+1/\alpha)\boldsymbol{I} + (1-\beta-1/\alpha)\tilde{\boldsymbol{P}}\right)^{-1}(\boldsymbol{I}+\beta\tilde{\boldsymbol{L}})\boldsymbol{H}$

异配性假设 (Homophily Assumption):
邻居节点的特征具有负相关性



优化函数

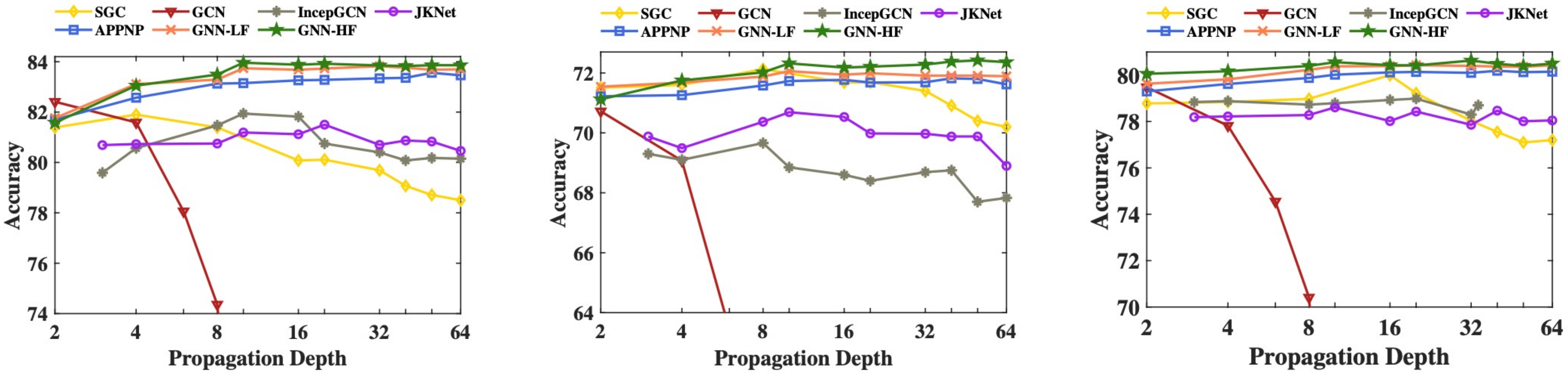
- GNN-LF/HF实验结果

Dataset	Classes	Nodes	Edges	Features	Train/Val/Test
Cora	7	2708	5429	1433	140/500/1000
Citeseer	6	3327	4732	3703	120/500/1000
Pubmed	3	19717	44338	500	60/500/1000
ACM	3	3025	13128	1870	60/500/1000
Wiki-CS	10	11701	216123	300	200/500/1000
MS Academic	15	18333	81894	6805	300/500/1000

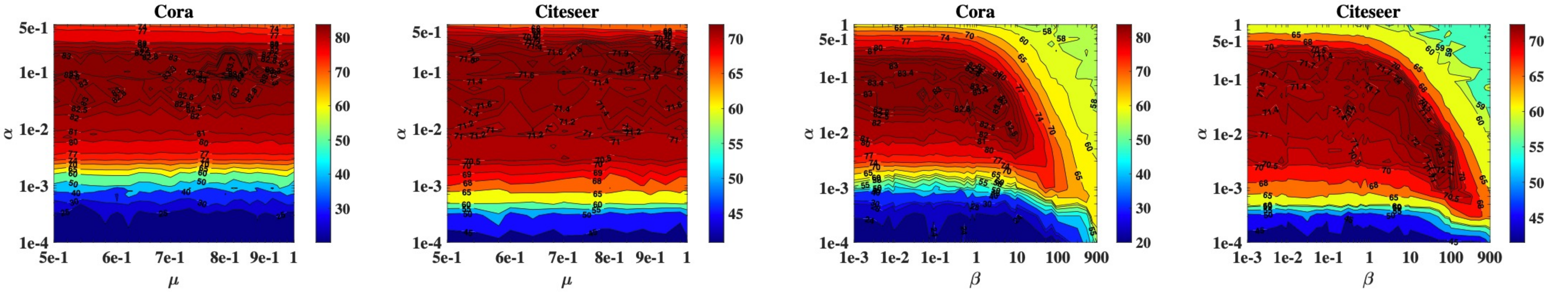
Model	Dataset					
	Cora	Citeseer	Pubmed	ACM	Wiki-CS	MS Academic
MLP	57.79±0.11	61.20±0.08	73.23±0.05	77.39±0.11	65.66±0.20	87.79±0.42
LP	71.50±0.00	50.80±0.00	72.70±0.00	63.30±0.00	34.90±0.00	74.10±0.00
ChebNet	79.92±0.18	70.90±0.37	76.98±0.16	79.53±1.24	63.24±1.43	90.76±0.73
GAT	82.48±0.31	72.08±0.41	79.08±0.22	88.24±0.38	74.27±0.63	91.58±0.25
GraphSAGE	82.14±0.25	71.80±0.36	79.20±0.27	87.57±0.65	73.17±0.41	91.53±0.15
IncepGCN	81.94±0.94	69.66±0.29	78.88±0.35	87.75±0.61	60.54±1.06	75.45±0.49
GCN	82.41±0.25	70.72±0.36	79.40±0.15	88.38±0.51	71.97±0.51	92.17±0.11
SGC	81.90±0.23	72.21±0.22	78.30±0.14	87.56±0.34	72.43±0.28	88.35±0.36
PPNP	83.34±0.20	71.73±0.30	80.06±0.20	89.12±0.17	74.53±0.36	92.27±0.23
APPNP	83.32±0.42	71.67±0.48	80.05±0.27	89.04±0.21	74.30±0.50	92.25±0.18
JKNet	81.19±0.49	70.69±0.88	78.60±0.25	88.11±0.36	60.90±0.92	87.26±0.23
GNN-LF-closed	83.70±0.14	71.98±0.33	80.34±0.18	89.43±0.20	**75.50±0.56**	**92.79±0.15**
GNN-LF-iter	83.53±0.24	71.92±0.24	80.33±0.20	89.37±0.40	75.35±0.24	92.69±0.20
GNN-HF-closed	**83.96±0.22**	**72.30±0.28**	80.41±0.25	89.46±0.30	74.92±0.45	92.47±0.23
GNN-HF-iter	83.79±0.29	72.03±0.36	**80.54±0.25**	**89.59±0.31**	74.90±0.37	92.51±0.16

