Beyond Homophily in Graph Neural Networks: Current Limitations and Effective Designs

NIPS 2020

2020-12

1 Introduction

- 这篇文章主要是探究graph中的异质偏好Heterophily和同质偏好Homophily
- Homophily：同质偏好，指个体更倾向于与相似的个体建立链结（即“物以类聚，人以群分”）
- Heterophily：异质偏好，指个体倾向于和相异的个体链接（“异性相吸”）
- 上面这两点特性体现在graph上，就是单个edge链接的两个node，是倾向于属于同一类node还是不同类node。比如在citing network中，paper和paper都是同一类，甚至是同一area；但是在dating network中，主要是不同性别的node相连。

1 Introduction

- 在GCN等很多GNN中，实际上假设了graph是倾向于同质偏好的，以GCN举例，它通过邻居信息的加权求和得到新的表示，这一机制的背后本质是对于不同的邻居都进行了相同的处理。

- 作者发现GCN，GAT这些GNN，擅长处理具有高homophily的graph，但是在高heterophily的graph下，性能甚至可能不如直接基于特征的MLP

1 Introduction

- 辨析Heterophily与Heterogeneity
- 同质（Homogeneity）与异质（Heterogeneity）是一对科学与统计学中常用的对物质的均匀性进行描述的概念。
- 如果一种材料或一幅图像具有同质性（或者说是同质的），那么它就是由同样的单元堆砌而成的。相对的，如果一种物质至少一种特征的分布明显不均匀，那么它就具有异质性。
- 在graph中，如果node和edge各自具有不同的类型，那么就可以说是具有Heterogeneity。在一个异质图中，异质偏好可能高也可能低。而如果具有了异质偏好，那首先肯定是异质图。

2 Learning Over Networks with Heterophily

- 本文作者在前面研究的基础上，综合了几种不同的设计思路，提出了3种有助于建模异质偏好的设计模式。
- Design 1: Ego- and Neighbor-embedding Separation

$$\mathbf{r}_v^{(k)} = \texttt{COMBINE}\left(\mathbf{r}_v^{(k-1)}, \texttt{AGGR}(\{\mathbf{r}_u^{(k-1)} : u \in \bar{N}(v)\})\right)$$

- 自身信息与邻居信息分离。COMBINE是一种不mixing邻居信息和自身信息的操作，最简单的是concat，这一设计方案已在GraphSAGE中采用。AGGR可以是GCN中的加权求和。
- Intuition: 如果是异质偏好很高的网络，那么一个node和邻居node的特征大概率是不同的，那么直接相加相当于混淆了自身的特性

2 Learning Over Networks with Heterophily

- Design 2: Higher-order Neighborhoods

$$\mathbf{r}_v^{(k)} = \texttt{COMBINE}\left(\mathbf{r}_v^{(k-1)}, \texttt{AGGR}(\{\mathbf{r}_u^{(k-1)} : u \in N_1(v)\}), \texttt{AGGR}(\{\mathbf{r}_u^{(k-1)} : u \in N_2(v)\}), \ldots\right)$$

- 在前面设计的基础上，在一次信息聚合的过程中，同时聚合多阶邻居信息。这个设计在MixHop和GCN-Cheby中采用。
- Intuition: 一个图涉及到的不同order的邻居提供的异质/同质信息是不同的，对于同质性强的graph，不同阶的邻居都能够为中心节点提供同质的信息；对于异质性强的graph，1阶邻居倾向于是异质的，而2阶邻居倾向于提供同质信息，这种同质信息就能够帮助进行中心节点的预测任务。
- Theorem：实际在论文中，作者证明了在一种理想的情况下，2阶邻居总是倾向于同质的。

