在现实世界的超图中，超边是如何重叠的？
——模式，测量指标与生成

任泽华

2021年11月2日

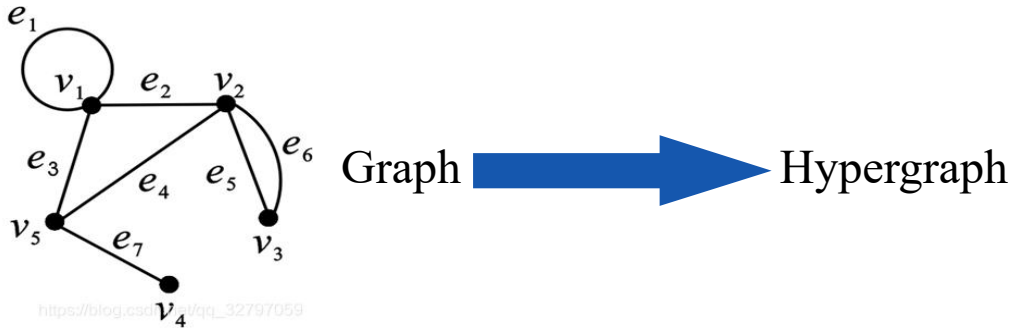
Geon Lee, Minyoung Choe, and Kijung Shin. 2021. How Do Hyperedges Overlap in Real-World Hypergraphs? - Patterns, Measures, and Generators. InProceedings of the Web Conference 2021 (WWW ’ 21), April 19–23, 2021, Ljubljana, Slovenia.ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3442381.3450010

一、背景与现状

1、问题背景

- **超图（hypergraph）**：是图（graph）的扩展，图建立了对象之间的关联关系，但仅在两个对象间连接边，无法表征更复杂的高阶关系（如多人合著）。于是提出了**超边（hyperedge）**可以同时连接两个以上的多个节点，由节点和超边组成的广义的图被叫做超图。超边之间拥有共同的节点被称为“**重叠**”。
- **现实中超图的特点**：（区别于随机生成超图）① 实质性：每个节点的自我网络中超边重叠性比随机网络更大。② 重尾性：所有二元、三元节点组上重叠超边的分布呈现重尾特性，近似于幂律分布。 ③ 亲同性：每个超边所包含的节点在结构上往往比随机图中的节点更相似。

超边如何重叠？有什么普遍的特征吗？什么潜在的过程会导致这样的模式？



一、背景与现状

2、研究现状

- **超图应用**：超图被广泛应用于各种领域，如计算机视觉、生物信息学、电路设计、社会网络分析和推荐算法。它被用于各种分析和学习任务，包括分类、聚类和超边关联预测等。
- **超图生成**：当我们无法获取广泛关联数据构建超图时，需要根据已有超图生成。或当我们需要验证某种算法是否有效时，需要构建大量相似的超图。

- Benson 等人研究了单纯闭包事件（一组超边节点在另一组中完全包含），从而探究超图的局部特征。他们考虑了现实超图中的序列(即相互关联的时序超边)，并指出现**实世界超图是每个序列中二元节点和三元节点上重叠的超边的数量往往比在随机超图模型中更大**。
- Do 等人将一个真实世界的超图投影到多个成对图中，发现了超边的**重尾分布**，同时提出了一个超图生成器：**HYPERPA** 。每个节点被选择的概率与超边包含的子节点集大小成正比。
- Kook 等人发现了现实超图重叠超边的比例和直径随时间逐渐减小，而超边的数量比节点的数量增加得更快。使用新定义的森林火灾传播模式开发了超图生成器：**HYPERFF**。
- Lee 等人研究了超图的模体（motif）并提出了**26种基本模体**，发现同一领域的超图模体特别相似。

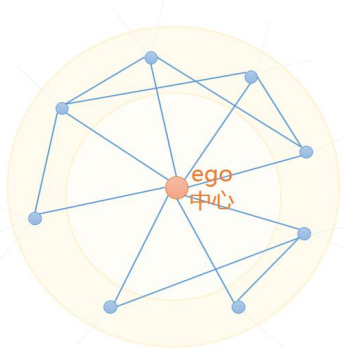
二、算法讲解

1、量测指标 （1）每个节点自我网络的重叠度

■ 超图自我网络：每个节点所在的超边及其超边包含节点之间的超边构成的子网络

➢ 原始关于重叠程度的定义：**密度**（density）

$$\rho(\mathcal{E}) := \frac{|\mathcal{E}|}{|\bigcup_{e\in\mathcal{E}} e|}.$$ 自我网络超边数/节点数



