计算机科学
COMPUTER SCIENCE

基于代表性节点扩张的保持社区结构的图采样算法

宏宇, 陈鸿昶, 张建朋, 黄瑞阳, 李邵梅

引用本文

宏宇, 陈鸿昶, 张建朋, 黄瑞阳, 李邵梅. 基于代表性节点扩张的保持社区结构的图采样算法[J]. 计算机科学, 2024, 51(4): 117-123.

HONG Yu, CHEN Hongchang, ZHANG Jianpeng, HUANG Ruiyang , LI Shaomei. Graph Sampling Algorithm Based on Representative Node Expansion to Maintain CommunityStructure [J]. Computer Science, 2024, 51(4): 117-123.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.230100018

基于代表性节点扩张的保持社区结构的图采样算法

宏　宇[1]　陈鸿昶[2]　张建朋[2]　黄瑞阳[2]　李邵梅[2]
1 郑州大学网络空间安全学院　郑州 450000
2 中国人民解放军战略支援部队信息工程大学信息技术研究所　郑州 450000
(1161738832@qq.com)

摘　要　作为一种能够简化大规模图并保留其指定属性的方法,图采样被广泛应用于现实生活中。然而当前研究大多集中于保留节点级的性质,如度分布等,而忽略了图的社区结构等更为重要的信息。针对此问题,提出了一种保持社区结构的图采样算法。算法主要分为两个步骤,第一步为初始化社区代表点,根据提出的节点重要度计算公式算出节点的重要度,然后选出每个社区的代表性节点;第二步为社区结构扩张,针对每个社区,选择可能引入最少额外邻居的节点加入社区中,直到达到该社区节点上限。在多个真实数据集上进行了对比实验,使用多个评价指标来评估实验结果。实验结果表明,所提出的采样算法能够很好地保持原始图的社区结构,为大规模图的社区结构采样提供了可行的解决方案。

关键词:图采样;社区结构;代表性节点;扩张;重要度
中图分类号　TP391

Graph Sampling Algorithm Based on Representative Node Expansion to Maintain Community Structure

HONG Yu[1],CHEN Hongchang[2],ZHANG Jianpeng[2],HUANG Ruiyang[2] and LI Shaomei[2]
1 College of Cyberspace Security,Zhengzhou University,Zhengzhou 450000,China
2 Institute of Information Technology,PLA Strategic Support Force Information Engineering University,Zhengzhou 450000,China

Abstract　Graph sampling is widely used in real life as a method to simplify large-scale graphs and retain specified properties. However,most of the current research focuses on preserving node-level properties,such as degree distribution,while ignoring more important information such as the community structure of graphs. To solve this problem,a graph sampling algorithm is proposed to maintain the community structure. The algorithm is divided into two steps. The first step is to initialize the community representative points,and the node importance is calculated according to the proposed node importance calculation formula,and then the representative nodes of each community are selected. The second step is to expand the community structure. For each community,it selects the node that can introduce the least additional neighbors to join the community until the upper limit of the community node is reached. Comparative experiments are conducted on a number of real data sets,and multiple evaluation indicators are adopted to evaluate the experimental results. Experimental results show that the proposed sampling algorithm can well maintain the overall community structure,and provides a feasible solution for sampling community structure of large-scale graphs.

Keywords　Graph sampling,Community structure,Representative nodes,Expansion,Importance

1　引言

在现实生活中,图已经变得很普遍,它的应用范围非常广泛,包括社交网络分析、生物信息学、电信网络、引文网络等。对这些图的分析有着重要意义,例如社区发现算法可以挖掘出图中紧密的社区结构,在现实生活中可以根据社区中成员的位置、行为以及个人喜好等为他们提供特定的服务;异常点检测可以找出异于其他节点行为的异常节点,可以用于欺诈检测等。然而,随着技术的进步以及互联网的广泛使用,现实生活中产生的数据也越来越多,在各种应用中,数百万甚至数千万节点的大规模图无处不在。在如此规模的图上进行挖掘、分析非常困难,主要表现在文件太大,数据无法全部加载到内存中并且计算时间代价很大。因此,如何有效地处理这些大图数据成为了图挖掘领域的一个关键问题。

到稿日期:2023-01-03　返修日期:2023-12-19
基金项目:国家自然科学基金(62002384);中国博士后科学基金面上项目(2020M683760);嵩山实验室项目(221100210700-03)
This work was supported by the National Natural Science Foundation of China (62002384), China Postdoctoral Science Foundation (2020M683760) and Songshan Laboratory Project(221100210700-03).
通信作者:张建朋(j_zhang_edu@sina.com)

