网络中的重叠社区检测：现有技术和比较研究

论文标题：Overlapping Community Detection in Networks:
the State of the Art and Comparative Study

论文作者：Jierui Xie, Stephen Kelley, Boleslaw K. Szymanski

本文主要的贡献：

1）介绍了重叠社区检测的背景知识；

2）回顾了多种重叠社区检测算法并对其进行了分类解释；

3）给出了两种社区检测质量度量的方法；

4）提出了社区检测基准的最新技术；

5）通过合成网络和真实的社交网络对上述的算法进行测试；

6）最后综合考虑各种算法在社区级和节点级的检测性能得出结论并提出现有算法中存在的问题。

背景

简介：

社区（模块化结构）是现实世界社交网络的重要属性。现有的社区检测的技术有：随机游走、频谱聚类、模块最大化、微分方程和统计力学等。社区检测中的重点是确定不相交的社区。但是，由于社交网络中的一个人可能有多个社区成员资格，所以，本文提供了关于识别一个节点属于多个集群的重叠社区检测算法的最新技术回顾。

符号表示：

Symbol	Description
G	a network or graph
V	a set of nodes
E	a set of edges
n	the number of nodes
m	The number of edges

A	the adjacency matrix for unweighted network
W	the adjacency matrix for weighted network
C	the set of clusters called a cover
a_{ic}	belonging factor:a measure of the strength of association between node i and cluster c.注意:$0 \le a_{ic} \le 1 \ \forall i \in V, \forall c \in C$ and $\sum_{c=1}^{|C|} a_{ic} = 1$
|C|	the number of clusters

算法输出结果的类型：

（1）fuzzy assignment(模糊赋值):节点成比例地与社区相关联

使用 threshold（阈值）划分为两部分

（2）crisp assignment(非模糊赋值):节点和社区之间的关系是二进制的

杰卡德指数(Jaccard Index):

两个集合 A 和 B 的交集元素在 A，B 的并集中所占的比例，称为两个集合的杰卡德相似系数，用符号 J(A,B)表示。

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

杰卡德相似系数是衡量两个集合的相似度一种指标。

RI(兰德指数)　ARI(调整兰德指数)

$$RI = \frac{a+b}{C_2^{n_{samples}}}$$

其中 C 表示实际类别信息，K 是聚类结果，a 表示在 C 与 K 中都是同类别的元素对数，b 表示在 C 与 K 中都是不同类别的元素对数

其中 $C_2^{n_{samples}}$ 表示数据集中可以组成的对数，RI 取值范围为[0,1]，值越大意味着聚类结果与真实情况越吻合。

RI 越大表示聚类效果准确性越高 同时每个类内的纯度越高

为了实现“在聚类结果随机产生的情况下，指标应该接近零”，调整兰德系数（Adjusted rand index）被提出，它具有更高的区分度：

$$\mathrm{ARI} = \frac{\mathrm{RI} - E[\mathrm{RI}]}{\max(\mathrm{RI}) - E[\mathrm{RI}]}$$

ARI 取值范围为[−1,1]，值越大意味着聚类结果与真实情况越吻合。从广义的角度来讲，ARI 衡量的是两个数据分布的吻合程度。

TP、TN、FP、FN：

True Positive（真正,TP）被模型预测为正的正样本；
True Negative（真负,TN）被模型预测为负的负样本 ；
False Positive（假正,FP）被模型预测为正的负样本；
False Negative（假负,FN）被模型预测为负的正样本。

算法

本文提供了多种不同算法的比较，根据如何识别社区分为了五类，在后面进行合成网络和真实的社交网络测试的算法有十四个。先对这十四个进行整体分类后面再详细介绍每一种。

- 派系过滤方法（Clique Percolation）CFinder
- 线图和链路分区（Line Graph and Link Partitioning）Link
- 局部扩展和优化（Local Expansion and Optimization）CIS、LFM、OSLOM、UEOC、iLCD、EAGLE、GCE
- 模糊检测（Fuzzy Detection）MOSES、NMF
- 基于代理和动态（Agent-Based and Dynamical）COPRA、SLPA、Game

一、派系过滤方法（Clique Percolation）

派系过滤方法 CPM 基于以下假设：社区由完全连接的子图的重叠集组成，并通过搜索相邻团来检测社区。

描述：定义 k-派系结构是网络中包含 k 个节点的完全子图，所有彼此连通的 k-派系的集合构成一个 k-派系社区。CPM 算法通过对网络中各节点的度可以判断可能存在的最大派系，从一个节点出发找到包含该节点的大小为 k 的派系，采

用迭代回归的算法寻找网络中大小不同的派系，最后将彼此连通的 k-派系集合为一个社区结构。

步骤：

Step1：识别网络中大小为 k 的所有团。

Step2：构造一个新的图，使得每个顶点表示为这些 k-派系中的一个。如果表示它们的 k-派系共享 k-1 个成员，则连接两个节点。

下面是一个 4-派系社区的图例：

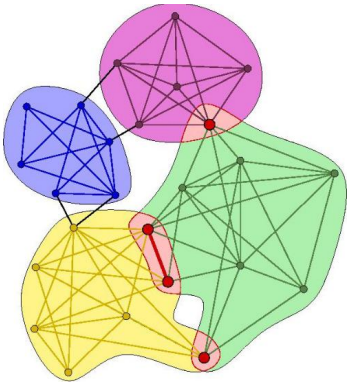


Fig.1. Illustration of the four k-clique communities at k = 4

注意：由于顶点可以同时处于多个 k-派系，所以社区之间可能存在重叠。

评价：CPM 适用于比较密集的网络，而且 k 值较小时结果较好。CFinder 是 CPM 的实现。由于即使小的网络可以包含大量的 k-派系，这种方法的实现是基于定位最大派系而不是个体 k-派系。因此，这种方法的复杂性在实践中有大的时间复杂性（尽管发现 k-派系是多项式），所以不能在大型社交网络中使用。

扩展：

加权派系过滤算法 CPMw[Farkas et al.2007]把子图强度有效地结合到派系搜索中，根据设定的固定强度阈值，将大于此阈值的 k-派系包含到社区中。与 CPM 算法不同的是，CPMw 算法从整体考虑派系强度，因而允许 k-派系中包括低于预设阈值的边，从而得到更稳定的社区划分。注意：子图中链路权重的几何平均值称为该子图的强度。

快速派系过滤算法 SCP[Kumpula et al.2008]该算法根据边的权重递减的顺序将边添加到网络，并记录新出现的 k-派系。再根据得到的 k-派系结构，检测 k-派系之间存在的重叠程度，以判断 k-派系是否已包含在之前 k-派系社区之中，逐步更新以确定最终的社区划分。由于每个 k-派系被处理两次，SCP 算法的计

算复杂度与网络中 k-派系数量成线性关系，其运行时间相比于 CPM 明显降低。该算法能有效揭示网络中社区的层次结构，另外在处理加权网络时，CPM 算法需迭代计算每个可能的权重阈值下的社区，而 SCP 算法只需要运行一次就能获得多阈值下的社区结构同时嵌套的社区结构。在处理大规模加权社交网络，当缺乏有关合适阈值的先验知识或研究多阈值下社区结构时，SCP 算法就显得非常有效。

二、线图和链路分区（Line Graph and Link Partitioning）

链路分割算法是分割链接而不是节点来发现社区结构，如果连接到一个节点的链接被放在多个簇中，则这些在原始图中的节点被称为是重叠的。

描述：通过边的相似性聚类划分链接。

步骤：

Step1：给定在节点 k 上的一对链路 e_{ik} 和 e_{jk}，通过 Jaccard 指数来计算相似性 $S(e_{ik},e_{jk})=\frac{|N_i\cap N_j|}{|N_i\cup N_j|}$ 其中 N_i 是包括节点 i 的邻域。