2 Learning Over Networks with Heterophily

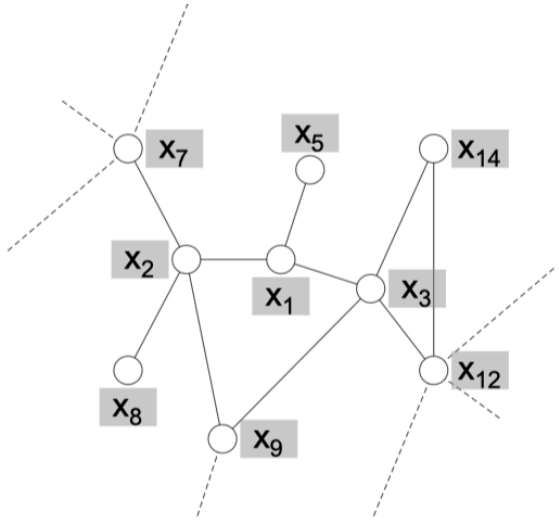
- Design 3: Combination of Intermediate Representations

$$\mathbf{r}_v^{(\text{final})} = \texttt{COMBINE}\left(\mathbf{r}_v^{(1)}, \mathbf{r}_v^{(2)}, \ldots, \mathbf{r}_v^{(K)}\right)$$

- 多次聚合结果联合。在Design 2的基础上，每轮round的传播结果都提供到最终的表示上。这种设计方案在Jumping Knowledge Network中提出。
- Intuition: 每一轮的消息传播，得到的是不同range的局部信息，传播次数越多，看到的“视野”越大。收集不同轮的传播结果，然后拼接起来产生最终的输出。
- Theorem：对于GCN来说，每一次的传播是不同pass的信息处理过程，不同的传播过程得到不同frequency的结果。

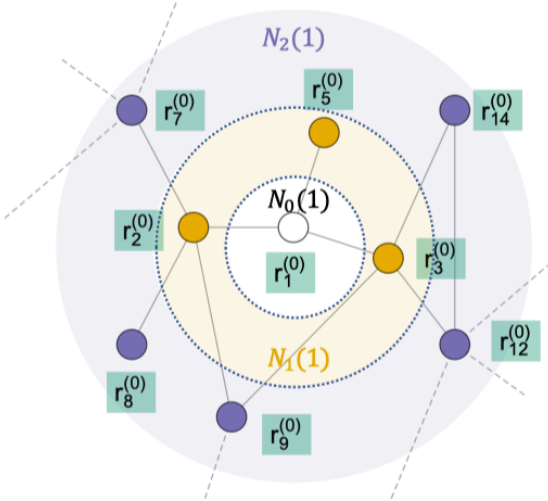
H_2GNN

将3种design结合起来，得到本文用作实验对比的模型



(S1)

$r_v^{(0)} = \sigma(x_v W_e)$

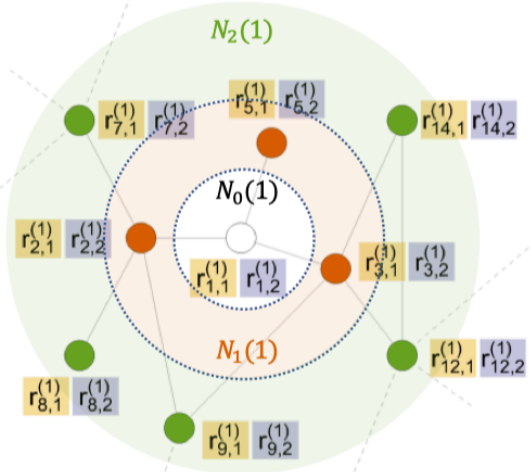


$r_{v,1}^{(1)} = \text{weighted\_avg}(\{r_u^{(0)} : u \in N_1(v)\})$

$r_{v,2}^{(1)} = \text{weighted\_avg}(\{r_u^{(0)} : u \in N_2(v)\})$

$r_v^{(1)} = (r_{v,1}^{(1)} || r_{v,2}^{(1)})$

(S2-Round 1)

