基于节点拓扑结构和属性的重叠社区检测算法

许加书，韩忠愿，顾惠健
（南京财经大学 信息工程学院，南京 210046）

摘　要：针对已有重叠社区检测通常只考虑节点的拓扑结构信息，忽略了节点的属性信息，导致数据间的重要结构遗漏的问题，提出了一种基于节点拓扑结构和属性相似度的重叠社区检测算法。首先，基于余弦相似度计算候选节点和局部社区之间的相似度，提高局部搜索效率；其次，改进局部模块度增量计算方法，使局部搜索模型收敛于发现潜在的真实社区；通过融合多个已检测到局部社区，计算隶属矩阵从而获取全局重叠社区结构；最后，在真实数据集上，与已有基于拓扑结构的社区检测算法进行实验对比。结论表明，该算法在模块度和 F_1-measure 的指标上取得了较好的表现且更适用于稀疏网络。

关键词：社区检测；节点属性；重叠社区；隶属矩阵；模块度
中图分类号：TP301　　**文献标志码**：A　　**文章编号**：1001-3695(2016)12-3615-05
doi:10.3969/j.issn.1001-3695.2016.12.022

Overlapping community detection with node structure and attribute

Xu Jiashu, Han Zhongyuan, Gu Huijian
(*College of Information & Engineering*, *Nanjing University of Finance and Economics*, *Nanjing* 210046, *China*)

Abstract: For overlapping community detection usually only considering the topology information of nodes, ignoring the node attribute information, causing the problem of missing important data structures, this paper proposed an overlapping community detection algorithm. The algorithm was based on the topology and the similarity of node attribute to find overlapping communities from a seed vertex. First, it computed the similarity between candidate nodes and the local community based on cosine similarity, improved the efficiency of local searching. Second, it improved the calculation method of local module-degree increments, made the local search model converge to find potential ground-truth communities. Third, it merged multiple local communities which have been detected and calculate a membership matrix, which can been seen as the global overlapping community structure of a graph. At the end, compared with the existing community detection algorithm based on topology on real data sets. The results shows the algorithm performs better on the modularity and F_1-measure. And it is more suitable for sparse networks.

Key words: community detection; node attribute; overlapping communities; membership matrix; modularity

0　引言

网络通常由相互连接的动态节点组成。社交网络、生物网络和计算机科学网络仅仅是复杂网络的一些代表，并且这些网络通常展示了具有拓扑特性的社区。社区或模块结构被认为具有现实社交网络的重要特性。

从网络中发现这些潜在的社区可视为将节点集合聚类成社区的问题。常见的重叠社区检测算法只注重节点与节点之间的连接信息这一种数据源。针对节点的拓扑结构数据源，重叠社区检测算法[1]通过节点的结构相似度来发现那些紧密相连的节点，进而划分出不同社区，但是往往忽略节点的属性信息。本文在结合了节点拓扑结构和属性的重叠社区检测的实验中验证了仅考虑节点拓扑结构进行重叠社区检测往往是不全面的，可能会忽略数据之间的重要结构。节点之间的连接信息可能会遇到缺失数据或噪声数据，节点的属性信息数据源可以弥补以上不足。结合节点拓扑结构信息和节点属性信息对重叠社区检测是至关重要的。然而，考虑节点的属性信息和节点的拓扑结构对社区检测仍是一种挑战。

目前已知的社区检测技术大致可以分为三种类型：a）基于全局的社区检测[2]，基于全局的社区检测方法将图看做一个整体，假设如果一个图区别于随机生成的图，那么该图就具有社区结构，例如 Newman 和 Girvan 的 null model[3]。b）基于节点相似度的社区检测[4,5]，基于节点相似度的方法假设社区是由众多结构相似的节点组成。传统方法包括层次聚类[6]、划分聚类[4]以及谱聚类方法[8]。c）基于局部的社区检测[9,10]，基于局部的方法比较了子图内部与外部的聚类密度。常见的方法有源于社交网络分析的 LS-set[8]，这种方法的条件是相当严格的，可以放宽到所谓的社区的弱定义[9]。LCD（local community detection）方法早已在十几年前被提出来用于包含特定种子节点的社区检测。根据评价局部社区检测质量方法的不同，可以将现存的 LCD 方法分为两类：基于度的方法通过研究节点的度来评价局部社区检测的质量[11,12]；基于相似度的方法是利用节点之间的相似度来评价局部社区划分质量[2,13]。

考虑到节点属性对数据结构的影响，本文在 LCD 算法的基础上，提出一种基于节点拓扑结构和节点属性的重叠社区检测算法 OCDNSA（overlapping community detection with node