优化函数

- GNN-LF/HF实验结果



(a) **Cora** (b) **Citeseer** (c) **Pubmed**



(a) **GNN-LF** (b) **GNN-HF**

问题

- 怎么通过优化函数刻画更多的模型，如GDC、GraphHeat、GAT？
- 优化函数是否可以刻画图神经网络的激活函数？
- 优化函数是否能够与不同的滤波器对应？

Graph Neural Networks Inspired by Classical Iterative Algorithms

(ICML, 2021)

Yongyi Yang, Tang Liu, Yangkun Wang, Jinjing Zhou,
Quan Gan, Zhewei Wei, Zheng Zhang, Zengfeng Huang, David Wipf

统一的GNN优化函数

- 从优化函数的视角理解GNN模型

$$\min_{\boldsymbol{z}} f(\boldsymbol{z}) = (1-\alpha)\boldsymbol{z}^T \gamma(\boldsymbol{L})\boldsymbol{z} + \alpha\|\boldsymbol{z}-\boldsymbol{x}\|_2^2$$

  - □ $\gamma(\boldsymbol{L})$ 为能量函数（Energy function），代表信号的传播速率；
  - □ 需要假设$\gamma(\boldsymbol{L})$ 是半正定的，即其特征值$\gamma(\lambda) \geq 0$。

- 通过令其导数等于零，可得最优解为：

$$\boldsymbol{z}^* = \alpha(\alpha\boldsymbol{I} + (1-\alpha)\gamma(\boldsymbol{L}))^{-1}\boldsymbol{x}$$

Yang Y, Liu T et al. Graph Neural Networks Inspired by Classical Iterative Algorithms. [ICML'2021]

统一的GNN优化函数

- 从优化函数的视角理解GNN模型

$$\min_{\boldsymbol{z}} f(\boldsymbol{z}) = (1-\alpha)\boldsymbol{z}^T \gamma(\boldsymbol{L})\boldsymbol{z} + \alpha\|\boldsymbol{z}-\boldsymbol{x}\|_2^2$$

  - □ 最优解：$\boldsymbol{z}^* = \alpha(\alpha\boldsymbol{I} + (1-\alpha)\gamma(\boldsymbol{L}))^{-1}\boldsymbol{x}$。

- 令$\gamma(\boldsymbol{L}) = e^{t\boldsymbol{L}} - \boldsymbol{I}$，$\alpha = 0.5$，优化函数的最优解：

$$\boldsymbol{z}^* = e^{-tL}\boldsymbol{x} = e^{-t(I-P)}\boldsymbol{x} = \sum_{k=0}^{\infty} e^{-t}\frac{t^k}{k!}\boldsymbol{P}^k\boldsymbol{x}$$

  - □ 可以用来解释 GDC 和 GraphHeat 使用的热核（Heat Kernel）。

Yang Y, Liu T et al. Graph Neural Networks Inspired by Classical Iterative Algorithms. [ICML'2021]