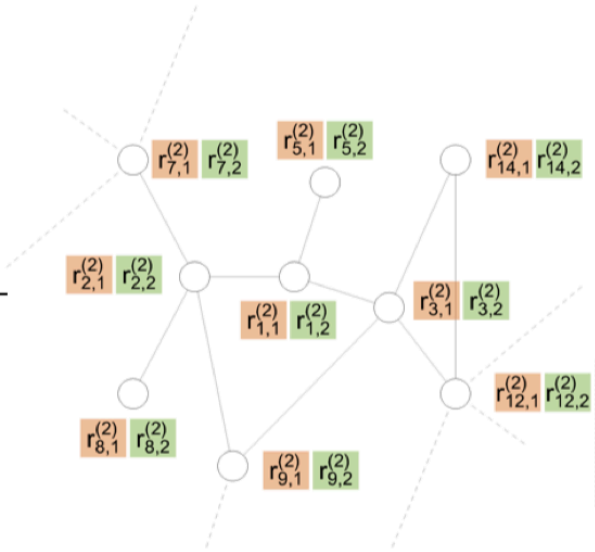


(S2-Round 2)

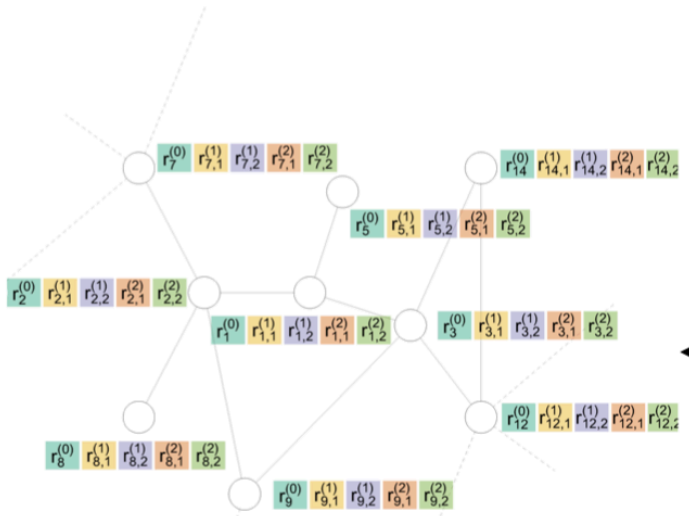
$r_{v,1}^{(2)} = \text{weighted\_avg}(\{r_u^{(1)} : u \in N_1(v)\})$

$r_{v,2}^{(2)} = \text{weighted\_avg}(\{r_u^{(1)} : u \in N_2(v)\})$

$r_v^{(2)} = (r_{v,1}^{(2)} || r_{v,2}^{(2)})$



(S3)



$r_v^{(\text{final})} = (r_v^{(0)} || r_v^{(1)} || r_v^{(2)})$

3 Empirical Evaluation

- 两大方面的实验
  - Evaluation on Synthetic Benchmarks
  - Evaluation on Real Benchmarks
- 在人造的数据集上，构造了不同edge homophily比例的数据集

$$h = \frac{|\{(u,v):(u,v)\in\mathcal{E}\wedge y_u=y_v\}|}{|\mathcal{E}|}$$

Benchmark Name	#Nodes $\lvert\mathcal{V}\rvert$	#Edges $\lvert\mathcal{E}\rvert$	#Classes $\lvert\mathcal{Y}\rvert$	#Features F	Homophily h	#Graphs
`syn-cora`	1,490	2,965 to 2,968	5	`cora` [30, 39]	$[0, 0.1, \ldots, 1]$	33 (3 per h)
`syn-products`	10,000	59,640 to 59,648	10	`ogbn-products` [13]	$[0, 0.1, \ldots, 1]$	33 (3 per h)