Step2：使用单链层次聚类来构造以 k 为根的链接树形图。

Step3：在某个阈值下，切割这个树形图来产生链接社区。

评价：利用链路分割的重叠社区检测不能保证比基于节点的检测有更高的质量，因为这些算法依赖于社区的模糊定义，比如说上述的各种阈值，并不能说明哪个才是最适合的。

扩展：

Evans[Evans and Lambiotte 2009;2010]将网络投影到加权线图中，原来图中的链接成为现在图中的节点，然后就可以应用不相交的社区检测算法。线图的节点分区对应于原始图边的分区。

CDAEO[Wu et al.2010]提供了用于确定重叠程度的后处理过程。当线图初步分割完成，对于每个节点 i 具有$|E_{icmin}|$或$|E_{icmax}|$的预定阈值，其中 $E_{icmin(cmax)}$是社区中具有最小（最大）连接数的边的集合，$|E_{icmin}|$或$|E_{icmax}|$表示边的数目，将线图中识别的 E_{icmin} 集合的链接去除，这实质上将节点 i 减少到成为单个成员，这样能更好的确定社区的重叠程度。

基于链接的扩展模块化[Nicosia et al. 2009]重叠社区的模块度建立在链接的归属系数上。在社区 c 中，节点 i 到节点 j 的一条链接 l(i,j)表示为$\beta_{l(i,j),c}= F(a_{ic},a_{jc})$，则从 i 到 j 的任何可能的链接 l(i,j)的期望归属系数可以被定义为 $\beta^{out}_{l(i,j),c} = \frac{1}{|V|}\sum_{j\in V} F(a_{ic},a_{jc})$（其实就是求均值），在社区中指向节点 j 的任何链路 l(i,j)的期望归属系数定义为$\beta^{in}_{l(i,j),c} = \frac{1}{|V|}\sum_{i\in V} F(a_{ic},a_{jc})$（它与上一个公式的不同之处在于“链接所指的方向”）。上述归属系数被用作观察链路 A_{ij} 和在空模型中从 i 到 j 开始的链路的概率$\frac{k_i^{out}k_j^{in}}{m}$的权重，因此得到新的模块化定义为$Q_{ov}^{Ni} = \frac{1}{m}\sum_c\sum_{i,j\in V}\left[\beta_{l(i,j),c}A_{i,j} - \beta^{out}_{l(i,j),c}\beta^{in}_{l(i,j),c}\frac{k_i^{out}k_j^{in}}{m}\right]$其中$k_i^{out(in)}$是节点 i 的输出（输入）链路的数量，m 是边的总数。Q_{ov}^{Ni}取决于链路属性系数 $F(a_{ic},a_{jc})$，其可以是 a_{ic} 和 a_{jc} 的乘积、平均值或最大值。

三、局部扩展和最优化（Local Expansion and Optimization）

利用局部扩展和优化的算法是基于增长的社区或者部分社区，它们中的大多数依赖于密集连接节点组质量的局部效益函数。

描述：Baumes [Baumes et al.2005]提出了“两步走”过程。

步骤：

Step1：初步形成“种子”

根据一些标准用 RankRemoval 算法对节点进行排序。然后，迭代地移除高度排序的节点，直到形成小的不相交的集群核。这些集群核作为第二步的“种子”。

Step2：迭代扫描（IS）

就是通过添加或删除节点来扩展由第一步形成的“种子”，每扩展一次就计算局部密度函数，直到局部密度函数的值不再改变为止。密度函数可以表示为 $f(c) = \frac{w_{in}^c}{w_{in}^c + w_{out}^c}$，其中$w_{in}^c$和$w_{out}^c$是社区 c 总的内部和外部的权重。社区检测的质量取决于种子的质量。

评价：由于该算法允许在扩展期间去除顶点，所以迭代扫描（IS）可能会产生部分断开的情况。

扩展：

CIS[Kelley2009]“两步走”的修改版本，在每次迭代之后检查连接性。当产生断开时，只保留密度最大的那个。CIS 还开发了包含边缘概率 e_p 的新的适应度函数$f(c)=\frac{w_{in}^c}{w_{in}^c+w_{out}^c}+\lambda e_p$。参数λ控制算法在网络的稀疏区域中的行为，节点的添加需要在内部密度的变化和边缘密度的变化之间取得平衡。

随机种子节点的局部优化方法 LFM [Lancichinetti et al. 2009]定义了一种节点适应度函数$f(c)=\frac{k_{in}^c}{(k_{in}^c+k_{out}^c)^\alpha}$来度量社区内部边与社区外部边的关系。基于节点适应度的贪婪优化，从随机种子节点扩展直到适应度$f(c)=\frac{k_{in}^c}{(k_{in}^c+k_{out}^c)^\alpha}$局部最大来形成社区，其中$k_{in}^c$和$k_{out}^c$分别是社区 c 内部和外部的总程度，α是控制社区大小的分辨率参数，通过对不同分辨率参数进行调节可得到社区的层次结构。在找到一个社区之后，LFM 随机选择尚未分配给任何社区的另一个节点来生长新社区。注意：由于 LFM 算法的种子节点的选取是随机的，可能导致不稳定的结果。

MONC[Havemann et al. 2011]修正 LFM 的适应度函数为$f(c)=\frac{k_{in}^c+1}{(k_{in}^c+k_{out}^c)^\alpha}$表示允许单个节点本身被认为是社区。这就避免了违反局部性原则，使得 MONC 能够找到其中一组节点是局部最优的α（分辨率参数）的范围。MONC 计算α的下一个最低值来进一步扩展社区。当一个节点 i 形成的社区是另一节点形成社区的子集时，i 的分析停止（因为如果继续扩展，必然还是形成原来已知的那个社区，现在停止就可以减少以后的计算量）。因为 MONC 在处理期间合并社区，所以与 LFM 相比能更快地揭示网络。

OSLOM[Lancichinetti et al. 2011]关于全局空模型（由配置模型生成的随机图）在社区扩展期间，测试了一个簇的统计数据。为了扩展当前社区，针对每个邻居计算 r 值(在空模型中，内部连接数≧从邻居到该社区的连接数的累积概率)，如果最小 r 值的累积概率小于给定容差，则认为是有意义的，将相应的节点添加到社区。否则，检查第二小的 r，以此类推。OSLOM 通常导致大量的异常值或单例社区。

UEOC [Jin et al. 2011]基于相应退火算法，采用随机游走的 l 步转移概率扩展节点形成社区。按转移概率降序排序节点之后，以某种适当的截止值提取社区。模拟退火算法以一定的概率来接受一个比当前解要差的解，因此有可能会跳出这个局部的最优解，达到全局的最优解。

OCA [Padrol-Sureda et al. 2010]基于将每个节点映射到 d 维向量的思想。节点 S 的每个子集被定义为该集合中的各个向量的和。适应度函数被定义为函数 O 上的有向拉普拉斯算子，其中 O 是子集向量的平方欧几里德长度。像 LFM 一样，从一些初始种子开始，OCA 试图去除或添加导致适应度函数的值最大增加的节点。

iLCD[Cazabet et al. 2010]能够检测静态和暂时的社区。给定创建的一组边，如果其第二邻居的数目和鲁棒第二邻居的数目大于期望值，则 iLCD 通过添加新节点来更新现有社区。如果检测到的是最小模式，则还允许由新边创建新社区。将两个社区之间的相似性定义为共同节点的比率，如果相似性高，则执行“合并”来提高检测效率。

EAGLE[Shen et al. 2009]使用聚集框架来产生树形图。

Step1：发现所有最大派系并且使其成为初始社区，派系对于许多局部优化算法是非常重要的。

Step2：合并具有最大相似度的一对社区。

GCE[Lee et al. 2010]第一步与 EAGLE 相同。

Step1：将最大派系识别为初始社区。

Step2：通过贪心算法优化局部适应函数来扩展社区，$1-\frac{|c_1 \cap c_2|}{min(|c_1|,|c_2|)}$ 定义为的社区 c_1 和 c_2 之间的距离，如果 c_1 与 c_2 类似(也就是说 $1-\frac{|c_1 \cap c_2|}{min(|c_1|,|c_2|)}$ 小于给定参数ϱ)，则去除与先前发现类似的社区。