收稿日期：2015-09-16；**修回日期**：2015-11-11

作者简介：许加书（1990-），男，江苏兴化人，硕士，主要研究方向为社区检测（jiashu42@sina.com）；韩忠愿（1963-），男，教授，博士，主要研究方向为自然语言处理、社区检测；顾惠健（1990-），男，硕士研究生，主要研究方向为社区检测.

structure and attribute),其主要创新点如下:a)提出局部搜索模型,结合网络拓扑结构和节点属性信息挖掘潜在的局部社区;b)基于余弦相似度计算候选节点和局部社区之间的相似度,提高了局部搜索效率;c)改进局部模块度增量计算方法,使局部搜索模型收敛于发现潜在的真实社区。

然而,针对真实的社会网络,伴随网络中节点属性缺失、不完整等一系列问题,上述基于属性相似性的局部搜索存在一定的局限性。因此,在后续的工作中,将继续努力改进算法以更好地应对节点属性缺失、不完整等问题。

1 基于节点拓扑结构和属性的重叠社区检测算法

1.1 问题描述

已知给定一个节点带有属性的无向图 $G=(V, E, X)$,其中 V 代表节点的集合($|V|=n$),E 代表连接节点的边的集合($|E|=m$),$\boldsymbol{X}$ 表示 $n\times y$ 的节点属性矩阵。X_{iy} 表示节点 i 的第 y 属性特征值。LCD 用来发现图的一个子集 $G'=(V', E')$,G' 是以属于 V 的节点 v_s 为种子节点发现的子集。除了已经检测到的局部社区,只能获取关于此网络的部分信息,包括已检测到社区的邻居节点,邻居节点的连接信息。为了方便处理,将发现的网络定义为三部分:已检测到的社区 C、社区 C 的邻域 B 以及未知区域 U。邻域 B 指的是社区 C 中所有节点的邻居节点集合。未知区域 U 指的是除已检测到社区 C 中节点和邻域 B 中节点的其余所有节点。算法开始,将种子节点 v_s 加入到社区 C 中($v_s\in V, 1\leqslant s\leqslant n$),然后将种子节点 v_s 的所有邻居节点加入到邻域 B 中,通过计算模块度增量来决定邻居节点是否计入到社区 C 中。在这种局部搜索模型中,社区 C 是由节点 v_s 根据定义扩展得到的。

现有的重叠社区检测算法 MOSES[13]、Link[14]、LCEO[15] 只是考虑了节点之间的拓扑结构,忽略了节点的属性信息对社区检测的重要影响。实际上,节点的属性信息对社区检测、发现节点之间的联系也存在着举足轻重的地位。基于以上认识,本文借鉴了局部社区检测 LCD 算法的思路,提出了基于节点结构和属性的重叠社区检测算法 OCDNSA,并将真实的在线金融论坛数据集和斯坦福大学的三个社交网络数据集应用到 MOSES、Link、LCEO、OCDNSA 四个算法中,比较其优劣。

1.2 计算方法

为实现基于节点拓扑结构和属性的局部社区检测 OCDNSA 算法,依据文献[16]作出以下定义方便计算:

定义 1 局部模块度。社区 C 的局部模块度用 $W(C)$ 表示。

$$W(C)=\left(\frac{I(C)}{|C|^2}-\frac{O(C)}{|C||C^c|}\right) \tag{1}$$

其中:C^c 是 C 的补集,$I(C)=\sum_{v_i,v_j\in C}A_{i,j}$,$O(C)=\sum_{v_i\in C,v_j\in C^C}A_{i,j}$,$\boldsymbol{A}_{ij}$ 是图 G 的一个 $n\times n$ 的邻接矩阵。

基于模块度定义,得到以下定理:当社区 C 的内部聚类密度大,而外部聚类密度小时,一个社区 C 的模块度值会增加。

证明 根据定义得知 $I(C)$ 是社区 C 内部边数值的两倍,$O(C)$ 是介于社区 C 和剩下网络的边数。使用 $|C|^2$ 而不是 $|C|(|C|-1)$ 来标准化目的是方便得到模块度增加的定义,在实际计算中这个几乎没变化。局部模块度可以描述为内部聚类密度减去外部聚类密度。在模块度定义中,依据文献[15]作了部分调整:

$$\hat{W}(C)=|C||C^c|\left(\frac{I(C)}{|C|^2}-\frac{O(C)}{|C||C^c|}\right) \tag{2}$$