一种广泛应用的解决方法是图采样技术，该方法可以通过采样原始图中的一部分节点和边得到一个规模更小的子图，同时还会保留一些原始图中的性质，如度分布和聚类系数等。而对图的分析通常都非常依赖于图中的某些性质，如果该子图能够很好地保存原始图中的这些性质，那么可以选择对规模较小的子图而不是规模巨大的原始图进行分析，得到和原始图上近似的结果。然而当前图采样的研究大多侧重于保存节点相关的性质，如度分布、介数中心性和紧密中心性等，忽略了真实世界中更为重要的社区结构特征。由于社区结构是相对于全局而不是节点而言的，因此对社区结构的研究更为复杂和必要。

本文提出了一种保持社区结构的图采样算法 GSRNCS (Graph Sampling algorithm based on Representative Node expansion to maintain Community Structure)，同时提出了一个节点重要性计算公式，综合了节点的度占比和聚类系数，能够较好地展示节点的重要性。算法的基本思想为：在每次迭代过程中，首先根据节点的重要性降序排序，选择重要性最高的节点弹出并将其加入到样本集，然后禁用节点的直接邻居(这是因为选择的节点是社区中最具代表性的节点，而它的邻居都可以通过该节点一步到达，因此无须选择邻居就可以代表该社区)，然后在剩余节点中进行相同的操作，直到达到要求的样本集大小。如果没有可选的节点且还没有达到终止条件，就重新激活被禁用的节点。重复上述过程，直到得到所需的子集。

综上所述，本文的贡献如下：

1)提出了一个由节点的度和聚类系数组成的节点社区结构重要性计算公式 $imp=\theta * ccf_v+(1-\theta) * d_v$，重要性越高的节点越容易成为社区的中心节点。

2)提出了 GSRNCS 算法，先采样社区中的代表性节点，然后在保持图的社区结构的同时不断采样图中剩下的节点，直至达到要求。

3)在多个真实数据集上进行了多个评价指标的实验，证明了所提算法相比大部分对比算法具有更好的效果。

2 相关工作

本章首先介绍一般性的图采样方法，然后根据文章主题介绍一些保持社区结构的采样算法。

2.1 图采样方法

图采样方法已经被研究人员广泛研究，它的目的是在保持图的某种特征属性的同时简化原始图，然后在得到的规模更小的样本图上进行相关分析，从而以较小的时间和空间代价来实现预期的效果。当前主要的采样方法按照采样策略可以分为基于节点的采样、基于边的采样和基于遍历的采样方法。

2.1.1 基于节点的采样方法

节点采样方法指根据一定的策略选择指定数目的节点，然后仅保留采样节点之间的边，忽略原始图中的其他边。Random Node(RN)，Random Page-rank Node(RPN)和 Random Degree Node(RDN)方法是最经典的节点采样算法。RN 随机选择一组节点，然后保留这组节点之间的边从而生成一个样本图；RPN 将节点选择概率设置为与 PageRank 权重成比例；RDN 则更倾向于高度节点，节点选择概率与度成比例。这些方法都被集成在一个 python 采样库 littleballoffur[1]中。Induced Random Vertex(IRV)[2]方法则是从原始图中等概率地选择指定数目的节点，然后在选择节点之间添加边。

2.2.2 基于边的采样方法

边采样方法指以某种方式选择原始图中的一定数量的边作为样本图的边集，然后将边集中包含的节点加入样本图的节点集。Random Edge(RE)[1]方法随机选择指定数目的边，每条边被选择的概率相同，由于随机性很大，因此采样出来的图比较稀疏，而且该方法倾向于采样高度节点的邻边。(这类方法引用和说明很少，一类方法要 3～6 个引用。)

2.1.3 基于遍历的采样方法

基于遍历的采样方法是通过遍历图中的节点来实现的。Breadth First Sampling(BFS)，Depth First Sampling(DFS)和 Random First Sampling(RFS)[1]是类似的方法，此类方法首先初始化一个队列，然后随机从一个节点开始，每次从队列中弹出一个节点，再将该节点加入样本集，将该节点的邻居节点入队。区别是出队的顺序不同，BFS 选择队头元素出队，DFS 选择队尾元素出队，而 RFS 则随机选择一个节点出队。Snow Ball(SB)[3]方法和 BFS 很相似，SB 采样方法每次循环中只选择固定数量的邻居节点加入队列中，长期以来一直被用于社会研究，即对隐藏人群(如吸毒者)的调查。Forest Fire(FF)[1]采样方法是 SB 的概率版本，SB 方法每轮选择 k 个邻居，而 FF 方法以一定的概率引燃节点的相邻节点，然后邻居节点再以相同的概率引燃它的邻居。