四、模糊检测（Fuzzy Detection）

模糊检测算法量化所有节点和社区对之间的关联强度。在这些算法中，为每个节点计算成员隶属向量或归因因子。缺点是需要确定向量的维数 k。

Nepusz [Nepusz et al. 2008]将重叠社区检测建模为非线性约束优化问题，可以通过模拟退火的方法来解决。目标是要最小化函数 $f=\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}(\tilde{s_{ij}}-s_{ij})^2$ 的值，w_{ij} 是预定义的权重，$\tilde{s_{ij}}$ 是节点 i 和 j 之间的已知的相似度，相似度 s_{ij} 定义为 $s_{ij}=\sum_{c} a_{ic}a_{jc}$，$a_{ic}$ 是社区 c 与节点 i 的模糊隶属度，满足 $0\le a_{ic}\le 1, \forall i\in V, \forall c\in C$

且 $\sum_{c=1}^{|C|} a_{ic} = 1$（f 表示的是已知的相似度与实际测得的相似度的差距）。节点 i 的归属因子被定义为 $Q_{ov}^{Nc} = \frac{1}{2m}\sum_{c}\sum_{i,j\in c}\left[A_{ij} - \frac{k_i k_j}{2m}\right] a_{ic}a_{jc}$，$a_{ic}$ 是节点 i 在社区 c 中的隶属程度，增加 k 的值，直到社区结构没有改善，可以确定社区 k 的值。

SSDE[Magdon-ismail and Purnell 2011]类似的混合模型被构造为节点的生成模型。

Step1：首先使用频谱聚类方法将网络映射到 d 维空间。

Step2：通过预期最大化算法训练高斯混合模型(GMM)。

Step3：当添加簇的对数似然性的增加不明显高于将簇添加到在同一空间上均匀的随机数据时，确定社区的数量。

非负矩阵因子分解（NMF）是经常在社区检测的机器学习中进行特征提取和维数降低的技术。NMF 将特征矩阵 V 近似因子分解为具有非负性约束的两个矩阵 V ≈WH，其中 V 是 n×m，W 是 n×k，H 是 k×m，k 是用户提供的社区数。W 表示缩减特征空间中的数据，节点 i 相对于社区 j 的依赖性量化为归一化的 W 中的每个元素 $w_{i,j}$ 值。[Zhang et al. 2007b]把特征矩阵 V 替换为扩散核，扩散核是网络中拉普拉斯算子的函数。[Zarei et al. 2009]特征矩阵 V 被定义为拉普拉斯的列的相关矩阵，使得比[Zhang et al. 2007b]有较好的性能。[Zhao et al. 2010]去除了近似的冗余约束，将 NMF 简化为对称非负矩阵分解(s-NMF)的问题。

Wang et al[Wang et al. 2009]使用局部优化算法与不相交检测方法的组合。

Step1：利用任意的不相交团体检测算法获得初始社区。

Step2：社区尝试添加或删除节点。

Step3：计算包括节点 i 或去除节点 i 的社区上的两个适合度分数的方差。

Step4：归一化方差形成节点 i 的模糊隶属向量。

Ding [Ding et al. 2010]采用相似性传播聚类算法进行重叠检测，其中簇由代表性样本识别。

Step1：通过通勤时间内核（逆拉普拉斯算子的函数）将节点映射为欧几里德空间中的数据点。

Step2：通过余弦距离来测量节点之间的相似性，由此相似性来确定社区结构。

五、基于代理和动态算法（Agent-Based and Dynamical Algorithms）

标签传播算法[Raghavan et al. 2007; Xie and Szymanski 2011]其中具有相同标签的节点形成社区，通过允许节点具有多个标签可以扩展到重叠的社区检测上。

COPRA[Gregory 2010]每个节点通过以同步方式，在每个时间步长，对来自其所有邻居的系数进行求平均值来更新其归属系数。

SLPA[Xie et al. 2011; Xie and Szymanski 2012]是基于通用讲话者的信息传播过程。它根据成对的交互规则在节点之间扩展标签，节点忘记在先前迭代中获得的信息，SLPA 向每个节点提供存储器以存储接收到的信息（标签）。在节点的存储器中观察标签的概率称为隶属强度。SLPA 不需要先前知道社区的数量，这是由网络中标签的聚类确定的。SLPA 还可以通过泛化交互规则（称为 SLPAw）来应用于加权和定向网络。

游戏理论框架[Chen et al. 2010]社区与纳什局部均衡有关。增益函数和损失函数与每个代理相关联。Game 假定每个代理都是自私的，它们根据自己的效用选择加入、离开和切换社区。只要能增加效用，就允许代理联合多个社区来处理重叠。

评价标准（质量度量）

不相交的社区检测评价标准已经提出过很多，只有少数措施适合于重叠的社区。最常用两种方法是：标准互信息(NMI)和 Omega 指数。

一、标准互信息（Normalized Mutual Information）

对于 cover C'中的每个节点 i，其社区成员可以表示为长度|C'|的二进制向量(C'中簇的数目)。如果节点 i 属于第 k 个簇 C'_k，$(X_i)_k=1$，否则$(X_i)_k = 0$。该向量的第 k 个条目可以看作随机变量 X_k，其概率分布为 $P(X_k=1)=n_k/n$,$P(X_k=0)=1-P(X_k=1)$ ($n_k=|C'_k|$表示群集 C'_k 中的节点的数量，n 是节点的总数)。同样 cover C''中的第 l 个簇的随机变量 Y_l 也满足上述概率分布。使用经验边缘概率分布 $P(X_k)$和联合概率分布 $P(X_k,Y_l)$进一步定义熵 H(X)和 $H(X_k,Y_l)$。给定 Y_l，群 X_k 的条件熵定义为 $H(X_k|Y_l)= H(X_k,Y_l)-H(Y_l)$。相对于整个向量 Y，$X_k$ 的熵是基于 X_k 与由 $H(X_k|Y)=\min_{l\in\{1,2,\ldots,|C''|\}}H(X_k|Y_l)$给出的 Y 的任意分量之间的最佳匹配（最小值）。cover X 相对于 Y 的标准化条件熵为 $H(X|Y)=\frac{1}{|C'|}\sum_k \frac{H(X_k|Y)}{H(X_k)}$，以相同的方式可以定义

H(Y|X)。最后由$NMI(X|Y) = 1 - [H(X|Y) + H(Y|X)]/2.$得出 cover C'和 C"的 NMI。NMI 在 0 和 1 之间，其中 1 对应于完全匹配。（可以看出，匹配得越好，H(Y|X)越小，NMI 越大）

二、Omega 指数（Omega Index）

Omega 指数是调整兰德系数(ARI)的重叠版本。它是基于在两个 cover 中的节点对，如果一对节点聚集在完全相同数量的社区（可能没有）中，则认为它们是一致的。也就是说，omega 指数考虑：有多少个节点对没有在集群中共同存在、有多少个节点恰好被放置在一个集群中、有多少个恰好被放置在两个集群中等等。令 K_1 和 K_2 分别是 coverC_1 和 C_2 中的社区数，Omega 指数定义为 $\omega(C_1, C_2) = \frac{\omega_u(C_1, C_2) - \omega_e(C_1, C_2)}{1 - \omega_e(C_1, C_2)}$。Omega 指数越大，两个 cover 之间的匹配就越好，Omega 指数等于 1 表示完全匹配，当没有重叠时，Omega 指数就是 ARI。

基准

GN 基准：给定预期程度，等大小的社区被嵌入到网络中，给定混合参数μ，测量内部连接与输出连接的比率。

缺点： 1）不能解决复杂网络中的异质性；

2）不允许嵌入的社区重叠。

LFR 基准：引入网络的异质性和社区大小分布。更接近在现实世界网络中观察到的特征。但是，要求重叠节点与相同数量的嵌入社区交互，这是是不现实的。

[Gregory 2011]提出的改进是通过对节点的出现添加属性系数。该系数定义为 $p_{ij} = s_{ij}p_1 + (1 - s_{ij})p_0$其中 p_k 是共同出现 k 次的节点 i 和 j 的连接概率，$s_{ij} = \sum_{c \in C} \alpha_{ic}\alpha_{jc}$ 是节点 i 和节点 j 的相似度，a_{ic} 是社区 c 中节点 i 的模糊隶属度。换句话说，边缘存在的概率不仅取决于节点 i 和 j 在一起出现的社区的数量，而且取决于它们对这些社区的隶属程度。

合成网络测试

在本节中，收集和测试比较了十四种算法在 LFR 网络上的性能，如表 I 所示。

Table I. Algorithms included in the experiments.