其中:$|C||C^c|$ 避免了极大或极小的社区,能够产生均衡的解决方法。基于文献[17]的局部模块度定义,提出如下局部搜索模型:如图 1 所示,令 C 是一个以节点 v_s 为种子节点生成的种子社区,B 为社区 C 中所有节点的邻居节点集合。区分三种连接:社区 C 内部连接 L,社区 C 和节点 v_i 的连接 L_{in} 以及介于社区 C 和邻域 B 中其他节点的连接 L_{out}。为了简化计算,本文用 L 和 k_i 表示内部连接(k_i 为节点 v_i 的度),得到 $L_{\text{in}}=a_1L=a_2k_i$,$L_{\text{out}}=b_1L$,其中 $b_1\geqslant 0$,$a_2\geqslant 1/L$,$a_2\geqslant 1/k_i$(邻域 B 中的节点至少有一个邻居节点存在于社区 C 中)。因此,文献[17]中局部模块度可改写为

$$\hat{W}(C)=\frac{n-|C|}{|C|}\cdot 2L-(a_1+b_1)L \tag{3}$$

其中:n 为节点数,$|C|$ 为已发现社区内部节点数。

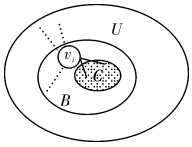

图 1 局部搜索模型图

定义 2 模块度增量。社区 C 加入节点 v_i 后的模块度增量表示为

$$\Delta\hat{W}_C(v_i)=\left(\frac{n-|C|-1}{|C|+1}\cdot 2L(1+a_1)-(k_i+b_1L-a_2k_i)\right)-$$

$$\left(\frac{n-|C|}{|C|}\cdot 2L-(a_1+b_1)L\right)=2n\cdot\frac{a_2k_i|C|-L}{|C|(|C|+1)}-k_i \tag{4}$$

其中:$|C|$ 表示当前社区 C 中已有节点的个数,k_i 表示节点 v_i 的度。$\Delta\hat{W}(v_i)$ 作为标准决定候选节点是否加入到社区 C。

定义 3 属性相似度。节点 v_i 和节点 v_j 之间的属性相似度表示为

$$S_{i,j}=\frac{\sum X_{iy}\cdot X_{jy}}{\sqrt{\sum X_{iy}^2}\cdot\sqrt{\sum X_{jy}^2}} \tag{5}$$

其中:X_{iy} 表示 v_i 节点的第 y 个属性值。

1.3 算法流程

这部分介绍基于节点拓扑结构和属性相似度的重叠社区检测算法的算法流程。首先,对节点进行排序。计算出节点的聚类系数,根据聚类系数降序排序,如果有节点的聚类系数相同,则根据节点的度降序排序。其次是核心节点集合挖掘,这部分迭代遍历未被访问的节点。考虑到节点的聚类系数越大,节点与其邻节点余弦相似度越大,它们越可能被划分到同一个核心节点集合中。因此,选出聚类系数最大的节点 v_s,并将其初始化为社区 C。将 v_s 的邻节点加入邻域 B,通过计算社区 C 中节点与邻域 B 中节点的属性相似度。选出邻域 B 中与社区 C 中拥有最高属性相似度的邻节点 v_i 作为候选节点加入到局部社区 C 中。此时计算 $\Delta\hat{W}(v_i)$,如果 $\Delta\hat{W}(v_i)$ 大于 0 ,则将节点 v_i 插入社区 C 中,否则将其从 B 中移除。通过不停迭代所有节点,直到整个以 v_s 为种子节点的局部社区 C 被检测出来。将发现的局部社区 C 中度最大的节点定义为此社区的核心节点,并作为此社区的标志。

算法 1 基于节点拓扑结构和节点属性的局部社区检测算法。

输入:$G=(V, E, X)$, v_s。

输出:$C(v_s)$, $l(C(v_s))$。

1 $C\leftarrow\{v_s\}$

2 $B\leftarrow\{v_i\mid v_i\in\Gamma(v_s)\}$

```
3  while B≠ ∅ do
4      use Eq(5) to compute S_{i,j}
5      v_i* = arg max_{v_i∈B} Σ v_i ∈ C^{S_{i,j}}
6    use Eq(3) to computer ΔŴ(v_i)
7    if ΔŴ>0 then
8        C←C∪{ v_i* }
9        B←B∪{ v_j' | v_j' ∈ Γ(v_i*), v_j' ∉ C}
10   else
11       B←B−{v_i*}
12   end if
13 end while
14 C(v_s) ← C
15 l(C(v_s)) ←arg max_{v_j∈C} k_j
```

算法 2　基于节点拓扑结构和属性的重叠社区检测算法。

输入:$G=(V, E, X)$,K。

输出:$U=[U_{ik}]$, $i=1,\cdots,n$, $k=1,\cdots,K$。

```
1  /* rank node according to clustering coefficient and degree */
2  calculate clustering coefficient
3  sort by clustering coefficient and degree of nodes descend
4  for s=1; s<=n; s++ do
5    [C(v_s), ls]← OCDNSA(v_s) // label v_s inactive, ls is the
     node with the max degree
6    Vector ← ls //add ls into the Vector if ls not exist in Vector
7    if len(Vector) >K then
8    break
9  end for
10 for i=1; i≤n; i++ do
11   for k=1; i≤K; k++ do
12     use Eq(6) to compute U_{ik}
13   end for
14 end for
```