上述方法都是不可回溯的方法，只能单向行走。除了这种不可回溯的方法，还有很多经典的可回溯的方法。Random Walk(RW)[1]采样方法随机从一个节点开始进行随机游走，该方法倾向于采样高度节点，且由于该方法可以游走到曾经路过的节点，因此容易陷入局部稠密区域，容易产生较大的偏差。Random Jump(RJ)[3]方法解决了这个问题，它给出了节点可以随机跳到图中的任何节点的概率，因此其不会陷入局部最优。Metropolalized Random Walk(MRW)[3]方法消除了 RW 采样算法的偏差，从而使目标分布均匀。Multi-Dimensional Random Walk(MDRW)[4]也是为解决偏差问题而提出来的，l 维 MDRW 初始化 L 包含随机 l 个节点，然后重复执行以下步骤直至达到目标数量节点：每步从 L 中以概率(概率和节点的度成正比)选择一个节点 v，然后随机选择 v 的一个邻居加入 L 中，将边 (u,v) 加入边集。最后将 L 作为采样的节点集。该方法被证实能够很好地降低 RW 的偏差。

还有一些方法不属于以上 3 种类型。文献[5]进行了一项调查，证明了当前算法不能有效保留图中稀有的但非常重要的结构，即超级枢纽结构、巨星结构、边缘结构和领带结构，并提出了 Mino-Centric Graph Sampling(MCGS)算法，能够在很好地保留少数结构的同时不丢失多数结构。

2.2 保持社区结构的采样

与 2.1 小节介绍的方法不同，本文方法是保持社区结构的采样方法。当前大多数研究都集中在节点级的属性，比如度分布，而整图级别的属性，如社区结构的研究较少，但却很

重要，因此本节介绍一些社区结构采样的研究。

Community Structure Expansion[6]（CSE）方法从一个随机节点开始，每次从已采样节点的邻居中选择具有最大扩张因子的邻居加入已采样的集合中，直到样本图达到所需规模。文献[7]提出了一种基于图傅里叶变换（FGFT）的图采样方法。此外，Jian 等[8]提出了 SInetL 方法来具体处理 Internet 的拓扑结构。他们使用两个归一化的拉普拉斯谱特征来解析 Internet 拓扑，并通过采样和合并这些图来实现拓扑大小的显著减小。使用该方法可以有效地减少这些图中的节点数，同时保持它们的本质属性。文献[9]提出了一种面向语境结构保存的采样方法，并采用图表示模型来提取语境结构。文献[10]利用以顶点为中心的图模型，提出了一个通用的图采样框架 GraphSDH，同时提出了一种基于顶点度的分层采样方法和一种基于采样位置分析的分层优化方案，大大加快了 PageRank 的计算速度。文献[11]提出了一种保留大规模社区结构的快速代表性子集采样方法 FURS，首先按照度对原始图中的节点降序排序，然后取所有度大于它们的中位数的节点进行采样，每次采样一个度最高的节点并将邻居停用，如果所有节点都被采样或停用但仍未满足要求，那么就将停用的节点重新置为激活节点，重复上述步骤直到满足要求。文献[12]提出了两种 top-leader 采样方法：TLS-e 和 TLS-i。首先初始化 k 个 leader，按照度降序排列所有节点，从度最高的节点开始判断每个节点是否满足两个条件：1）当前节点的邻域与已选择的每个 leader 的邻域交集不能超过一个给定的阈值；2）当前节点的邻域中不能有 k 个及以上节点的度高于该节点。如果满足两个条件则将该节点设为 leader。TLS-e 方法是每个社区等大小扩张，每次选择使得当前社区的邻域增加最少的节点；TLS-i 方法则是优先将内部节点加入社区中，将内部节点加入社区不会引起社区邻居节点的增加。

节点的度虽然一定程度上能表示节点的聚集程度，但是它只能表示有更多的节点与之直接连接，而该节点的邻居节点之间的聚集程度是未知的，仅聚类系数才能更好地表示社区的紧密程度，因此本文采用了综合两种度量方式的策略来计算节点的重要程度。

3　符号定义与重要概念

3.1　符号定义

定义 1（静态无向图）　一个静态无向图表示为 $G=(V,E)$，其中 V 表示节点集，$E\subseteq V\times V$ 表示边集，$N=|V|$ 为节点数量，$M=|E|$ 为边的数量。由于是无向图，因此节点 u 和 v 之间的边表示为 (u,v)。

定义 2（采样图）　一个采样图 $S=(V_S,E_S)$ 为原始图 G 的一个子图，其中 $V_S\subseteq V$，$E_S\subseteq E$。

定义 3（采样率）　采样率为样本节点数和原始节点数的比值，即 $p=|V_S|/|V|$。

定义 4（簇）　一个簇 C 为一个非空节点集的集合，$C=\{C_1,C_2,\cdots,C_k\}$，其中每个 C_i（$i\in[1,k]$）是 V 的子集，称为组。