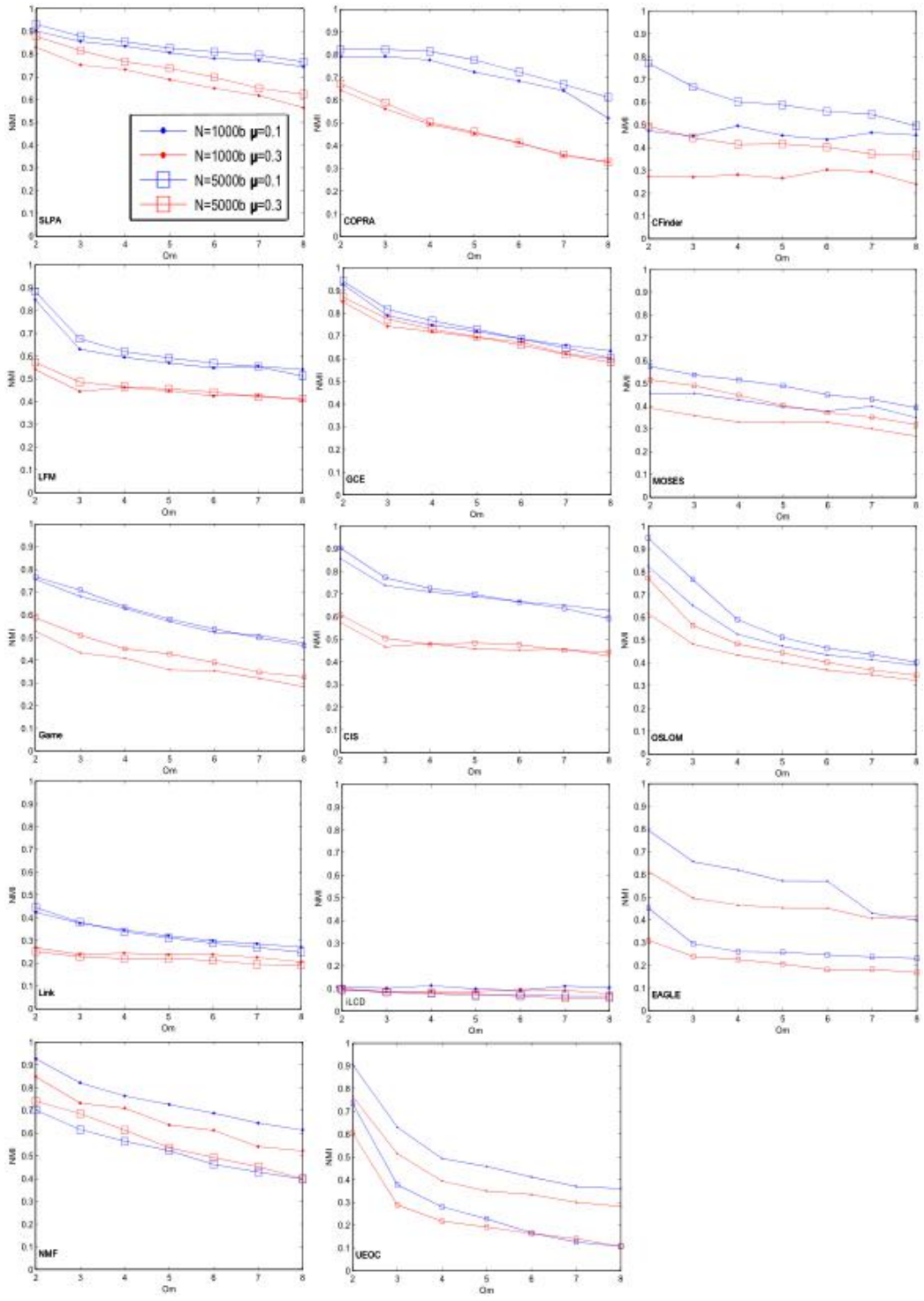
Algorithm	Reference	Complexity	Imp
CFinder	[Palla et al. 2005]	-	C++
LFM	[Lancichinetti et al. 2009]	$O(n^2)$	C++
EAGLE	[Shen et al. 2009]	$O(n^2 + (h+n)s)$	C++
CIS	[Kelley 2009]	$O(n^2)$	C++
GCE	[Lee et al. 2010]	$O(mh)$	C++
COPRA	[Gregory 2010]	$O(vm \log(vm/n))$	Java
Game	[Chen et al. 2010]	$O(m^2)$	C++
NMF	[Psorakis et al. 2011]	$O(kn^2)$	Matlab
MOSES	[McDaid and Hurley 2010]	$O(en^2)$	C++
Link	[Ahn et al. 2010]	$O(nk_{max}^2)$	C++
iLCD	[Cazabet et al. 2010]	$O(nk^2)$	Java
UEOC	[Jin et al. 2011]	$O(ln^2)$	Matlab
OSLOM	[Lancichinetti et al. 2011]	$O(n^2)$	C++
SLPA	[Xie et al. 2011; Xie and Szymanski 2012]	$O(tm)$	C++

注意：给出的时间复杂度是最坏时间复杂度。

重叠度由两个参数确定：O_n(重叠节点的数量)，O_m(每个重叠节点所属的社区的数量)。O_n 设置为节点总数的 10%和 50%，分别表示低和高重叠密度，也允许从 2~8 变化，表示重叠节点的重叠多样性。两个先前讨论的方法：Omega 指数和 NMI，用于量化由算法发现的 cover 的质量。