值得注意的是,在局部社区检测算法 LCD 中,候选节点的选择依据是拥有最大结构相似度与属性相似度的节点,然后计算 $\Delta\hat{W}(v_i)$来判断是否将节点 v_i 加入局部社区 C 中。结构相似度代表着图的局部连接的稠密度。局部社区 C 的节点与邻域 B 中的节点结构相似度越高意味着这两个节点共享的邻居越多,这两个节点在一个社区的概率越大。同理,属性相似度代表着节点之间的相关性。属性相似度越高,意味着节点在同一个社区的概率越大。

重叠社区的检测:对于一个重叠分区,重叠社区可以表示为一个隶属矩阵 $U=[u_{i,k}]$,$i=1,\cdots,n$, $k=1,\cdots K$,其中 $u_{i,k}\in[0,1]$表示节点 i 隶属社区 C_k 的概率。在 OCDNSA 算法的基础上,可以将其扩展为基于节点拓扑结构和节点属性的重叠社区检测算法。在执行 OCDNSA 算法时不必遍历所有节点,只需遍历大部分节点。通过检测到的局部社区 C 和节点 i 的邻居节点所在社区,可以计算出未遍历节点所在社区的概率,$u_{i,k}$ 可定义为

$$u_{i,k}=\frac{\sum_{s=1,\cdots,n\wedge l_s=k}\chi(v_i,C(v_s))}{\sum_{s=1,\cdots,n}\chi(v_i,C(v_s))} \tag{6}$$

当节点 v_i 在社区 C 中,$\chi(v_i, C(v_s))$等于 1;否则为 0。当隶属矩阵 U 被求解出来时,重叠社区亦被求出。

1.4　算法复杂度分析

基于节点拓扑结构的社区检测算法计算候选节点与已发现社区中所有节点的拓扑结构相似度。比如,LCEO 算法考虑候选节点与社区内部所有节点的结构相似性的和,是一种线性的叠加。而本文中的基于节点拓扑结构和属性的社区检测算法分别计算所有的候选节点的属性与已发现社区内部所有节点的向量中心的相似度 $S_{i,j}$,其中社区内部所有节点的向量中心的计算是迭代更新的,比如,已发现社区内部节点有 v_1、v_2、v_3、v_4,内部节点的向量中心为$[x_{1a}+x_{2a}+x_{3a}+x_{4a},\cdots, x_{1n}+x_{2n}+x_{3n}+x_{4n}]/4$(第一个下标表示节点序号,第二个下标表示属性序号),候选节点 v_5 与向量中心的属性相似度依据式(5)计算可得,同理计算其他候选节点与向量中心的属性相似度,选出属性相似度最大的候选节点计算模块度增量 $\Delta\hat{W}(v_i)$,若 $\Delta\hat{W}(v_i)$大于 0 则将候选节点加入到社区中,并重新计算社区内所有节点的向量中心。若 $\Delta\hat{W}(v_i)$小于 0 则选出属性相似度次大的候选节点计算模块度增量。

算法 1 主要是计算节点属性相似度,即算法 1 的第 4 行。算法首先计算出已加入社区 C 中的所有节点属性值的向量中心,时间复杂度为 $O(1)$。再根据式(5)分别计算向量中心与所有候选节点的属性相似度,时间复杂度为 $O(n)$。因此算法 1 的总的时间复杂度为 $O(n)$。而重叠社区检测算法 2 的大部分运行时间被消耗在 10 ~ 13 行,即计算节点隶属矩阵,时间复杂度为 $O(nK)$,其中 n 为节点数,K 是社区个数。因此,基于节点拓扑结构和属性的重叠社区检测算法总的时间复杂度为 $O(nK)$。

算法 1 的空间复杂度主要有两部分组成:一是属性相似度的计算,空间复杂度为 $O(y^2)$,y 为节点属性数目;二是模块度增量的计算,空间复杂度为 $O(nk)$,k 是所有节点的平均度。因此,算法 1 的总的空间复杂度为 $O(y^2+nk)$。算法 2 的空间复杂度为 $O(nK)$。因此,基于节点拓扑结构和属性的重叠社区检测算法总的时间复杂度为 $O(y^2+nk+nK)$。