定义 5（邻居）　S 的邻居为 $N(S)=\{w\in V-V_S\mid \exists v\in S,\text{s.t.}(v,w)\in E\}$。

定义 6（额外邻居）　额外邻居 $EN(v,S)$ 是如果节点 v 被采样到 S 中将会引入的新邻居的集合。$EN(v,S)=N(v)-N(S)\cup S$。

3.2　重要概念

节点重要度：当前的研究大多侧重于依照节点度的大小判定节点是否可以作为社区代表点[11-12]，然而度高的节点只能说明该节点具有较多的直接邻居，而社区结构代表点不仅应有较多的邻居，更重要的是应以该节点为中心形成一个类似包围圈的紧密结构，即中心节点的邻居之间应该有较多的连边，而这种特性可以用聚类系数表示。图 1(a)中节点 1 的度为 8，但是其他节点之间并无任何关联；而图 1(b)中节点 1 的度为 7，其他节点之间关联非常的紧密。虽然图 1(b)中节点 1 的度比图 1(a)中的低，但是显然图 1(b)更像一个社区，且结构非常紧密，然而如果只考虑度而不考虑聚类系数，那么就会认为图 1(a)是更好的社区结构。因此必须考虑节点的聚类系数，但也不能忽略节点的度。故本文提出了一个综合聚类系数和度的节点重要性评估指标，称为节点重要度（imp），节点 v 的重要度表示为：

$$imp_v=\theta\times ccf_v+(1-\theta)\times d_v \tag{1}$$

其中，ccf_v 为节点 v 的聚类系数；d_v 为节点 v 的度。θ 为影响因子，且 $\theta\in[0,1]$，可在值域范围内任意调节，值越大表示越优先考虑聚类系数，值越小表示越优先考虑度，当 θ 为 0 或者 1 时，表示只考虑节点的度或聚类系数。

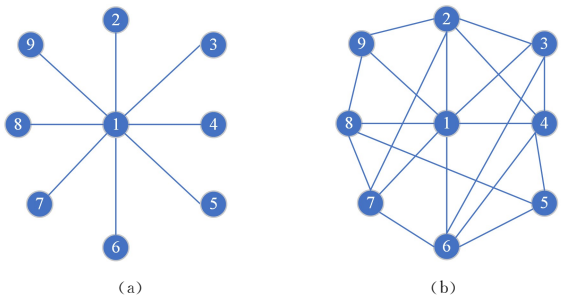


图 1　示例图

Fig. 1　Example diagram

等比例采样扩张：虽然给定了总体节点的采样率，但是每个社区具体采样多少节点才能保证能保留更多更完整的社区结构也是一个问题。真实图中往往有很多个社区，每个社区覆盖的节点数并不一定相同，如果不关心每个社区的采样率，就很容易导致大的社区采样少、小的社区采样多的情况出现，这并不是一个很好的方法。文献[12]提出了等大小社区结构扩张 TLS-e，但这样可能会导致原始图中大的社区被采样的比例较少，而原始图中小的社区被采样的比例较大，对每个社区而言并不均衡。本文选择每个社区等比例的采样扩张策略，由于初始化阶段每个社区结构基本上由一个代表点及其直接邻居组成，因此按照每个社区代表点和直接邻居数量乘以采样率 p 得到每个社区的采样节点数，之后的采样过程就以该数值为每个社区采样节点数的上限。

4　本文提出的算法

4.1　算法描述

本章描述所提算法的原理和细节，算法分为两部分：初始

化社区代表点阶段和等比例社区结构扩张阶段。

初始化社区代表点：首先根据每个节点的度和聚类系数以及给定的参数 θ 计算出节点的重要度，然后对重要度降序排序。由于社区结构的代表节点一般会有较高的度和聚类系数，因此不考虑那些重要度较低的节点，只选择重要度大于中位数的节点作为候选集（选择中位数而不是平均数是因为平均数容易受离群点值的影响而中位数则不会），然后对候选集执行采样过程。算法伪代码见算法 1。

算法 1 初始化社区代表点

输入：节点集 V，原始图 G

输出：节点集 V_S

```
1. 计算节点重要度 imp，M 为 imp 的中位数
2. L=(V,imp(V)), ∀v∈V,imp(v)>M
3. L=sort(V),按照重要度降序排序
4. While L!=∅
5.   #弹出重要度最高的节点
6.   v=L.pop()
7.   V_S=V_S∪v
8.   #移除 v 的所有邻居节点
9.   L.remove(N(v),imp(N(v)))
10. end
11. return V_S
```

每个社区代表点代表社区结构的中心，一般表现为有较高的度和聚类系数，每个社区以代表点为中心开始扩张，不仅总的采样节点数需要达到目标要求，每个社区包含的节点数也应该有一定的限制，如果大的社区和小的社区都包含相同数量的节点，那么大社区将不会有很高的覆盖率，而小社区则会覆盖过多不必要的节点，导致冗余的产生。等大小的社区是不可行的，因为两个社区之间可能相差很多，而按照指定比例采样的方式更能反映出社区的保留程度，因此，本文采取每个社区保留原社区固定比例节点的采样策略，比例设置为和总的采样比例一致。