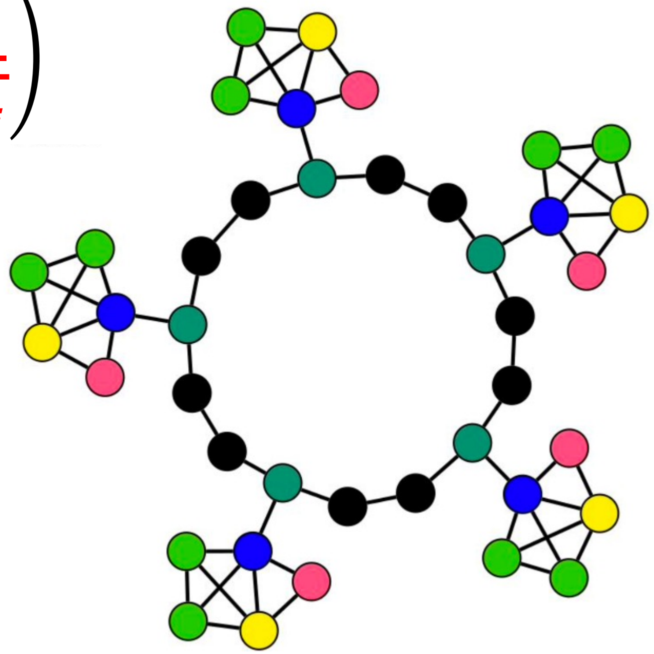
统一的GNN优化函数

- 从优化函数的视角理解GNN模型

$$\min_{\boldsymbol{z}} f(\boldsymbol{z}) = (1-\alpha)\boldsymbol{z}^T \gamma(\boldsymbol{L})\boldsymbol{z} + \alpha\|\boldsymbol{z}-\boldsymbol{x}\|_2^2$$

$\boxed{\text{当 } \gamma(\boldsymbol{L}) = 2\boldsymbol{I} - \boldsymbol{L}}$

$$\boldsymbol{z}^T(2\boldsymbol{I}+\boldsymbol{I})\boldsymbol{z} = \sum_{(i,j)\in E} A_{ij}\cdot\left(\frac{z_i}{\sqrt{d_i}} + \frac{z_j}{\sqrt{d_j}}\right)^2$$



- 解释异配性假设：
  - □ 邻居节点的特征具有负相关性

Yang Y, Liu T et al. Graph Neural Networks Inspired by Classical Iterative Algorithms. [ICML'2021]

统一的GNN优化函数

- 带非负正则项的优化函数

$$\min_{\boldsymbol{z}} \hat{f}(\boldsymbol{z}) = (1-\alpha)\boldsymbol{z}^T\gamma(\boldsymbol{L})\boldsymbol{z} + \alpha\|\boldsymbol{z}-\boldsymbol{x}\|_2^2 + \eta(\boldsymbol{z})$$

  - $\eta(\boldsymbol{z})$：**非负**正则项，给 $\boldsymbol{z}$ 中小于**0**的项一个正无穷的penalty。

- 近端梯度下降（proximal gradient descent）
  - 带$\eta(z)$正则项函数的优化方法
  - 解释激活函数：

$$\min_{\boldsymbol{z}} \hat{f}(\boldsymbol{z}) \xrightarrow[\eta(\boldsymbol{z}) = \sum_j I_\infty[z_i < 0]]{\gamma(\boldsymbol{L}) = \tilde{\boldsymbol{L}}} \boldsymbol{z}^{(k+1)} = \mathrm{ReLU}\left(\tilde{\boldsymbol{D}}^{-1/2}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}}^{-1/2}\boldsymbol{z}^{(k)}\right)$$

Yang Y, Liu T et al. Graph Neural Networks Inspired by Classical Iterative Algorithms. [ICML’2021]

统一的GNN优化函数

- 从优化函数的视角理解GNN模型

$$\min_{\boldsymbol{z}} f(\boldsymbol{z}) = (1-\alpha)\boldsymbol{z}^T\gamma(\boldsymbol{L})\boldsymbol{z} + \alpha\|\boldsymbol{z}-\boldsymbol{x}\|_2^2$$

最优解：$\boldsymbol{z}^* = \alpha(\alpha\boldsymbol{I} + (1-\alpha)\boxed{\gamma(\boldsymbol{L})})^{-1}\boldsymbol{x}$。

对拉普拉斯矩阵进行特征值分解

Yang Y, Liu T et al. Graph Neural Networks Inspired by Classical Iterative Algorithms. [ICML'2021]