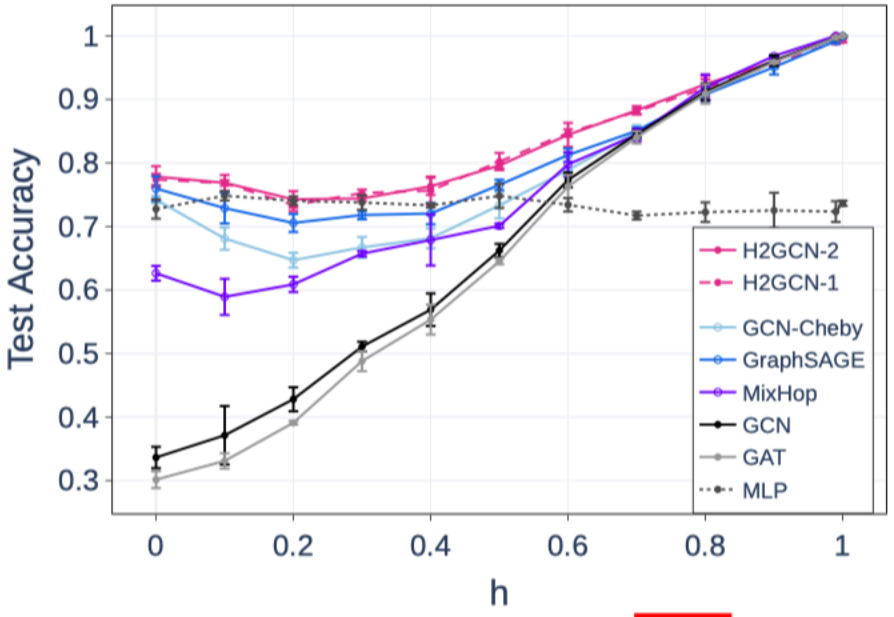
3.1 Evaluation on Synthetic Benchmarks

- 在syn-cora数据集上，评估不同的GNN
- 结果：
  - 在异质偏好高h=0.1的情况下，GCN这些方法甚至不如MLP
  - 在异质偏好低h=0.7的情况下，GCN表现出了较好的performance
  - 作者的模型总能取得较好的结果

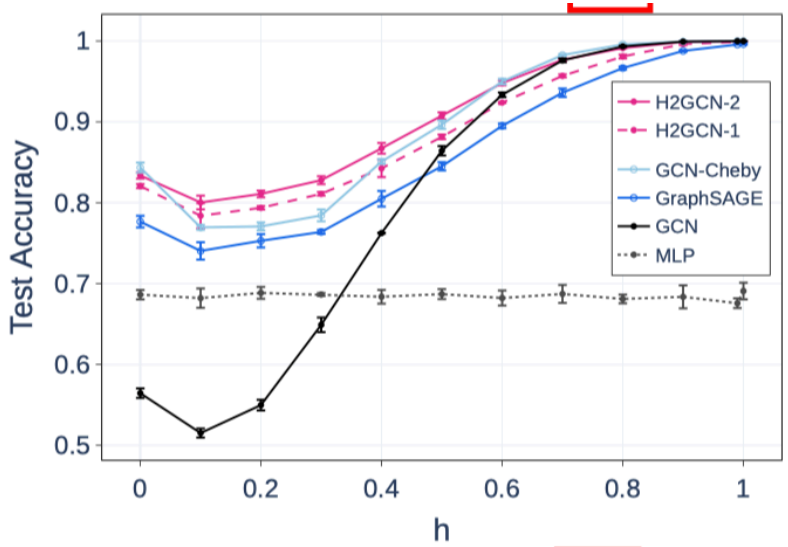
	$h = \mathbf{0.1}$	$h = \mathbf{0.7}$
GCN [17]	$37.14_{\pm 4.60}$	$84.52_{\pm 0.54}$
GAT [36]	$33.11_{\pm 1.20}$	$84.03_{\pm 0.97}$
GCN-Cheby [7]	$68.10_{\pm 1.75}$	$84.92_{\pm 1.03}$
GraphSAGE [11]	$72.89_{\pm 2.42}$	$85.06_{\pm 0.51}$
MixHop [1]	$58.93_{\pm 2.84}$	$84.43_{\pm 0.94}$
MLP	$74.85_{\pm 0.76}$	$71.72_{\pm 0.62}$
H_2GCN **(ours)**	$\mathbf{76.87}_{\pm \mathbf{0.43}}$	$\mathbf{88.28}_{\pm \mathbf{0.66}}$

3.1 Evaluation on Synthetic Benchmarks

- 在syn数据集上，随着异质偏好程度降低的变化
- 结果：
  - 随着异质程度降低，模型效果逐渐提升；对应的MLP不变化



(a) `syn-cora` (Table G.2)



(b) `syn-products` (Table G.3). MixHop acc $< 30\%$; GAT acc $< 50\%$ for $h < 0.4$.