➢ 要表示超边重叠程度，必须满足：

1. 超边大小（每条超边节点数）相同、总节点数相同，超边越多越重叠。
2. 超边数相同，超边大小相同，节点数量越少越重叠。
3. 超边数相同，节点数相同，超边大小越大越重叠。（密度指标不满足）

➢ 重叠性的定义：**重叠度**（overlapness） $o(\mathcal{E}) := \frac{\sum_{e\in\mathcal{E}} |e|}{|\bigcup_{e\in\mathcal{E}} e|}.$ 超边大小之和/节点数

二、算法讲解

（2）二元、三元节点组重叠水平

- 给定一对或三对节点，有多少超边在它们上重叠？

$$d^{(2)}(\{i,j\}) := |E_{\{i,j\}}| \qquad d^{(3)}(\{i,j,k\}) := |E_{\{i,j,k\}}|$$

（3）每条超边的同质度

- 每条超边中节点的相似程度

$$homogeneity(e) := \begin{cases} \dfrac{\sum_{\{u,v\}\in\binom{e}{2}} |E_{\{u,v\}}|}{\binom{|e|}{2}}, & \text{if } |e| > 1 \\ 0, & \text{otherwise,} \end{cases}$$

$\binom{|e|}{2}$ 是节点对的集合

$|E_{\{u,v\}}|$ 是重叠的超边的数目

两个节点之间的结构相似性是根据在它们上重叠的超边的数量来测量的

二、算法讲解

2、超图生成算法——随机法HYPERCL

Algorithm 1: HyperCL: Random Hypergraph Generator

Input : (1) distribution of hyperedge sizes $\{s_1, \ldots, s_{|E|}\}$
(2) distribution of node degrees $\{d_1, \ldots, d_{|V|}\}$

Output: random hypergraph $\tilde{G} = (\tilde{V}, \tilde{E})$

```
1 Ṽ ← V and Ẽ ← ∅
2 for each i = 1, ..., |E| do
3     ẽ_i ← ∅
4     while |ẽ_i| < s_i do
5         v ← select a node with prob. proportional to the
            degree
6         ẽ_i ← ẽ_i ∪ {v}
7     Ẽ ← Ẽ ∪ {ẽ_i}
8 return G̃ = (Ṽ, Ẽ)
```

输入：原超图每条超边大小
　　原超图每个节点度数
输出：生成的随机超图
初始化：节点、空超边集合
遍历每条超边（总数已知）
循环添加节点，以节点在原图中的度为正比的概率选取
把生成超边加入超边集合
返回随机超图

二、算法讲解

2、超图生成算法——HYPERLAP（多层HYPERCL）

输入增加：HYPERCL层数L，每层权值w

Algorithm 2: HyperLap: Realistic Hypergraph Generator

Input : (1) distribution of hyperedge sizes $\{s_1, \cdots, s_{|E|}\}$
(2) distribution of node degrees $\{d_1, \cdots, d_{|V|}\}$
(3) number of levels L ($\leq \log_2 |V|$)
(4) weights of each level $\{w_1, \cdots, w_L\}$

Output: synthetic hypergraph $\hat{G} = (\hat{V}, \hat{E})$

/* Initialization */
$\hat{V} \leftarrow \{1, \cdots, |V|\}$ and $\hat{E} \leftarrow \emptyset$
/* Hierarchical Node Partitioning */
$S_1^{(L)}, \cdots, S_{2^{L-1}}^{(L)} \leftarrow$ uniformly partition $\hat{V}$ into 2^{L-1} groups
for each level $\ell = L-1, \cdots, 1$ **do**
  for each group $g = 1, \cdots, 2^{\ell-1}$ **do**
    $S_g^{(\ell)} = S_{2g-1}^{(\ell+1)} \cup S_{2g}^{(\ell+1)}$
/* Hyperedge Generation */
for each $i = 1, \cdots, |E|$ **do**
  $\ell \leftarrow$ select a level with prob. proportional to the weight
  $S_g^{(\ell)} \leftarrow$ select a group at level ℓ uniformly at random
  $\hat{e}_i \leftarrow \emptyset$
  while $|\hat{e}_i| < s_i$ **do**
    $v \leftarrow$ select a node from $S_g^{(\ell)}$ with prob. proportional to the degree
    $\hat{e}_i = \hat{e}_i \cup \{v\}$
  $\hat{E} = \hat{E} \cup \{\hat{e}_i\}$
return $\hat{G} = (\hat{V}, \hat{E})$