一、μ、n 和 O_m 的影响

首先研究了由 NMI 测量的性能随着图 1 中不同网络大小(n)和体内强度(μ)的成员数 O_m 从小值到大值（即 2 到 8）的变化而变化。



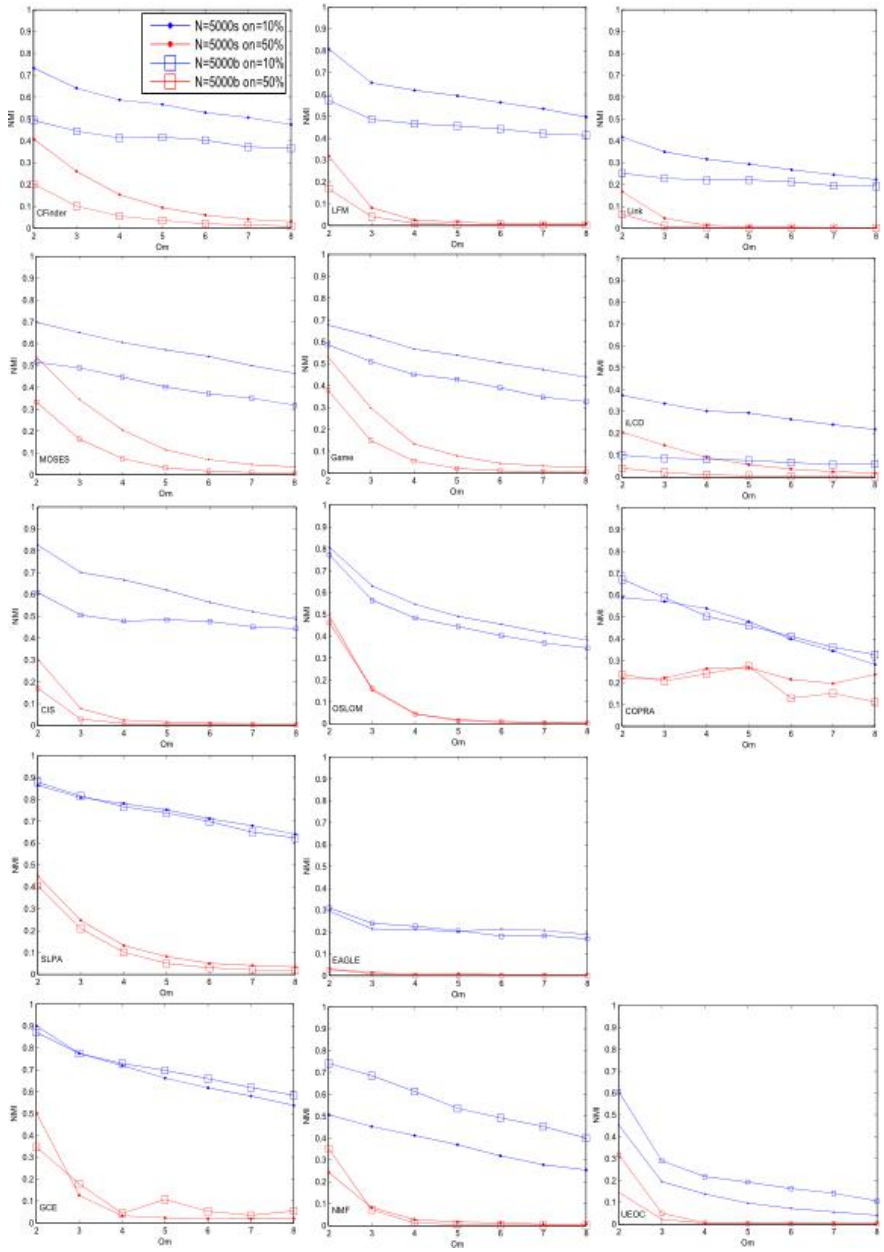
图一

一般来说，网络拓扑的变化，特别是混合值μ，对不相交的社区检测具有相似的影响。一方面，μ的值越大，检测算法产生的结果越差（图 1 中的红色曲线<蓝色曲线）。这对于大多数算法都是如此，唯一的例外是 NMF；另一方面，将网络大小从 1000 增加到 5000 通常有略微更好的性能（图 1 中的方形>点），除

了 EAGLE、NMF 和 UEOC 之外。对于 iLCD 和 Link，观察到性能略有波动。 随着重叠的多样性增加（即 O_m 变大），除了 OSLOM 和 UEOC 之外，检测性能通常以中等速率衰减。

二、社区大小的范围和重叠密度的影响

我们评估了 O_n 和社区大小的范围对 n = 5000 和μ= 0.3 的网络上的每个单独算法的影响，NMI 的结果显示在图 2 中。



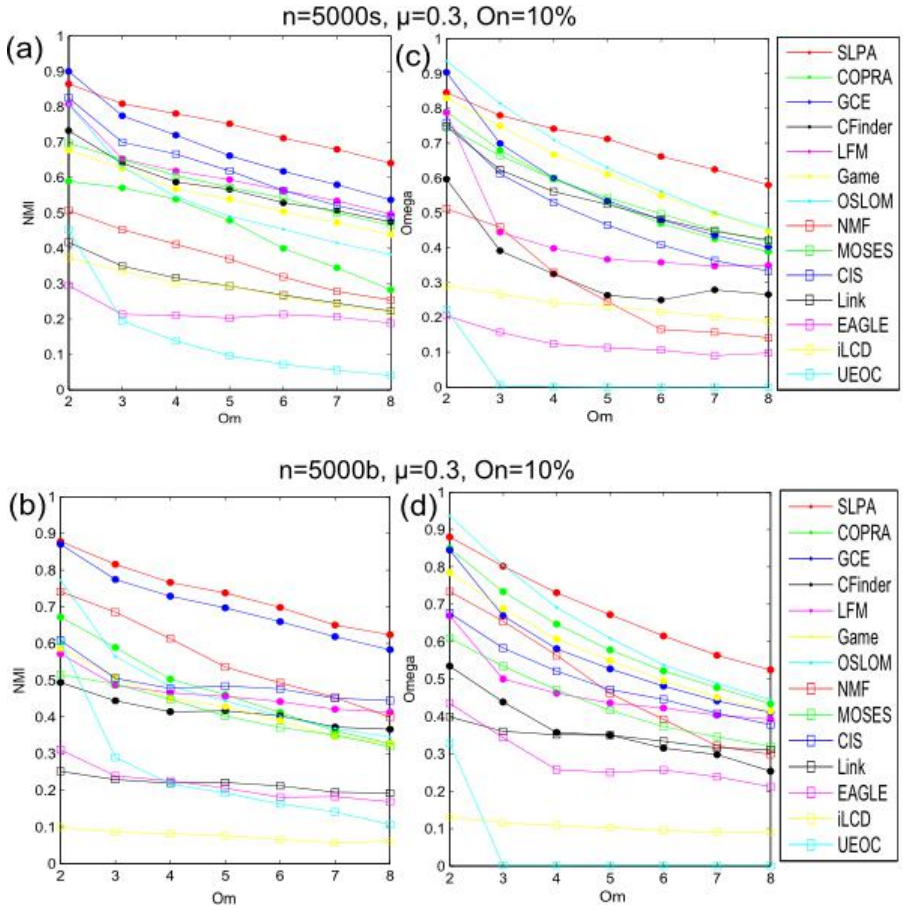
图二

如所预期的，对于所有算法，高重叠密度（即红色曲线 O_n=50％）情况下，

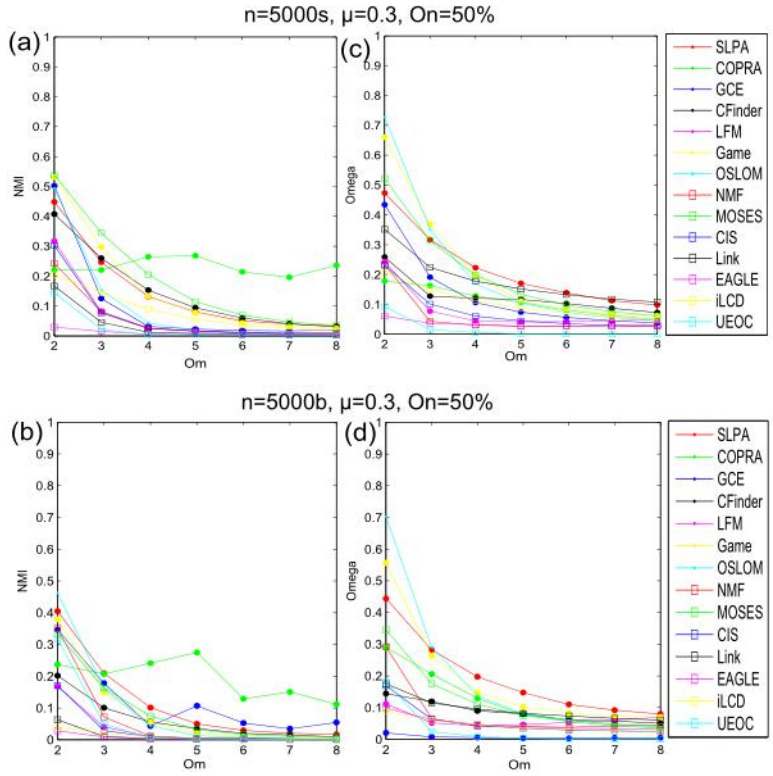
检测的性能一致且显著下降。但是，在低重叠密度的情况下，小社区的大小范围和大社区的大小范围之间的性能差异（具有相同颜色的两条曲线之间的间隙）更大。而且，对于具有小社区大小 s=(10,50)的网络的 NMI 通常高于具有大社区大小范围 b=(20,100)的网络（图 2 中的点>正方形）的网络。由此可见，众所周知的分辨率限制在这里不起作用，因为所有测试的算法既不基于模块化也不基于扩展模块化。由 CFinder、LFM、Link、MOSES、Game、iLCD、CIS 和 OSLOM 算法可以证明，其在小和大社区大小的范围之间具有显著的性能差距，在两个测试范围内只有很小的性能差异，我们还得出结论，社区大小范围对包括 SLPA、COPRA、EAGLE 和 GCE 在内的算法的影响有限。