统一的GNN优化函数

- 从优化函数的视角理解GNN模型

$$\min_{\boldsymbol{z}} f(\boldsymbol{z}) = (1-\alpha)\boldsymbol{z}^T\gamma(\boldsymbol{L})\boldsymbol{z} + \alpha\|\boldsymbol{z}-\boldsymbol{x}\|_2^2$$

最优解：$\boldsymbol{z}^* = \alpha(\alpha\boldsymbol{I} + (1-\alpha)\boxed{\gamma(\boldsymbol{L})})^{-1}\boldsymbol{x}$。

$$= \boldsymbol{U}diag\left[\frac{\alpha}{\alpha+(1-\alpha)\gamma(\lambda_1)},\dots,\frac{\alpha}{\alpha+(1-\alpha)\gamma(\lambda_n)}\right]\boldsymbol{U}^T\boldsymbol{x}$$

$$\Downarrow \qquad \Downarrow$$

滤波器：$h(\lambda) = \frac{\alpha}{\alpha+(1-\alpha)\gamma(\lambda)}$

Yang Y, Liu T et al. Graph Neural Networks Inspired by Classical Iterative Algorithms. [ICML'2021]

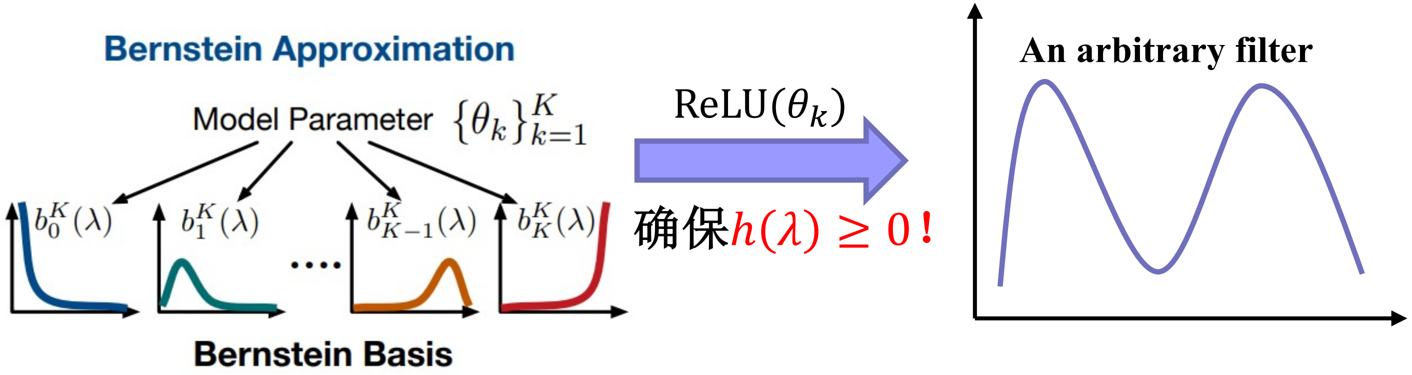
统一的GNN优化函数

- 优化函数的最优解 $\boldsymbol{z}^* = \alpha(\alpha \boldsymbol{I} + (1-\alpha)\gamma(\boldsymbol{L}))^{-1}\boldsymbol{x}$

- 因为$\gamma(\boldsymbol{L})$半正定，有$\gamma(\lambda) \geq 0$， $\lambda \in [0,2]$，
  - 解释理想的滤波器范围：

$$0 \leq h(\lambda) = \frac{\alpha}{\alpha+(1-\alpha)\gamma(\lambda)} \leq 1, \quad \forall \lambda \in [0,2]$$

  - 对应BernNet：



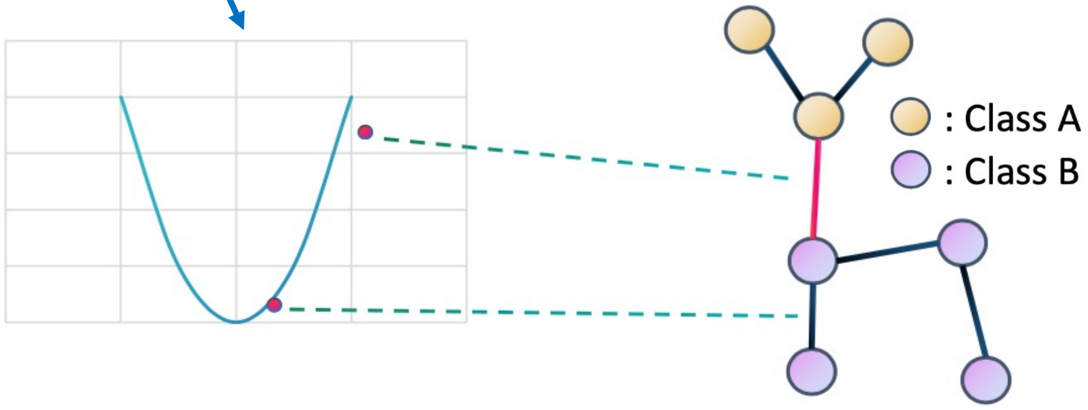
Yang Y, Liu T et al. Graph Neural Networks Inspired by Classical Iterative Algorithms. [ICML'2021]