等比例社区结构扩张：首先初始化每个组包含一个社区代表点，将代表点之外的所有节点置为活跃节点；然后对于每个组，代表点及邻居节点数乘以采样率为每个社区应采样的节点个数，不断从当前组的邻居集合中采样能够引入更少额外邻居的节点（被采样的点多数情况下是代表点的直接邻居，因为社区属于一个比较稠密的区域，且与外界联系稀疏，直接邻居能够引入的额外邻居较少）直到达到该社区的上限或者该社区的邻居集合中没有活跃节点。迭代完毕后，将所有社区的节点取并集，得到采样节点集。由于初始化阶段中并未考虑一些重要度较低的节点，因此采样代表点及邻居数量的总和小于总节点数，故最终采样到的节点数一定小于或等于总采样节点数。迭代完成后，算法将从未被采样的节点中按照重要度降序的顺序采样节点，直到达到采样节点数。算法伪代码见算法 2。

算法 2 等比例社区结构扩张

输入：原始图 G=(V,E)，采样率 p，社区的个数 k，节点集 V_S

输出：样本图 $S=(V_S,E_S)$

```
1. C←∅,act←V−V_S  #act 为活跃节点集
2. for v∈V_S
3.   C_i∈C_i∪v
4. end
5. for C_i∈C do
6.   numnode←(|N(C_i)|+1)×p
7.   while|C_i|<numnode
8.     if act==∅
9.       break
10.      else
11.        v←act, min_{v∈N(C_i)} |EN(v,C_i)|
12.        C_i←C_i∪v
13.        act.pop(v)
14.    end
15.  end
16. end
17. V_S←C_1∪C_2∪…∪C_k
18. if |V_S|<n
19.  按照重要度降序的顺序添加节点到样本集直到|V_S|==n
20. end
21. for e=(u,v)∈E,s.t.,{u,v}∈V_S
22.  E_S=E_S∪{e}
23. end
24. return S
```

4.2 算法复杂度分析

时间复杂度：首先是初始化阶段，需要对节点按照重要度降序排列，排序的时间复杂度为 $O(|V|\times\log(|V|))$；然后是扩张阶段，遍历每个聚类集合，将每个集合添加到指定数目的节点，时间复杂度为 $O(p\times|V|)$。因此总的时间复杂度为 $O(|V|\times\log(|V|))$。

空间复杂度：算法需要维护一个列表用于保存所有节点，空间复杂度为 $O(|V|)$；然后需要保存活跃节点集 act，空间复杂度为 $O(|V-V_S|)$。最终的空间复杂度为 $O(|V|)$。

5 实验设计与分析

本章首先描述实验设置和对比算法，然后列出所使用数据集和评估标准，最后进行实验验证与结果分析。

5.1 实验设置

实验运行在一个 Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20 GHz 的 Ubuntu 服务器上，其中包含 4 个 CPU，每个 CPU 12 核，程序代码用 Python 语言实现。对比算法包括 Forest Fire (FF)[1]，Metropolis Hastings Random Walk (MHRW)[1]，Random Node(RN)[1]，Random Walk(RW)[1]，Community Structure Expansion (CSE)[1]，GraphSDH[10]，TLS-e[12]，TLS-i[12]。

5.2 数据集

使用真实世界的图数据集进行了实验，数据集均为无向无权图，它们可以大致分为两类：大型网络和小型网络。前 5 个数据集是小型网络，包括 Football，Karate，Dolphin，Polblogs 和 Polbooks；后 5 个是规模更大的图，包括 Dblp，Friendster，LiveJournal，Orkut 和 Youtube。数据集具体信息如表 1 所列。

表 1　数据集描述

Table 1　Bataset description

datasets	$\|V\|$	$\|E\|$	Density
Football	115	613	0.094
Karate	34	78	0.139
Dolphin	62	159	0.084
Polblogs	1224	16718	0.022
Polbooks	105	441	0.08100
Dblp	93432	335520	0.00009
Friendster	220015	4031793	0.00017
LiveJournal	84438	1521988	0.00600
Orkut	731514	21992510	0.00008
Youtube	39841	224235	0.00030

5.3　评估标准

本节描述了实验所使用的评估标准，包括节点级别的属性以及整图级别的属性等。

使用 Precision 和 Recall 来评价原始图和样本图之间社区结构的采样质量，这里 precision 和 recall 使用文献[13]中的 δ-precision 和 δ-recall，其中参数设置为 0。使用 Normalized Mutual Information(NMI)[14] 和 Adjusted Rand Index (ARI)[14] 来评估采样图的聚类质量，使用无监督评价指标 Modularity[15] 来评估采样的聚类结果，使用 Fraction of communities(Frac)[11] 来表示采样方法保留的社区的百分比，范围在 0～1 之间。最后使用 Coverage Coverage(Cov)[11] 来表示样本图覆盖的节点比例，定义为采样子图中的节点直接可达的唯一节点总数与图中节点总数的比率，即所采样的节点的邻居节点数与总节点数的比值，表示为 $|\bigcup_{v_i \in V_S} N(v_i)|/|V|$。