1.5　算法流程示例

下面将结合一个具体案例图 1 展示怎么通过基于节点拓扑结构和节点属性的重叠社区检测算法检测到重叠社区。原始网络包含了 12 个节点、20 条边以及 12 个节点的 5 个属性特征值。算法开始,选出聚类系数最大的节点 v_1,并初始化为局部社区 $C_1(0)=\{v_1\}$,邻域 $B_1(0)=\{v_2,v_3,v_5,v_9\}$。然后从邻域 $B_1(0)$中选出与局部社区 $C_1(0)$中节点 v_1 的属性相似度最大的节点 v_5。再通过计算 $\Delta\hat{W}(v_5)$的值决定是否将 v_5 加入到 $C_1(0)$。计算出的 $\Delta\hat{W}(v_5)$大于 0,节点 v_5 加入到局部社区 $C_1(0)$中,并更新邻域 $B_1(1)=\{v_2,v_3,v_6,v_7,v_8,v_9\}$。由于邻域 $B_1(1)$未空,进行下一轮迭代。第二轮迭代结束后,此时局部社区为 $C_1(2)=\{v_1,v_5,v_9\}$,邻域为 $B_1(2)=\{v_2,v_3,v_6,v_7,v_8,v_{10},v_{11},v_{12}\}$。当迭代结束,即邻域 B 为空后,算法就退出本轮社区更新。经过 10 次迭代,最终收敛的局部社区 $C_1=C_1(10)=\{v_1,v_5,v_9\}$,选择度最大的节点 v_5 作为其社区标签,并将 v_1 标记为无效点(下次选择聚类系数最大的节点时将其排除在外)。同样,对其他节点作同样的迭代,直至所有节点都被标记为无效点。此时,全局的非重叠社区就被检测出来。通过计算隶属矩阵 $U=[u_{i,k}]$,就可以检测出全局的重叠社区结构。

2　实验

2.1　实验设置

数据集:本文实验选用“斯坦福大学大型网络数据集”中的三个社交网络数据集[14]作为测试集,分别为 facebook、gplus 和 twitter,以及一个本文抓取的在线金融论坛的数据集。表 1 列出了这四个数据集的节点数、边数以及属性数。facebook、gplus 和 twitter 社交网络数据集中节点编号表示社交网络中的每个用户,边集表示用户之间有联系,属性信息包含用户的性

别、国籍、地区、工作单位、所在大学等。guba 数据集中节点表示了在线金融论坛的用户，一条边表示两个用户在两个或两个以上的帖子下有互动行为，则两个用户构成一条边，节点属性信息用特征值 0 或 1 表示每个用户是否在沪深 300 指数的股票论坛下有过评论或发帖行为。

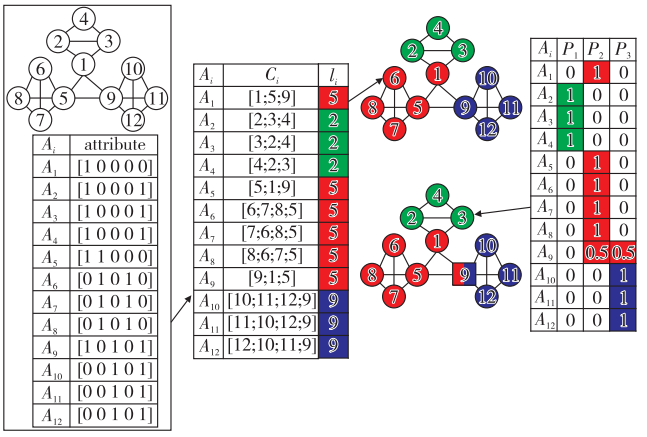

图 2 基于节点拓扑结构和节点属性的重叠社区检测算法示例图

表 1 实验数据名称及特征描述

数据集名称	节点数	边数	属性数目	平均度
facebook	4039	88234	9	43.69
gplus	3335	140598	183	84.32
twitter	2155	51973	32	48.23
guba	13714	54273	300	7.92

比较方法：观察这四个网络的平均度可以得知来自斯坦福大学的三个网络数据集 facebook、gplus、twitter 相对 guba 网络较稠密，而 guba 网络相对稀疏。因此，这四个网络数据集中既包含了稠密网络也包含了稀疏网络，完全适用于已有的基于节点拓扑结构的社区检测算法。实验选取了四个算法 MOSES、Link、LCEO、OCDNSA 在四个网络上社区检测得到的 10 ~ 100 个(间隔为 10)结果，比较在不同社区个数计算得到的模块度 Q 值、PRF 指标以及算法运行时间。

评价指标：本文选取了两个指标来评价社区划分的优劣，即内部指标—模块度 Q(Modularity)和外部指标—F_1-measure。

其中，模块度指标 Q[17] 定义为

$$Q=\frac{1}{2m}\cdot\sum_{vw}\left(A_{vw}-\frac{k_v k_w}{2m}\right)\cdot\delta_{vw} \tag{7}$$