统一的GNN优化函数

- 带鲁棒正则化(Robust Regularization)的优化函数

$$\min_{\boldsymbol{z}} \acute{f}(\boldsymbol{z}) = (1-\alpha)\sum_{(i,j)\in E} \rho\left[A_{ij}\left(\frac{z_i}{\sqrt{d_i}} - \frac{z_j}{\sqrt{d_j}}\right)^2\right] + \alpha\|\boldsymbol{z}-\boldsymbol{x}\|_2^2$$

对应不同边的权重



Yang Y, Liu T et al. Graph Neural Networks Inspired by Classical Iterative Algorithms. [ICML'2021]

统一的GNN优化函数

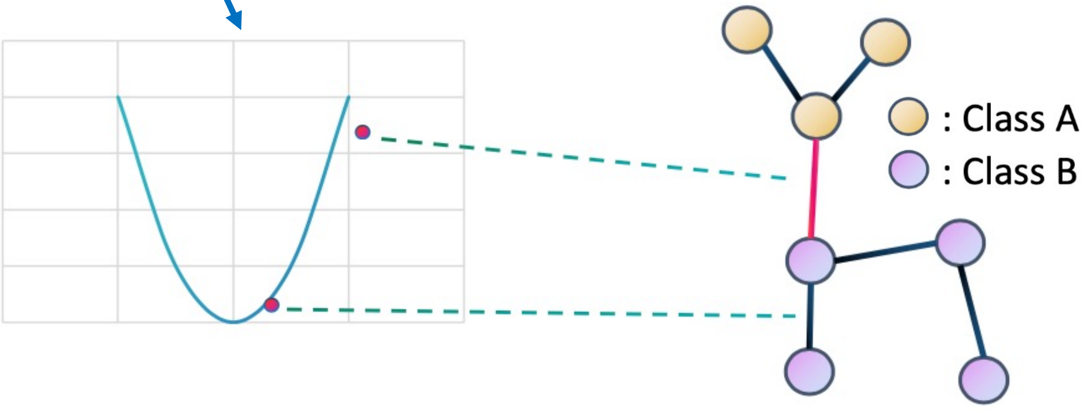
- 带鲁棒正则化(Robust Regularization)的优化函数

$$\min_{\boldsymbol{z}} \acute{f}(\boldsymbol{z}) = (1-\alpha)\sum_{(i,j)\in E} \rho\left[A_{ij}\left(\frac{z_i}{\sqrt{d_i}} - \frac{z_j}{\sqrt{d_j}}\right)^2\right] + \alpha\|\boldsymbol{z}-\boldsymbol{x}\|_2^2$$

ρ函数决定边的Smooth程度，
e.g. $\rho(x) = x^{1/2}$

对应不同边的权重

$\rho(c^2)$



记$c = A_{ij}\left(\frac{z_i}{\sqrt{d_i}} - \frac{z_j}{\sqrt{d_j}}\right)$　　c^2

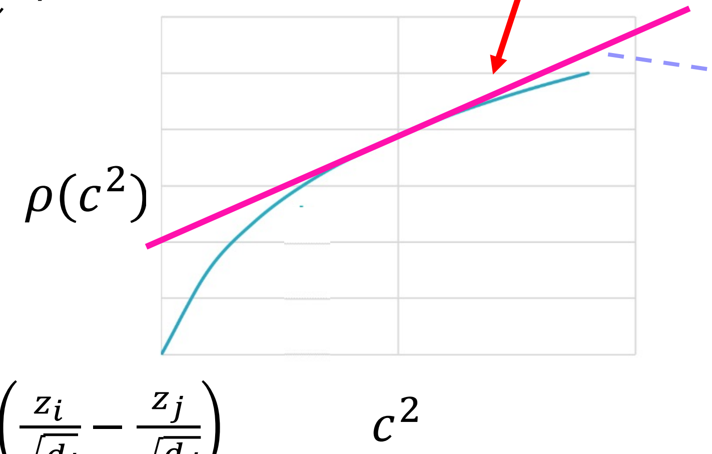
Yang Y, Liu T et al. Graph Neural Networks Inspired by Classical Iterative Algorithms. [ICML'2021]