5.4　实验结果

本节进行了算法的对比实验及分析。

对于小型网络，采用较大的样本率($p=0.5$)来验证其样本质量。对于大型图，由于它们在规模上是相对庞大的，且在计算上对它们进行聚类比较困难，甚至是不可行的，因此，使用 15% 的抽样率来分析抽样方法的性能。实验结果如表 2、表 3 所列，其中粗体为最佳结果。

大体上看，GSRNCS 算法的实验结果较其他算法来说更好，这是因为该方法是按照聚类系数和度的综合指标来选择节点，选出来的节点更倾向于作为社区的中心节点，且在扩张阶段每次只选择引入最少的邻居节点，这样能够保持引入的大部分都是能够被检测归类为该社区的节点。其他的对比算法中，TLS-i 和 TLS-e 方法表现相对较好，这是因为这两种算法目的也是保持图的社区结构，且 TLS-i 效果会更好一些，因为 TLS-i 能够将更多的内部节点包含到社区中，而 TLS-e 限制了每个社区固定数量的节点，这就导致有一些引入较多邻居节点的节点也被加入到社区中，会在一定程度上影响采样效果。可以观察到，TLS-i 在现实世界的小图形中获得了相对较好的结果。这是因为这些图的规模较小且采样率较高，能够覆盖大多数节点，因此取得的效果较好。然而，对于大型图，要获得令人满意的聚类结果则困难得多。例如，在 Youtube 和 Orkut 网络中，许多检测到的集群不位于任何元数据组中(recall 较低)。

表 2　小型网络实验结果

Table 2　Experiment results of small network

datasets		GSRNCS	MHRW	RN	CSE	RW	FF	GraphSDH	TLS-e	TLS-i
Polbooks	*NMI*	**0.895**	0.739	0.692	0.664	0.499	0.541	0.503	0.526	0.543
	ARI	**0.908**	0.735	0.586	0.479	0.237	0.345	0.370	0.374	0.389
	precision	**0.966**	0.750	0.937	0.820	0.720	0.7992	0.882	0.826	0.909
	recall	**0.426**	0.418	0.360	0.427	0.480	0.397	0.301	0.252	0.332
	frac	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
	cov	0.885	0.961	1.000	0.971	0.857	0.742	0.934	**0.980**	0.885
	modularity	0.574	0.487	0.560	0.446	0.359	0.312	0.601	**0.618**	0.604
Polblogs	*NMI*	0.886	0.921	0.645	0.674	**0.978**	0.894	0.482	0.481	0.548
	ARI	0.942	0.957	0.737	0.637	**0.991**	0.953	0.513	0.434	0.618
	precision	**0.995**	0.881	0.984	0.946	0.994	0.982	0.935	0.905	0.959
	recall	**0.646**	0.141	0.535	0.144	0.145	0.093	0.285	0.257	0.308
	frac	0.454	0.250	0.495	0.416	0.333	0.272	0.439	0.467	**0.502**
	cov	0.741	0.964	0.962	0.901	**0.970**	0.959	0.923	0.783	0.813
	modularity	0.425	0.422	0.428	0.433	0.421	0.373	0.401	**0.454**	0.413
Dolphins	*NMI*	0.086	0.074	0.071	0.069	0.074	0.069	0.091	**0.095**	**0.095**
	ARI	**0.069**	0.054	0.048	0.060	0.065	0.050	0.023	0.050	0.050
	precision	0.927	0.784	0.935	0.800	0.705	0.840	0.951	**1.000**	**1.000**
	recall	0.363	**0.455**	0.369	0.410	0.452	0.386	0.092	0.143	0.095
	frac	**1.000**	**1.000**	**1.000**	0.833	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
	cov	**0.925**	0.903	0.903	0.822	0.887	0.822	0.837	0.837	0.878
	modularity	**0.640**	0.488	0.529	0.454	0.448	0.348	0.299	0.393	0.339
Karate	*NMI*	0.694	**1.000**	**1.000**	0.711	0.808	0.593	0.719	0.753	0.748
	ARI	0.302	**1.000**	**1.000**	0.530	0.777	0.349	0.599	0.625	0.607
	precision	**1.000**	**1.000**	**1.000**	0.833	0.878	0.887	**1.000**	**1.000**	**1.000**
	recall	0.228	0.498	**0.570**	0.398	0.409	0.376	0.319	0.413	0.385
	frac	0.750	0.750	**1.000**	0.750	**1.000**	0.750	0.750	**1.000**	**1.000**
	cov	0.705	0.941	**1.000**	0.941	**1.000**	0.941	0.941	**1.000**	**1.000**
	modularity	0.455	0.379	0.398	0.363	0.322	0.268	0.425	0.442	0.443
Football	*NMI*	**1.000**	0.934	0.867	0.890	0.841	0.940	0.831	**1.000**	**1.000**
	ARI	**1.000**	0.886	0.631	0.784	0.715	0.892	0.765	**1.000**	**1.000**
	precision	0.888	0.878	0.776	0.868	0.848	0.903	0.828	**1.000**	**1.000**
	recall	0.510	0.503	0.510	0.438	**0.5146**	0.498	0.412	0.482	0.467
	frac	**1.000**	0.900	**1.000**	**1.000**	0.9000	0.900	0.900	**1.000**	**1.000**
	cov	0.886	0.947	**1.000**	0.991	0.947	0.956	0.878	0.903	0.946
	modularity	**0.679**	0.567	0.628	0.598	0.634	0.551	0.508	0.631	0.640