将节点按层次平均分组，组数为 $2^{\ell-1}$
（层次越高组数越多，每组节点越少）

如HYPERCL，但仅在规定组内选取节点
选取的组是均匀随机选取的，不同的是
选取分组的层次按照每层权值正比随机产生

二、算法讲解

2、超图生成算法——HYPERLAP+（自动参数选择）

Algorithm 3: HyperLap$^+$: Automatic Parameter Selection

Input : (1) input hypergraph $G = (V, E)$
(2) update resolution p
Output : synthetic hypergraph $\hat{G} = (\hat{V}, \hat{E})$

$\hat{G} = (\hat{V}, \hat{E}) \leftarrow$ run HyperCL using the distributions in G
for each level $\ell = 2, \cdots, L$ **do**
  $i^* \leftarrow \arg\min_{i \in \{1, \cdots, 1/p\}} HHD\left(G, \texttt{update}(\hat{G}, p \cdot i, \ell)\right)$
  $\bar{G} \leftarrow \texttt{update}(\hat{G}, p \cdot i^*, \ell)$
  if $HHD(G, \bar{G}) < HHD(G, \hat{G})$ **then** $\hat{G} \leftarrow \bar{G}$
  else break
return $\hat{G} = (\hat{V}, \hat{E})$

update$(\hat{G} = (\hat{V}, \hat{E}), q, \ell)$
  $\bar{G}(\bar{V}, \bar{E}) \leftarrow \hat{G}(\hat{V}, \hat{E})$
  remove $(q \cdot 100)\%$ of the hyperedges created at level $\ell - 1$
  create the same number of hyperedges at level ℓ
  return $\bar{G} = (\bar{V}, \bar{E})$

输入：原图G、分辨率p
输出：生成超图
初始化：使用HYPERCL产生随机超图
遍历从2到L，优化生成图到原图的相似度，一旦更加相似，选取对应的分组。

子程序：更新生成的超图
随机将q×100%的超边去掉，换成1级别（高一级，每组节点更少）的分组
返回新超图。

三、实验内容

1、实验设计

- 设计了对照试验，将**HYPERLAP+**与其他四种超图生成方法做了对比。分别是：**HYPERCL、HYPERPA、HYPERFF**和**无优化参数的HYPERLAP。**
- 测量实际超图和生成超图分布之间的相似性，使用Kolmogorov-Smirnov D-statistic参数：

$$D = \max_{x}\{|F(x) - F'(x)|\}$$

F和F‘是原图和生成图累积超边同质性分布

2、实验数据集

- 来自六个领域的真实超图：

Dataset	|V|	|E|	$\text{avg}_{e\in E}\|e\|$	$\max_{e\in E}\|e\|$
email-Enron	143	1,459	3.13	37
email-Eu	986	24,520	3.62	40
contact-primary	242	12,704	2.41	5
contact-high	327	7,818	2.32	5
NDC-classes	1,149	1,049	6.16	39
NDC-substances	3,767	6,631	9.70	187
tags-ubuntu	3,021	145,053	3.42	5
tags-math	1,627	169,259	3.49	5
threads-ubuntu	90,054	115,987	2.30	14
threads-math	153,806	535,323	2.61	21
coauth-DBLP	1,836,596	2,170,260	3.43	280
coauth-geology	1,091,979	909,325	3.87	284
coauth-history	503,868	252,706	3.01	925