三、社区检测的排名

对不同的重叠密度和社区大小范围进行了比较。由 NMI 和 Omega 指数针对 n=5000 和μ=0.3 测量的性能在图 3 和图 4 中表示。



图三



图四

为了更直观的比较两个测量总结的大量实验结果，提出 $RS_M(i)$，给定算法 i 相对于某个测量 M 的平均排名得分为 $RS_M(i) = \sum_{j=1} rank(i, O_m^j)$ 函数 rank 返回给定 O_m 的算法 i 的排名。以递增的顺序对 RS_M 进行排序，给出算法中的最终排名。排名被总结为四个等级 RS^s_{NMI}, RS^b_{NMI}, RS^s_{Omega} and RS^b_{Omega}，$RS^{s(b)}_{NMI(Omega)}$ 表示基于具有小（或大）社区大小范围网络的 NMI（或 Omega）的排名。

表 II 中总结的是图 3 中的低重叠密度情况的结果。

Table II. The community detection ranking for $n = 5000$, $\mu = 0.3$ and low overlapping density $O_n = 10\%$.

Rank	RS^s_{NMI}	RS^s_{Omega}	RS^b_{NMI}	RS^b_{Omega}	$RS^s_{NMI,Omega}$	RS^s_F
1	SLPA	SLPA	SLPA	SLPA	SLPA	SLPA
2	GCE	OSLOM	GCE	OSLOM	GCE	CFinder
3	CIS	Game	NMF	COPRA	OSLOM	Game
4	LFM	GCE	CIS	Game	CIS	OSLOM
5	MOSES	MOSES	COPRA	GCE	Game	MOSES
6	CFinder	COPRA	OSLOM	CIS	COPRA	COPRA
7	Game	Link	LFM	NMF	LFM	iLCD
8	OSLOM	CIS	Game	LFM	MOSES	Link
9	COPRA	LFM	CFinder	MOSES	NMF	LFM
10	NMF	CFinder	MOSES	CFinder	CFinder	UEOC
11	Link	NMF	Link	Link	Link	EAGLE
12	iLCD	iLCD	EAGLE	EAGLE	EAGLE	GCE
13	EAGLE	EAGLE	UEOC	iLCD	iLCD	CIS
14	UEOC	UEOC	iLCD	UEOC	UEOC	NMF

表 III 中总结的是图 4 中的高重叠密度情况的结果。

Table III. The community detection ranking for $n = 5000$, $\mu = 0.3$ and high overlapping density $O_n = 50\%$.

Rank	RS^s_{NMI}	RS^s_{Omega}	RS^b_{NMI}	RS^b_{Omega}	$RS^*_{NMI,Omega}$	RS^*_F
1	MOSES	SLPA	COPRA	SLPA	SLPA	Link
2	COPRA	Link	SLPA	Game	MOSES	UEOC
3	CFinder	Game	GCE	OSLOM	Game	SLPA
4	Game	MOSES	MOSES	Link	COPRA	Game
5	SLPA	CFinder	CFinder	MOSES	CFinder	LFM
6	GCE	OSLOM	Game	CFinder	OSLOM	CFinder
7	iLCD	COPRA	OSLOM	COPRA	Link	CIS
8	OSLOM	GCE	LFM	LFM	GCE	MOSES
9	CIS	iLCD	CIS	NMF	LFM	OSLOM
10	LFM	LFM	NMF	EAGLE	CIS	iLCD
11	NMF	CIS	UEOC	CIS	iLCD	GCE
12	Link	NMF	iLCD	iLCD	NMF	COPRA
13	EAGLE	EAGLE	Link	UEOC	EAGLE	EAGLE
14	UEOC	UEOC	EAGLE	GCE	UEOC	NMF

我们首先比较成对相似性(RS^s_{NM},RS^s_{Omega})和(RS^b_{NM},RS^b_{Omega})。如表所示，对于(RS^s_{NM},RS^s_{Omega})，在前七个算法中，这两个排名，有 4 对匹配 O_n=10%，5 对匹配为 O_n=50%；对于(RS^b_{NM},RS^b_{Omega})则更多，O_n=10%和 O_n=50%都有 6 对匹配。这表明 NMI 和 Omega 指数在一定程度上提供类似的总体评价。

基于这四个排名，我们进一步导出平均排名 $RS^*_{NMI,Omega}$ 作为整体社区检测的性能，分别列在表 II 和表 III 的倒数第二列。在这两个最终排名中，对于低重叠密度网络，前七个算法是基于代理的算法（SLPA、Game 和 COPRA）和基于本地扩展的算法（GCE、OSLOM、CIS 和 LFM），参见图 3。对于高重叠密度网络，基于代理的算法保持在前七个，连同表示模糊算法的 MOSES，表示派系过滤算法的 CFinder，表示链接分割的 Link，参见图 4。然而，对比图 3 和图 4，低重叠密度网络显示出优异的性能，而高重叠密度网络的性能实际上相当低的事实，可以得出结论：对于具有高重叠密度和高重叠分集（例如，对于 O_n=50%和 O_m>2）的网络，所有算法尚未实现令人满意的性能。

四、比较 LFR 中检测到的社区规模分布

为了验证前面得到的排名，给出了社区规模（CS）的分布（用直方图表示），并与已知的地面实际情况进行比较。这里只提供 n=5000，μ=0.3，O_n=10%的分析（相应的排名是表 II 中的 RS^b_{NMI}）。

如图 5 所示，在地面真实分布情况下，SLPA、GCE 和 NMF 的社区大小分布是单峰分布的，而且单一的峰值是 CS = 20。这解释了为什么它们在排名 RS^b_{NMI} 方面表现良好。LFM 和 MOSES 在 CS = 5 附近具有峰值，性能较低。

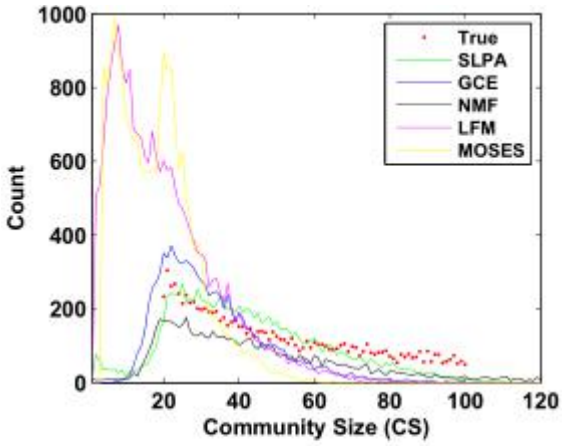


Fig. 5. Histogram of the detected community sizes for SLPA, GCE, NMF, LFM and MOSES created from the results for LFR networks with $n = 5000$, $\mu = 0.3$ and $O_n = 10\%$.