3.2 Evaluation on Real Benchmarks

- 在实际数据集上的表现

	Texas	Wisconsin	Actor	Squirrel	Chameleon	Cornell	Cora Full	Citeseer	Pubmed	Cora
Hom. ratio h	**0.11**	**0.21**	**0.22**	**0.22**	**0.23**	**0.3**	**0.57**	**0.74**	**0.8**	**0.81**
#Nodes $\|\mathcal{V}\|$	183	251	7,600	5,201	2,277	183	19,793	3,327	19,717	2,708
#Edges $\|\mathcal{E}\|$	295	466	26,752	198,493	31,421	280	63,421	4,676	44,327	5,278
#Classes $\|\mathcal{Y}\|$	5	5	5	5	5	5	70	7	3	6
H_2GCN-1	84.86±6.77	86.67±4.69	35.86±1.03	36.42±1.89	57.11±1.58	82.16±4.80	68.13±0.49	77.07±1.64	89.40±0.34	86.92±1.37
H_2GCN-2	82.16±5.28	85.88±4.22	35.62±1.30	37.90±2.02	59.39±1.98	82.16±6.00	69.05±0.37	76.88±1.77	89.59±0.33	87.81±1.35
GraphSAGE	82.43±6.14	81.18±5.56	34.23±0.99	41.61±0.74	58.73±1.68	75.95±5.01	65.14±0.75	76.04±1.30	88.45±0.50	86.90±1.04
GCN-Cheby	77.30±4.07	79.41±4.46	34.11±1.09	43.86±1.64	55.24±2.76	74.32±7.46	67.41±0.69	75.82±1.53	88.72±0.55	86.76±0.95
MixHop	77.84±7.73	75.88±4.90	32.22±2.34	43.80±1.48	60.50±2.53	73.51±6.34	65.59±0.34	76.26±1.33	85.31±0.61	87.61±0.85
GraphSAGE+JK	83.78±2.21	81.96±4.96	34.28±1.01	40.85±1.29	58.11±1.97	75.68±4.03	65.31±0.58	76.05±1.37	88.34±0.62	85.96±0.83
Cheby+JK	78.38±6.37	82.55±4.57	35.14±1.37	45.03±1.73	63.79±2.27	74.59±7.87	66.87±0.29	74.98±1.18	89.07±0.30	85.49±1.27
GCN+JK	66.49±6.64	74.31±6.43	34.18±0.85	40.45±1.61	63.42±2.00	64.59±8.68	66.72±0.61	74.51±1.75	88.41±0.45	85.79±0.92
GCN	59.46±5.25	59.80±6.99	30.26±0.79	36.89±1.34	59.82±2.58	57.03±4.67	68.39±0.32	76.68±1.64	87.38±0.66	87.28±1.26
GAT	58.38±4.45	55.29±8.71	26.28±1.73	30.62±2.11	54.69±1.95	58.92±3.32	59.81±0.92	75.46±1.72	84.68±0.44	82.68±1.80
GEOM-GCN*	67.57	64.12	31.63	38.14	60.90	60.81	N/A	77.99	90.05	85.27
MLP	81.89±4.78	85.29±3.61	35.76±0.98	29.68±1.81	46.36±2.52	81.08±6.37	58.76±0.50	72.41±2.18	86.65±0.35	74.75±2.22

4 Conclusion

- 这篇文章研究的3中设计都是已经在之前的论文提出。但是没有理论分析。这篇文章集中在研究异质偏好性，讨论了不同的设计方法导致的GNN的学习异质偏好性能力的偏差
- 工作很充实（10页正文+17页附录）
- 这三种设计方案实际有一个本质的思想，不盲目的混合输入，将能够收集的所有信息直接拼接起来，得到输出
- 缺点在于，由于仍然是直接加权求和邻居，无法分辨邻居之间的区别

个人实验计划

- ✓ 基础代码结构
- ✓ 原始R-GCN的实现
- ○ 分batch训练graph，并且修改评估打分方法
- ○ 设计更加复杂的消息函数，让不同的邻居进行不同的消息构造方式，考虑先统一将邻居投影到不同的关系空间中，然后进行卷积等操作，再投影到消息空间进行聚合