统一的GNN优化函数

- 带鲁棒正则化(Robust Regularization)的优化函数

$$\min_{\boldsymbol{z}} \acute{f}(\boldsymbol{z}) = (1-\alpha) \sum_{(i,j)\in E} \rho \left[A_{ij} \left(\frac{z_i}{\sqrt{d_i}} - \frac{z_j}{\sqrt{d_j}} \right)^2 \right] + \alpha \|\boldsymbol{z} - \boldsymbol{x}\|_2^2$$

ρ函数决定边的Smooth程度，
e.g. $\rho(x) = x^{1/2}$



记$c = A_{ij} \left(\frac{z_i}{\sqrt{d_i}} - \frac{z_j}{\sqrt{d_j}} \right)$

$$\sum_{(i,j)\in E} h_{ij} \left[A_{ij} \left(\frac{z_i}{\sqrt{d_i}} - \frac{z_j}{\sqrt{d_j}} \right)^2 \right] + \tilde{\rho}(h_{ij})$$

对偶变量h_{ij}：图注意力 (graph attention) 权重
$\tilde{\rho}(h_{ij})$：concave conjugate

Yang Y, Liu T et al. Graph Neural Networks Inspired by Classical Iterative Algorithms. [ICML'2021]

统一的GNN优化函数

- 带鲁棒正则化(Robust Regularization)的优化函数

$$\min_{\boldsymbol{z}} \acute{f}(\boldsymbol{z}) = (1-\alpha)\sum_{(i,j)\in E} \rho\left[A_{ij}\left(\frac{z_i}{\sqrt{d_i}}-\frac{z_j}{\sqrt{d_j}}\right)^2\right] + \alpha\|\boldsymbol{z}-\boldsymbol{x}\|_2^2$$

ρ函数决定边的Smooth程度，
e.g. $\rho(x) = x^{1/2}$

$h_{ij} = \frac{\partial \rho(c^2)}{\partial c^2}$对应了注意力权重

记$c = A_{ij}\left(\frac{z_i}{\sqrt{d_i}}-\frac{z_j}{\sqrt{d_j}}\right)$

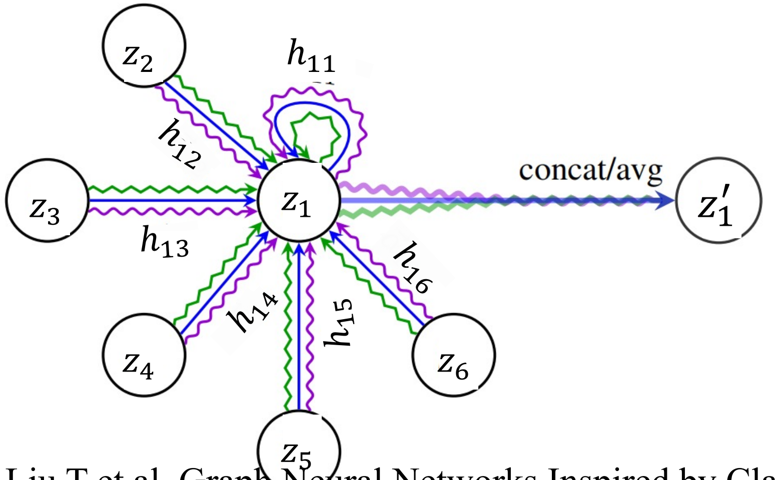
Yang Y, Liu T et al. Graph Neural Networks Inspired by Classical Iterative Algorithms. [ICML'2021]

统一的GNN优化函数

- 带鲁棒正则化(Robust Regularization)的优化函数

$$\min_{\mathbf{z}} \acute{f}(\mathbf{z}) = (1-\alpha)\sum_{(i,j)\in E} \rho\left[A_{ij}\left(\frac{z_i}{\sqrt{d_i}} - \frac{z_j}{\sqrt{d_j}}\right)^2\right] + \alpha\|\mathbf{z}-\mathbf{x}\|_2^2$$

□ $\rho\left[A_{ij}\left(\frac{z_i}{\sqrt{d_i}} - \frac{z_j}{\sqrt{d_j}}\right)^2\right]$：边缘不确定性，对应注意力机制，解释GAT



$$c = A_{ij}\left(\frac{z_i}{\sqrt{d_i}} - \frac{z_j}{\sqrt{d_j}}\right)$$

图注意力权重：$h_{ij} = \frac{\partial\rho(c^2)}{\partial c^2}$

Yang Y, Liu T et al. Graph Neural Networks Inspired by Classical Iterative Algorithms. [ICML'2021]