表 3 大型网络实验结果

Table 3 Experiment results of large network

datasets		GSRNCS	MHRW	RN	CSE	RW	FF	GraphSDH	TLS-e	TLS-i
Youtube	*NMI*	**0.862**	0.737	0.602	0.599	0.740	0.697	0.791	0.786	0.834
	ARI	**0.078**	0.068	0.065	0.058	0.070	0.074	0.047	0.039	0.053
	precision	**0.995**	0.859	0.983	0.741	0.891	0.882	0.643	0.637	0.800
	recall	0.094	0.095	0.093	0.087	0.097	0.087	0.067	**0.119**	0.102
	frac	0.046	0.041	0.039	**0.078**	0.038	0.044	0.040	0.043	0.045
	cov	0.754	0.665	0.559	0.584	0.722	0.736	0.745	0.752	**0.773**
	modularity	**0.723**	0.643	0.717	0.621	0.560	0.525	0.401	0.556	0.453
Dblp	*NMI*	0.722	0.632	0.712	0.526	0.642	0.604	0.753	0.746	**0.783**
	ARI	0.009	0.023	**0.036**	0.023	0.036	0.038	0.025	0.008	0.034
	precision	**0.989**	0.816	0.889	0.864	0.859	0.790	0.836	0.916	0.864
	recall	**0.154**	0.026	0.098	0.016	0.023	0.021	0.115	0.112	0.125
	frac	**0.921**	0.255	0.791	0.264	0.255	0.264	0.832	0.892	0.903
	cov	0.593	0.393	**0.638**	0.448	0.449	0.455	0.593	0.603	0.623
	modularity	**0.995**	0.925	0.988	0.870	0.877	0.843	0.734	0.650	0.881
Orkut	*NMI*	0.802	0.715	0.481	0.723	0.761	0.729	0.765	0.826	**0.843**
	ARI	**0.782**	0.691	0.276	0.678	0.742	0.680	0.097	0.126	0.219
	precision	**0.972**	0.730	0.969	0.812	0.766	0.819	0.614	0.695	0.741
	recall	0.099	0.085	0.094	0.082	0.097	0.091	0.093	0.075	**0.129**
	frac	**0.843**	0.620	0.827	0.593	0.620	0.586	0.728	0.823	0.836
	cov	**0.923**	0.819	0.775	0.902	0.858	0.903	0.901	0.913	0.918
	modularity	0.763	0.672	0.705	0.673	0.683	0.696	0.667	0.707	**0.791**
Livejournal	*NMI*	0.961	0.827	0.877	0.854	0.842	0.858	0.953	0.963	**0.982**
	ARI	0.924	0.378	0.386	0.678	0.692	0.693	0.883	0.837	**0.941**
	precision	**0.967**	0.933	0.946	0.963	0.942	0.957	0.912	0.952	0.948
	recall	**0.148**	0.138	0.132	0.020	0.019	0.021	**0.102**	0.074	0.097
	frac	**0.102**	0.072	0.073	0.098	0.096	0.097	0.092	0.101	0.093
	cov	**0.951**	0.738	0.860	0.334	0.328	0.336	0.342	0.724	0.924
	modularity	0.903	0.937	**0.981**	0.874	0.867	0.868	0.897	0.900	0.960
Friendster	*NMI*	0.932	0.721	0.721	0.760	0.772	0.789	0.885	0.922	**0.947**
	ARI	**0.642**	0.372	0.383	0.481	0.546	0.533	0.412	0.258	0.606
	precision	**0.946**	0.926	0.919	0.731	0.792	0.767	0.882	0.910	0.932
	recall	**0.175**	0.143	0.128	0.12	0.123	0.135	0.114	0.125	0.152
	frac	0.734	0.682	0.702	0.738	0.722	0.622	0.728	0.743	**0.762**
	cov	0.884	0.913	0.902	0.769	0.583	0.589	0.902	0.892	**0.921**
	modularity	**0.939**	0.891	0.908	0.882	0.876	0.878	0.543	0.766	0.904