其中：m 为边数；用 δ_{vw} 表示节点 v、w 是否在同一社区中；矩阵 A 表示图 G 的邻接矩阵；A_{vw} 表示节点 v、w 之间有无连接。设置 C_k 是第 K 个重叠社区，其中 $C_1\cup\cdots\cup C_K\subseteq V$，以及隶属关系阈值 α，$0<\alpha\leqslant 1$，来控制想要观测到的重叠社区规模。

准确率(P)、召回率(R)和综合评价指标(F)来评价系统摘要的质量，计算公式分别为

$$P=\frac{TP}{TP+FP},R=\frac{TP}{TP+FN},F=\frac{2\cdot PR}{P+R} \tag{8}$$

根据文献[15]定义，$C_k(\alpha)=\{v_i\mid u_{i,k}\geqslant\alpha\}$。因此，计算重叠社区中的每个节点来获取本文算法得到的平均准确率、召回率和 F_1-measure 值。根据文献[15]计算准确率 $P(\alpha)$、召回率 $R(\alpha)$以及 $F_1(\alpha)$为

$$P(\alpha)=\frac{\sum_{k=1,\cdots,K}\sum_{v_i\in C_k(\alpha)}\frac{|C_k(\alpha)\cap T_i|}{|C_k(\alpha)|}}{\sum_{k=1,\cdots,K}\sum_{v_i\in C_k(\alpha)}1} \tag{9}$$

$$R(\alpha)=\frac{\sum_{k=1,\cdots,K}\sum_{v_i\in C_k(\alpha)}\frac{|C_k(\alpha)\cap T_i|}{|T_i|}}{\sum_{k=1,\cdots,K}\sum_{v_i\in C_k(\alpha)}1} \tag{10}$$

$$F_1(\alpha)=\frac{2P(\alpha)R(\alpha)}{P(\alpha)+R(\alpha)} \tag{11}$$

2.2 实验结果

结合表 2 分析，对于数据集 facebook、gplus、guba，基于节点拓扑结构和节点属性的重叠社区检测算法在不同社区个数计算得到的模块度 Q 值要普遍好于仅仅考虑节点拓扑结构的社区检测算法 MOSES[13]、Link[14] 和 LCEO[15]，说明了同时考虑节点拓扑结构和节点属性信息的社区检测算法划分社区效果更好。这也证实了在社区检测算法的数据源中，同时考虑节点拓扑结构和节点属性信息的社区检测能够更加有效地发现重要的数据特征。四个数据集在四种算法下计算得出的模块度 Q 值变化也相对较平稳。但也发现了异常点，facebook 在 LCEO[15] 与 OCDNSA 算法检测到 10 个社区的模块度 Q 值都明显低于 facebook 其他数量社区模块度 Q 值，检测到 10 个以上社区的模块度 Q 值则趋于稳定，而 facebook 在 MOSES 与 Link 算法下的模块度 Q 值变化则比较平稳。其他三个数据集在同一种算法下检测到不同数目的社区时的模块度变化相对也比较平稳。这说明同一个数据集在同一个算法检测到不同个数的社区时，社区检测的效果可能相差较大。可能是较小的社区个数得到的模块度 Q 值并不能很好地体现社区检测算法的优劣。对于数据集 twitter，在 OCDNSA 算法下计算得到的模块度 Q 值普遍低于在 LCEO 算法下计算得到的模块度 Q 值，但高于 MOSES 和 Link 算法下计算得到的模块度 Q 值。这可能是由于 twitter 数据集节点关键属性信息的数量和质量相对较少导致。由此可以看出重叠社区检测算法在融合节点属性信息和节点拓扑结构信息时，需要注重属性信息的数量与质量，这样才能获得更好的社区检测效果。