统一的GNN优化函数

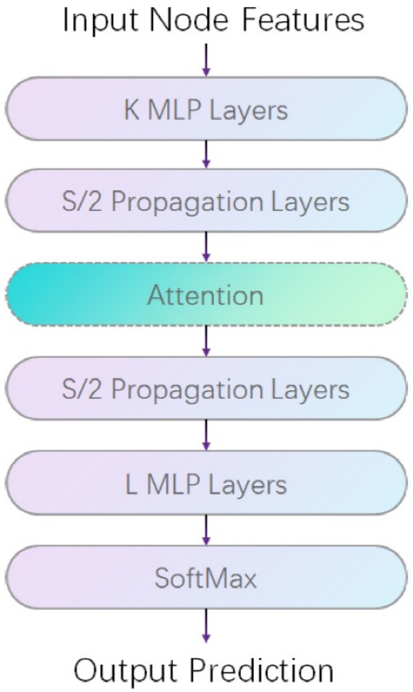
- 带鲁棒正则化(Robust Regularization)的优化函数

$$\min_{\boldsymbol{z}} \acute{f}(\boldsymbol{z}) = (1-\alpha)\sum_{(i,j)\in E} \rho\left[A_{ij}\left(\frac{z_i}{\sqrt{d_i}}-\frac{z_j}{\sqrt{d_j}}\right)^2\right] + \alpha\|\boldsymbol{z}-\boldsymbol{x}\|_2^2$$

- 使用迭代重加权最小二乘算法解该优化问题
  - Iterative reweighted least squares
  - 新的GNN模型**TWIRLS**

$\rho(c^2)$	$\partial\rho(c^2)/\partial c^2$	Range
$\log(c^2+\epsilon)$	$1/c^2+\epsilon$	$(0,1/\epsilon]$
$\begin{cases} c^2 & c<\tau \\ \tau & c\geq\tau \end{cases}$	$\begin{cases} 1 & c<\tau \\ 0 & c\geq\tau \end{cases}$	$\{0,1\}$

$\min_{\boldsymbol{z}} \acute{f}(\boldsymbol{z})$ →



Yang Y, Liu T et al. Graph Neural Networks Inspired by Classical Iterative Algorithms. [ICML'2021]

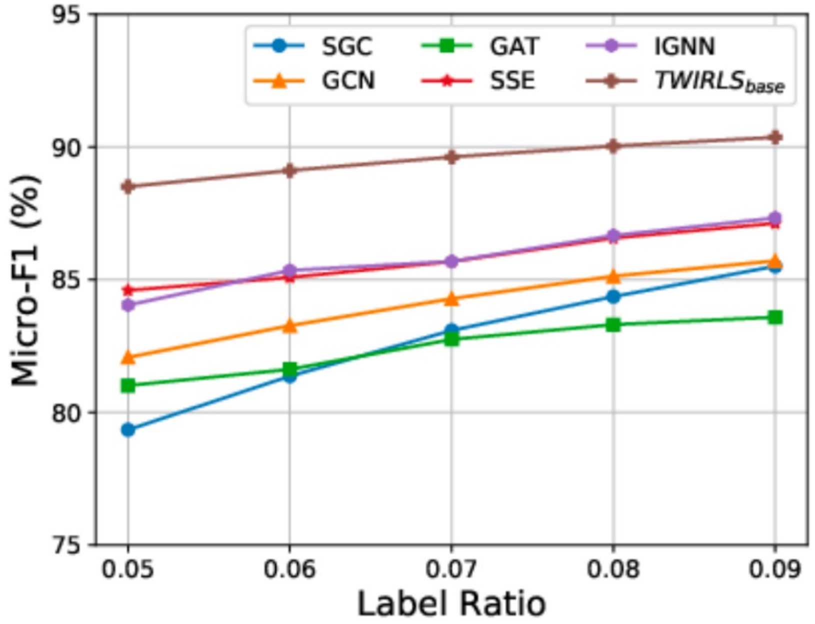
TWIRLS：实验结果

- 半监督节点分类

MODEL	CORA	CITESEER	PUBMED	ARXIV
SGC	81.7 ± 0.1	71.3 ± 0.2	78.9 ± 0.1	69.79 ± 0.16
GCN	81.5	71.1	79.0	71.74 ± 0.29
APPNP	83.3	71.8	80.1	71.74 ± 0.29
JKNET	81.1	69.8	78.1	72.19 ± 0.21
GCNII	**85.5 ± 0.5**	73.4 ± 0.6	80.3 ± 0.4	72.74 ± 0.16
DAGNN	84.4 ± 0.5	73.3 ± 0.6	80.5 ± 0.5	72.09 ± 0.25
$\text{TWIRLS}_{\text{BASE}}$	84.1 ± 0.5	**74.2 ± 0.45**	**80.7 ± 0.5**	**72.93 ± 0.19**

- 稀疏标签

（Amazon co-purchase)