如图 6 所示，OSLOM、Game、COPRA、CIS 和 CFinder 算法的特征是双峰分布，在 CS=1~5 处具有峰值，这意味着算法找到大量的小社区。

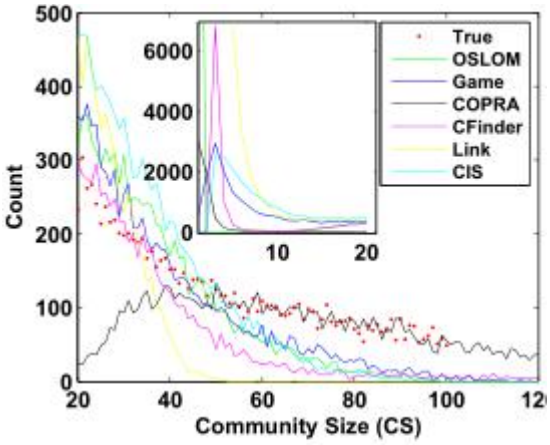


Fig. 6. Histogram of the detected community sizes for OSLOM, Game, COPRA, CFinder, Link and CIS created from the results for LFR networks with $n = 5000$, $\mu = 0.3$ and $O_n = 10\%$.

如图 7 所示，EAGLE、UEOC、iLCD 的分布大多移动在预定范围 20~100 之外。社区大小的分布可以用来解释性能和排名，具有这种分布的算法创建了相对较小的社区，所以性能不好。

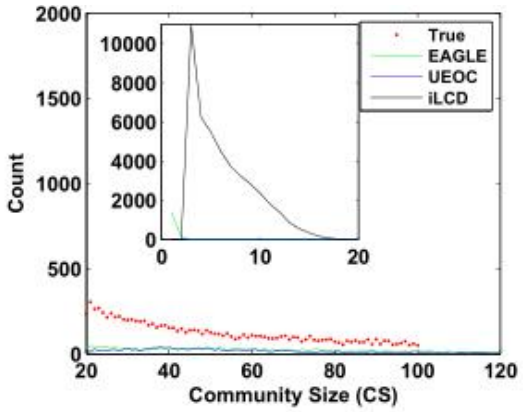


Fig. 7. Histogram of the detected community sizes for EAGLE, UEOC and iLCD crated from the results for LFR networks with $n = 5000$, $\mu = 0.3$ and $O_n = 10\%$.

五、识别 LFR 中的重叠节点

社区重叠表现为存在具有多个社区成员资格的节点，称之为重叠节点，识别重叠节点的能力对于评估社区检测算法的准确性是至关重要的。像 NMI 和 Omega 指数这样的措施只关注于提供算法精度的总体度量，但这些措施无法准确地描述节点级别发生的情况。定义：O^d_n 为检测到的重叠节点的数量，O^d_m 为检测到的成员的数量。注意：单独的重叠节点的数量 O^d_n 不足以准确地量化检测性能，因为它包含正的正样本(TP)和正的负样本(FP)。为了提供更精确的分析，我们认为重叠节点的识别是二元分类问题。只要 $O_m>1$ 或 $O^d_m>1$，节点就被标记为重叠，否则标记为不重叠。使用 F-分数作为检测精度的度量，F-分数定义为 $F = \frac{2 \cdot precision \cdot recall}{precision + recall}$ 其中 recall 是正确检测的重叠节点的数量除以重叠节点的真实数量 O_n，precision 是正确检测的重叠节点的数量除以检测到的重叠节点 O^d_n 的数量。F-分数考虑检测的数量和质量之间的平衡，并且分别达到其在 1 和 0 处的最佳和最差值。

图 8 和图 9 显示了对于不同的参数设置得到的 F-分数、precision 和 recall。

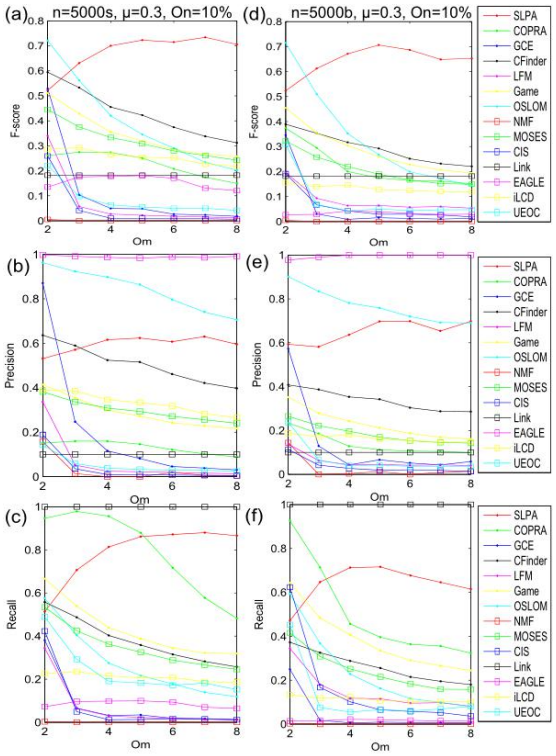


Fig. 8. Evaluations of overlapping node detection on LFR networks with low overlap density $O_n = 10\%$. Plots show F-score (together with precision and recall) as a function of the number of memberships for $n = 5000$ and $\mu = 0.3$. Results for small community size range are shown in the left column, and results for large community size range are shown in the righ column.

图 8

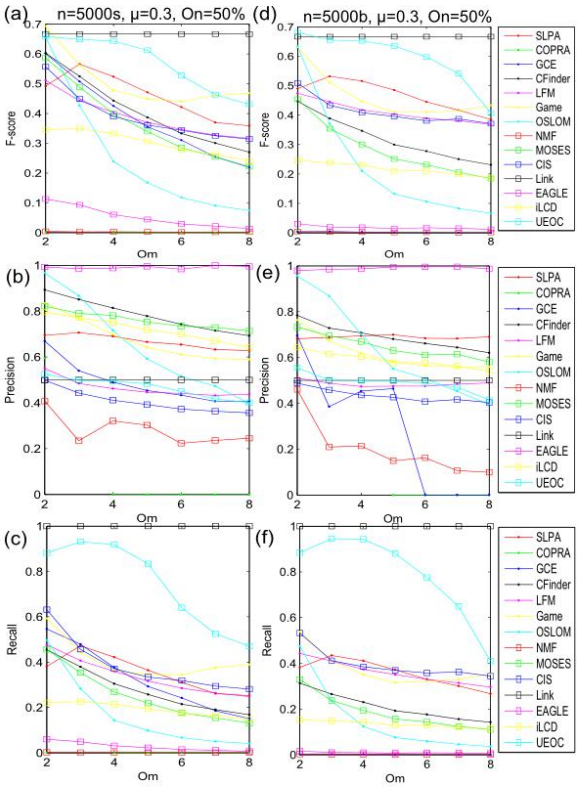


Fig. 9. Evaluations of overlapping node detection on LFR networks with high overlap density $O_n = 50\%$. Plots show F-score (together with precision and recall) as a function of the number of memberships for $n = 5000$ and $\mu = 0.3$. Results for small community size range are shown in the left column, and results for large community size range are shown in the righ column.