表 2 四种算法下得到的模块度 Q 值

dataset	algorithm	Q_{10}	Q_{20}	Q_{30}	Q_{40}	Q_{50}	Q_{60}	Q_{70}	Q_{80}	Q_{90}	Q_{100}
facebook	LCEO	0.101	0.578	0.663	0.663	0.664	0.675	0.673	0.687	0.689	0.687
	MOSES	0.417	0.431	0.475	0.494	0.478	0.477	0.485	0.482	0.468	0.461
	Link	0.007	0.009	0.011	0.011	0.012	0.012	0.014	0.015	0.015	0.015
	OCDNSA	0.230	0.707	0.698	0.683	0.696	0.690	0.758	0.766	0.761	0.756
gplus	LCEO	0.510	0.486	0.468	0.459	0.448	0.441	0.437	0.435	0.432	0.431
	MOSES	0.338	0.235	0.173	0.150	0.125	0.110	0.103	0.096	0.094	0.090
	Link	0.023	0.010	0.008	0.008	0.038	0.043	0.043	0.039	0.036	0.035
	OCDNSA	0.499	0.481	0.474	0.461	0.456	0.450	0.447	0.443	0.442	0.441
guba	LCEO	0.238	0.260	0.350	0.350	0.355	0.355	0.364	0.367	0.387	0.401
	MOSES	0.400	0.414	0.507	0.476	0.515	0.513	0.516	0.534	0.527	0.537
	Link	0.457	0.356	0.361	0.341	0.506	0.561	0.568	0.499	0.512	0.512
	OCDNSA	0.594	0.624	0.702	0.699	0.708	0.727	0.745	0.728	0.737	0.735
twitter	LCEO	0.337	0.401	0.401	0.395	0.388	0.383	0.380	0.375	0.372	0.371
	MOSES	0.187	0.189	0.184	0.176	0.169	0.173	0.173	0.145	0.144	0.130
	Link	0.002	0.028	0.028	0.028	0.028	0.021	0.021	0.021	0.023	0.023
	OCDNSA	0.335	0.337	0.330	0.325	0.323	0.321	0.319	0.318	0.316	0.315

在比较准确率、召回率以及 F_1 值时，选取了能够观察到真实社区的三个数据集 facebook、gplus 以及 twitter。阈值 α 设置为[0.1,1]，间隔为 0.1。结合图 2 ~ 5，观测到本文的方法在 facebook 数据集上计算得到的平均准确率 P、召回率 R 以及 F_1-measure 值，要好于其他三种算法。而在数据集 twitter 和 gplus 上，本文的方法计算得出的平均准确率 P、召回率 R 和 F_1-measure 值与 LCEO 算法和 MOSES 算法相差不大，但是相对于 Link 算法，表现不好。同时，也观测到三种数据集在 Link 算法下计算得出的平均准确率 P、召回率 R 以及 F_1-measure 值变幅较大。这表明了本文的方法在稀疏网络取得的效果要好于稠密网络，这可能是网络的稀疏度对本文的方法有一定的影响。

在比较算法运行时间时，结合图 6，也观测到 OCDNSA 算法运行不同数据集时，在四个算法中所花费的时间是最少的，

MOSES 算法花费的时间最多，LCEO 算法和 OCDNSA 算法所需时间相差不大，但总体不如 OCDNSA。综合来看，本文的算法 OCDNSA 在时间效率上取得了最好的效果。