3、实验结果

- 在相似性距离指标上，新方法明显优于旧方法。
- 在直观分布图上，**HYPERLAP+**分布曲线更接近原图。

三、实验内容

3、实验结果

- 在相似性距离指标上，新方法明显优于旧方法。

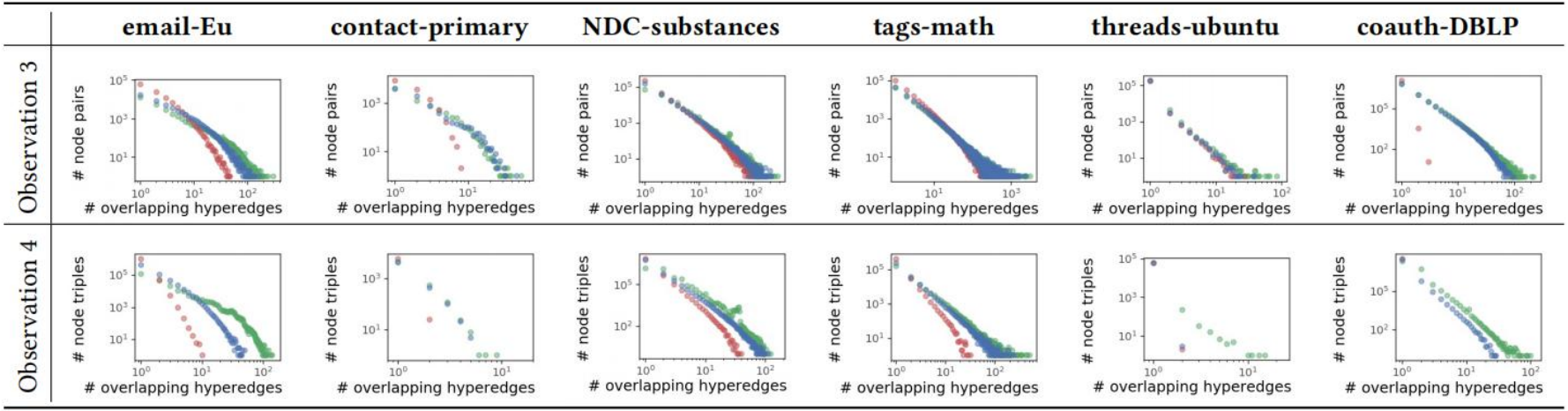
Dataset	Density of Egonets (Obs. 1)					Overlapness of Egonets (Obs. 2)					Homogeneity of Hyperedges (Obs. 5)				
	H-CL	H-PA	H-FF	H-LAP	**H-LAP$^+$**	H-CL	H-PA	H-FF	H-LAP	**H-LAP$^+$**	H-CL	H-PA	H-FF	H-LAP	**H-LAP$^+$**
email-Enron	0.545	0.202	0.391	0.405	**0.125**	0.517	0.398	0.398	0.391	**0.111**	0.498	0.241	0.656	0.191	**0.136**
email-Eu	0.724	-	0.402	0.577	**0.310**	0.534	-	0.639	0.432	**0.197**	0.505	-	0.688	0.247	**0.168**
contact-primary	0.896	0.537	0.975	0.334	**0.128**	0.867	0.471	0.942	0.285	**0.095**	0.430	0.236	0.484	**0.142**	0.188
contact-high	0.948	0.529	0.880	0.522	**0.345**	0.874	0.431	0.703	0.486	**0.296**	0.423	0.196	0.336	**0.120**	0.178
NDC-classes	0.694	0.785	0.731	0.696	**0.635**	0.302	0.715	0.406	**0.231**	0.248	0.274	0.410	0.484	0.272	**0.225**
NDC-substances	0.451	-	0.801	0.426	**0.366**	0.321	-	0.338	0.243	**0.157**	0.377	-	0.740	0.262	**0.108**
tags-ubuntu	0.522	**0.162**	0.216	0.410	0.300	0.432	**0.117**	0.398	0.487	0.210	0.245	0.136	0.844	0.105	**0.011**
tags-math	0.496	0.350	0.561	**0.195**	0.227	0.460	0.325	0.709	**0.151**	0.186	0.337	0.217	0.921	0.086	**0.015**
threads-ubuntu	0.159	0.856	-	0.163	**0.159**	0.299	0.953	-	0.300	**0.297**	0.020	0.291	-	0.016	**0.011**
threads-math	0.137	0.492	-	**0.120**	0.135	0.232	0.714	-	0.235	**0.229**	0.060	0.368	-	0.102	**0.019**
coauth-DBLP	0.228	-	-	0.227	**0.132**	0.302	-	-	0.267	**0.244**	0.715	-	-	0.540	**0.026**
coauth-geology	0.200	-	-	0.202	**0.138**	**0.248**	-	-	0.252	0.266	0.624	-	-	0.481	**0.044**
coauth-history	**0.087**	-	-	0.090	0.089	**0.316**	-	-	0.321	0.324	0.154	-	-	0.125	**0.020**
Average	0.468	0.489	0.619	0.335	**0.237**	0.439	0.515	0.566	0.313	**0.219**	0.358	0.261	0.644	0.206	**0.088**

-: out of time (taking more than 10 hours) or out of memory

三、实验内容

3、实验结果

■ 在直观分布图上，HYPERLAP+分布曲线更接近原图。



红色：HYPERCL　　蓝色：HYPERLAP+　　绿色：原图

四、主要贡献和可能的改进方向

1、本文创新点

- 真实超图中超边重叠的特点（来自六个领域的真实超图）：**实质性、重尾性、亲同性**。
- 新的测度指标：定义了**重叠度**和**同质度**两个评估超边重叠特征的指标。
- 新的超图生成模型：**HYPERLAP、HYPERLAP+**。前者可以生成指定超边分布的超图，后者可以自动确定生成超图时的参数。

2、可改进点

- 使用原始图分布的先验信息指导节点层的选取。
- 定义一些其他评估超边重叠性的指标，考察方法在这些指标上的表现。
- 探索更好的优化权值方法，以获得更优的结果。
- 能否添加一些条件，简化优化过程，缩短训练时间？

谢谢!