图 9

一般来说，在低和高重叠密度，算法在小社区上获得更好的 F-分数。但是，对于各种算法，原因是不同的。例如，EAGLE 在小社区的 F-分数的增加是由于 recall 的增加（见图 8 中(c)和(f)所示），而 iLCD 的增加主要是由于 precision 的增加（见图 8 中(b)和(e)所示）。此外，F-分数通常随着重叠分集 O_m 的增加而适度地衰减。显然，O_m 对 OSLOM 具有很大的影响，对于大的 O_m 具有快速下降。SLPA 是唯一一个在低重叠密度情况下与 O_m 具有正相关关系。

实验还揭示了一些算法的 precision 和 recall 的不平衡，部分是由于过检测（检测到比所存在更多的重叠节点）或欠检测（仅识别非常少的重叠节点）。极端的例子是 EAGLE 和 Link，尽管 EAGLE 实现了非常高的检测精度（见图 8 中的(b)和(e)所示），但是它遭受欠检测，仅识别非常少的重叠节点（从图 10 中可以看出），得到低的 recall 得分，结果导致低的 F-分数（见图 8 中的(a)和(b)所示）；对于 Link，即使它具有非常高的 recall（见图 8 中的(c)和(f)所示），但是它的 F-分数较低，这是因为 Link 遭受过度检测，声明了比预期更多的重叠节点（从图 10 中可以看出）。

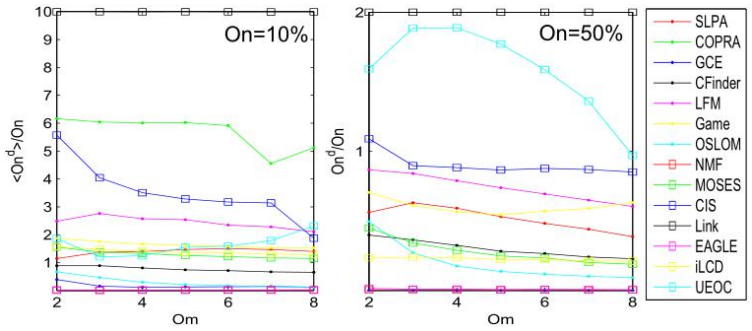


Fig. 10. The number of detected overlapping nodes (normalized by O_n) based on the results for LFR networks with $n = 5000b$ and $\mu = 0.3$. A value larger than 1 is possible.

图 10

关于不同社区大小范围的 F-分数的排名表示在表 IV 和 V 中，RS^s_F 和 RS^b_F 对应不同的重叠密度情况，RS^*_F 是两个社区大小范围的平均排名。

Table IV. The overlapping node detection ranking for $n = 5000$, $\mu = 0.3$ and low overlap density $O_n = 10\%$.

Rank	RS^s_F	RS^b_F	RS^*_F
1	SLPA	SLPA	SLPA
2	CFinder	CFinder	CFinder
3	OSLOM	Game	Game
4	Game	OSLOM	OSLOM
5	MOSES	COPRA	MOSES
6	iLCD	MOSES	COPRA
7	COPRA	Link	iLCD
8	Link	iLCD	Link
9	EAGLE	LFM	LFM
10	GCE	UEOC	UEOC
11	UEOC	CIS	EAGLE
12	LFM	EAGLE	GCE
13	CIS	GCE	CIS
14	NMF	NMF	NMF

Table V. The overlapping node detection ranking for $n = 5000$, $\mu = 0.3$ and high overlap density $O_n = 50\%$.

Rank	RS^s_F	RS^b_F	RS^*_F
1	Link	Link	Link
2	UEOC	UEOC	UEOC
3	Game	SLPA	SLPA
4	SLPA	Game	Game
5	CFinder	LFM	LFM
6	LFM	CIS	CFinder
7	CIS	CFinder	CIS
8	GCE	MOSES	MOSES
9	MOSES	OSLOM	OSLOM
10	iLCD	iLCD	iLCD
11	OSLOM	COPRA	GCE
12	COPRA	EAGLE	COPRA
13	EAGLE	NMF	EAGLE
14	NMF	GCE	NMF

为了便于比较，将 RS^*_F 复制到表 II 和 III 中。清楚地显示，在表 II 中，社区质量等级 $RS^*_{NM,Omega}$ 和节点质量等级 RS^*_F 可能有完全不同的性能表现。对于低重叠密度情况（如表 II），在检测社区中具有低排名的算法在识别重叠节点时，实际上可以具有很好的性能（例如 CFinder，iLCD 和 MOSES），而高排名的算法，可能分别由于欠检测和过检测而表现不好（例如 GCE 和 CIS），还有些则均具有非常稳定和良好的性能（SLPA）。这些观察表明，如果这些算法应用在重叠社区检测，旨在识别具有多个社区成员资格的节点，需要仔细地处理具有高 NMI 或 Omega 指数的算法，防止出现欠检测和过度检测情况。对于高重叠密度情况可以得出类似的结论。

六、最终排名

两种类型的排名提供了补充信息，排除过度检测和欠检测情况，综合考虑 RS^*_F 和 $RS^*_{NMI,Omega}$ 的前七位得出结论：

（a）对于低重叠密度网络，SLPA、OSLOM、Game 和 COPRA 提供比其他测试算法更好的性能；

（b）对于高重叠密度网络，SLPA 和 Game 提供更好的性能。

在真实世界社会网络上的测试

在高中友谊网络上检查算法的性能，其中已知地面实况。在调查中列出的学生的年级（从 7 到 12）对应 6 个社区（见图 11 所示）。

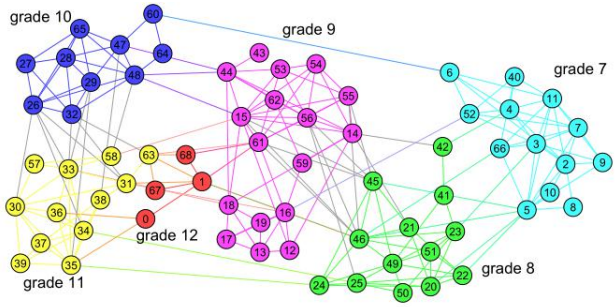


Fig. 11. High school network ($n = 69, \overline{k} = 6.4$). Colors represent known communities corresponding to grades ranging from 7 to 12. Grade 9 is separated into two subgroups that correspond to white (upper) and black (lower) students respectively. Numbers are the node id's.

结果表示在表 VI。

Table VI. Test on a high school friendship network.

Algorithm	Num. of communities	Overlapping nodes	NMI
CFinder	2	{12, 18}	0.1679
CIS	9	total 34	0.7495
COPRA	6	total 14	0.7966
EAGLE	4	{18}	0.4962
Game	10	total 14	0.4673
GCE	6	{0, 21, 45, 46, 61}	0.8333
iLCD	7	{5, 21, 26, 29, 31, 32, 33, 46, 61}	0.3713
LFM	7	{0, 45}	0.8134
Link	20	total 31	0.3155
MOSES	10	total 18	0.5037
NMF	7	{0, 12, 18, 45}	0.643
OSLOM	11	{45, 46}	0.4315
SLPA	6	{1, 42, 45, 59}	0.6788
UEOC	7	{0, 12, 18, 26, 29, 45}	0.8148

现有算法中存在的问题及未来研究的方向

测试结果表明在这样的网络中的检测仍然存在没有完全解决的问题。

一、算法可能对一些特定的结构敏感

例如，只有 SLPA，LFM，CIS 和 Game 在具有高度稀疏结构的网络（例如 P2P）中具有令人满意的性能，其中 COPRA 仅发现一个单一的巨大社区而且 GCE 一个也没有。所以需要研究出对于任意的网络结构都能够具有稳定高效的算法。

二、一些算法倾向于过度检测重叠

如 LFR 网络的情况。CIS 和 Link 在测试中性能低，因为他们发现相对于其他算法所标识的节点，有太多的重叠节点或成员资格。故可以在避免过度检测方面继续研究。

三、大多数算法未考虑权重

大部分算法针对的是未加权网络。但是，在许多应用中，权重具有重要的信息（例如，生物学研究中的相关网络）。明确考虑权重并允许重叠的算法（例如 CPMw 和 SLPAw）预计比其他算法具有优势。

四、两个基本问题

何时应用重叠方法和重叠有多重要。

关于重叠的必要性的讨论在很大程度上尚未探讨。

五、一种验证重叠的方法

在[Kelley 2009]中作者认为，对于 A 和 B 两个群体，A∩B 与集合 A-B 和 B-A 之间的性状相似性高于 A-B 和 B-A 之间的相似性。

结论

回顾了大量的社区检测算法以及质量度量方法和几个现有的基准。执行了多次测试，包括不同的网络结构和不同程度的重叠。质量评估在社区级别和节点级别上分别执行，结果表明，在具有高重叠密度和高重叠多样性的网络中的检测仍然具有改进的空间。