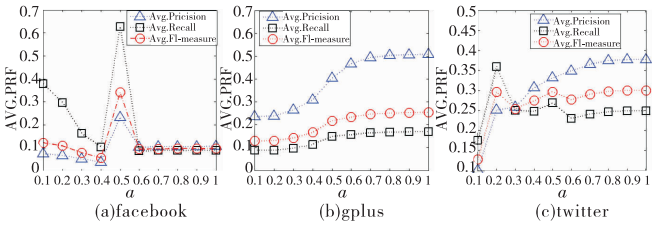

图 3　Link 算法不同阈值下的平均 PRF

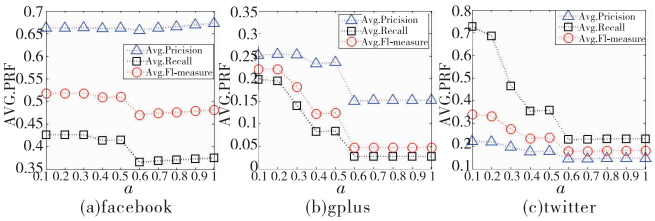

图 4　MOSES 算法不同阈值下的平均 PRF

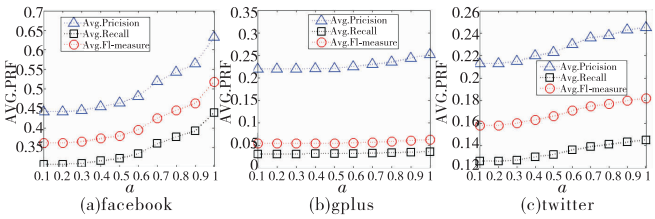

图 5　LCEO 算法不同阈值下的平均 PRF

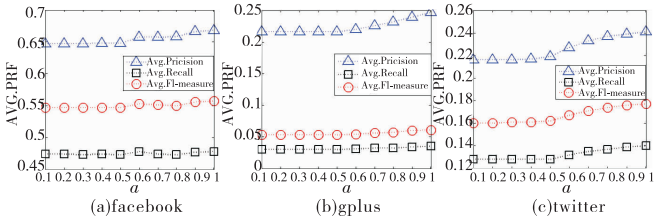

图 6　OCDNSA 算法不同阈值下的平均 PRF

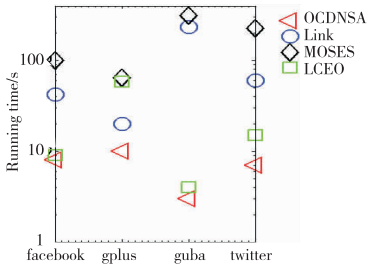

图 7　算法运行时间比较

3　结束语

本文通过已有工作从基于节点拓扑结构和节点属性的重叠社区检测算法中得到启示：节点的拓扑结构信息和节点的属性信息对社区检测关系重大。只考虑节点的拓扑信息时，若遇到节点拓扑信息缺失或错误时，会丢失数据之间的重要结构信息。因此，本文结合了两种数据源，一方面可以相互弥补数据源的缺失、错误等常见问题，另一方面也能够正确有效地获取数据的重要结构信息。本文提出基于节点拓扑结构和节点属性相似度的重叠社区检测算法 OCDNSA。通过在 facebook、gplus、guba 和 twitter 四个数据集上，MOSES、Link、LCEO、OCDNSA 四种算法实验的对比结果，验证了基于节点拓扑结构和节点属性的重叠社区检测算法的有效性，在模块度指标 Q 值、准确率、召回率、F_1-measure 值以及算法运行时间方面均取得了较好的效果。同时在结合节点属性和节点拓扑结构时进行的社区检测效果对于 twitter 数据集取得的效果不太好，在数据集 gplus 上计算得出准确率 P、召回率 Q 以及 F_1-measure 值不太理想。需要进一步研究本文方法在稀疏网络和稠密网络上的效率和有效性。

参考文献：

[1] Xie Jierui, Kelley S, Szymanski B K. Overlapping community detection in networks: the state of the art and comparative study[J]. ACM Computing Surveys, 2013, 45(4): Article No. 43.

[2] Chen Hunghsuan, Gou Liang, Zhang Xiaolong, *et al.* Discovering missing links in networks using vertex similarity measures[C]//Proc of the 27th Annual ACM Symposium on Applied Computing. 2012: 138-143.

[3] Newman M E J, Girvan M. Finding and evaluating community structure in networks[J]. Physical Review E, 2004, 69(2): 026113.

[4] Hlaoui A, Wang Shengrui. A direct approach to graph clustering [C]//Proc of IASTED International Conference on Neural Networks and Computational Intelligence. 2004: 158-163.

[5] Von Luxburg U. A tutorial on spectral clustering[J]. Statistics and Computing, 2007, 17(4): 395-416.

[6] Hastie T, Tibshirani R, Friedman J. The elements of statistical learning[M]. New York: Springer, 2009.

[7] Yang Bo, Liu Jiming. Discovering global network communities based on local centralities[J]. ACM Trans on the Web, 2008, 2(1): Article ID 9.

[8] Luccio F, Sami M. On the decomposition of networks in minimally interconnected subnetworks [J]. IEEE Trans on Circuit Theory, 1969, 16(2): 184-188.

[9] Radicchi F, Castellano C, Cecconi F, *et al.* Defining and identifying communities in networks[J]. Proceedings of the National Academy of Sciences of the United States of America, 2004, 101(9): 2658-2663.

[10] Bagrow J P, Bollt E M. Local method for detecting communities[J]. Physical Review E, 2005, 72(4): 46108.

[11] Bagrow J P. Evaluating local community methods in networks[J]. Journal of Statistical Mechanics: Theory and Experiment, 2007, 2007(5): P05001.

[12] Huang Jianbin, Sun Heli, Liu Yaguang, *et al.* Towards online multiresolution community detection in large-scale networks [J]. PloS one, 2011, 6(8): e23829.

[13] McDaid A, Hurley N. Detecting highly overlapping communities with model-based overlapping seed expansion[C]//Proc of International Conference on Advances in Social Networks Analysis and Mining. [S.l.]: IEEE, 2010: 112-119.

[14] Ahn Y Y, Bagrow J P, Lehmann S. Link communities reveal multiscale complexity in networks[J]. Nature, 2010, 466(7307): 761-764.

[15] Bu Zhan, Gao Guangliang, Wu Zhiang. Local community extraction for non-overlapping and overlapping community detection[M]// Advanced Data Mining and Applications. 2014: 98-111.

[16] Zhao Yunpeng, Levina E, Zhu Ji. Community extraction for social networks[J]. Proceedings of the National Academy of Sciences, 2011, 108(18): 7321-7326.

[17] Newman M E J. Fast algorithm for detecting community structure in networks[J]. Physical Review E, 2004, 69(6): 066133.

[18] Leskovec J, Krevl A. Stanford large network dataset collection[DB/OL]. 2014-06. https://snap.stanford.edu/data/index.html.