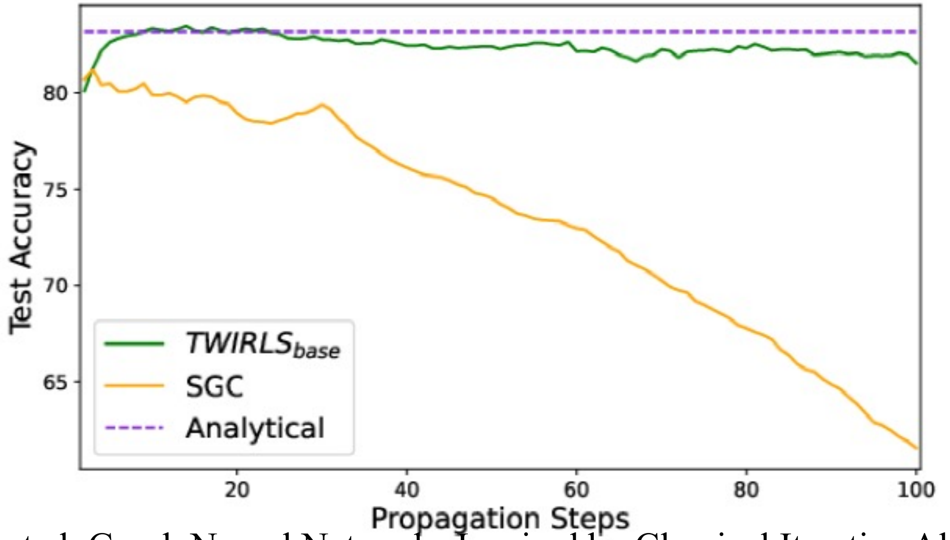
Yang Y, Liu T et al. Graph Neural Networks Inspired by Classical Iterative Algorithms. [ICML'2021]

TWIRLS：实验结果

- 异配图数据集

DATASET	TEXAS	WISCONSIN	ACTOR	CORNELL
HOM. RATIO ($\mathcal{H}$)	0.11	0.21	0.22	0.3
GCN	59.46±5.25	59.80±6.99	30.26±0.79	57.03±4.67
GAT	58.38±4.45	55.29±8.71	26.28±1.73	58.92±3.32
GRAPHSAGE	82.43±6.14	81.18±5.56	34.23±0.99	75.95±5.01
GEOM-GCN	67.57	64.12	31.63	60.81
H_2GCN	**84.86±6.77**	86.67±4.69	35.86±1.03	82.16±4.80
MLP	81.89±4.78	85.29±3.61	35.76±0.98	81.08±6.37
TWIRLS$_{\text{BASE}}$	81.62±5.51	82.75±7.83	37.10±1.07	83.51±7.30
TWIRLS	84.59±3.83	**86.67±4.19**	**37.43±1.50**	**86.76±5.05**

- 避免过平滑



Yang Y, Liu T et al. Graph Neural Networks Inspired by Classical Iterative Algorithms. [ICML'2021]

报告提纲

- 图神经网络的应用和概述
- 图神经网络的三个视角
  - □ 滤波器 ⇨ 学习任意的滤波器
  - □ 随机游走 ⇨ 基于重启随机游走的深度GNN
  - □ 优化函数 ⇨ 统一的GNN优化函数
- 展望与总结

Take home message

- 从三个不同的视角对图神经网络进行了解释；
  - 滤波器：GCN是一个低通滤波器，层数叠加会过平滑；
  - 随机游走：GCN随着随机游走会趋于稳态，导致过平滑；
  - 优化函数：GCN会使得狄利克雷能量趋于零，导致过平滑；
- Take home message：
  - 如果要设计任意的滤波器，可以用BernNet：Bernstein多项式；
  - 如果要设计深层神经网络，可以用GCNII: Initial residual和Identity mapping；
  - 如果要解释更复杂的GNN，可以用带有$\gamma(\boldsymbol{L})$的优化函数。

团队成员和合作者

- 团队成员


Zhewei Wei


Zhe Yuan


Hanzhi Wang

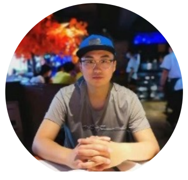

Yanping Zheng

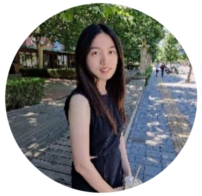

Mingguo He


Jiajun Li


Tianjing Zeng


Fangrui Lv


Gengmo Zhou


Yuhe Guo


Jinjia Feng


Ruoqi Zhang


Xu Liu


Mingji Yang

- 合作者


Zengfeng Huang


Hongteng Xu


Bolin Ding


Yaliang Li

谢谢！
Q&A