5.5 采样率的影响

为探究不同采样率对采样结果的影响，在 Dblp 数据集上进行了实验，采样率范围在 0.05～1 之间，增长幅度为 0.05。实验结果如图 2 所示。从图中可以看出，Precision 和 NMI 较为平坦，而 Recall 和 ARI 随着采样率的增加而增长，这意味着随着采样率的增加，得到的样本图的社区能够更好地覆盖原始图的社区，这也是符合常识的，即采样的节点越多，覆盖的社区数更多的可能性越大。

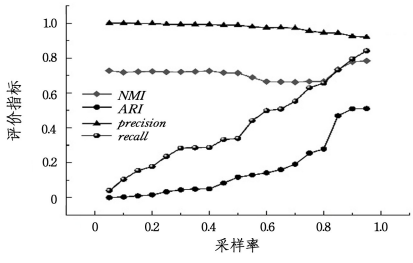


图 2 不同采样率对实验结果的影响

Fig. 2 Effect of different sampling rates on experimental results

5.6 参数 θ 的影响

为了观察参数 θ 的不同值对采样结果的影响，针对 Dblp 数据集进行了实验，采样率范围在 0～1 之间，增长幅度为 0.05，实验结果如图 3－图 6 所示。从图中可以看出，随着参数 θ 的增加，各评价指标基本都呈上升趋势，这意味着随着聚类系数占比的增加，采样效果会越来越好，说明将聚类系数参与计算重要度这一策略是可行的。

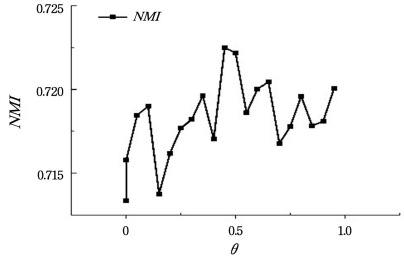


图 3 θ 不同值对 NMI 的影响

Fig. 3 Effect of different values of θ on NMI

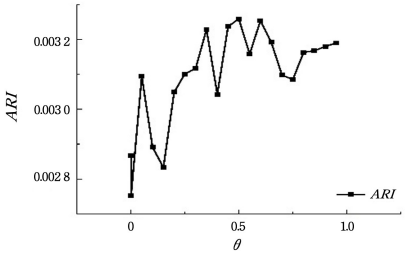


图 4 θ 不同值对 ARI 的影响

Fig. 4 Effect of different values of θ on ARI

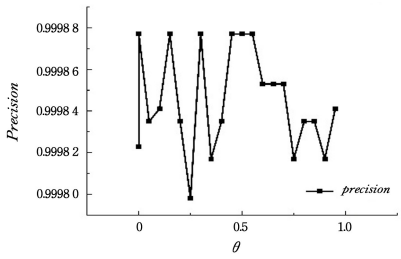


图 5　θ 不同值对 precision 的影响

Fig. 5　Effect of different values of θ on precision

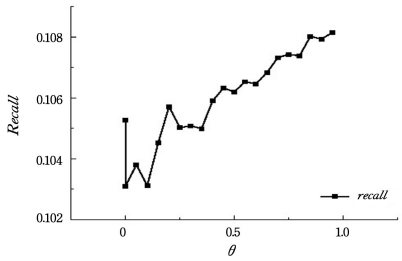


图 6　θ 不同值对 recall 的影响

Fig. 6　Effect of different values of θ on recall

结束语　本文提出了一种保持社区结构的图采样方法，用于在保持原始图社区结构的同时简化原始图。经过了理论分析和实验验证，证明了所提算法能够很好地保持原始图的社区结构，并具有比对比方法更好的效果。然而，本文提出的算法只适合于静态同质图，在未来的工作中，可以研究在同时保留多个属性的情况下对大规模图进行采样，以及对动态图、异质图和加权图等更复杂的图进行采样。

参 考 文 献

[1] ROZEMBERCZKI B, KISS O, SARKAR R. Little Ball of Fur: A Python Library for Graph Sampling[C]// Proceedings of the 29th ACM International Conference on Information & Knowledge Management. Virtual Event Ireland: Association for Computing Machinery, 2020: 3133-3140.

[2] AHMED N, NEVILLE J, KOMPELLA R. Network Sampling: From Static to Streaming Graphs [J]. ACM Transactions on Knowledge Discovery from Data, 2014, 8(2): 1-56.

[3] LESKOVEC U, FALOUTSOS C. Sampling from large graphs [C]// Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: Association for Computing Machinery, 2006: 631-636.

[4] RIBERIO B, TOWSLEY D. Estimating and sampling graphs with multidimensional random walks [C]// Internet Measurement Conference. New York: Association for Computing Machinery, 2010: 390-403.

[5] ZHAO Y, JIANG H J, CHEN Q A, et al. Preserving Minority Structures in Graph Sampling [J]. IEEE Transactions on Visualization and Computer Graphics, 2021, 27(2): 1698-1708.

[6] MAIYA A S, BERGER-WOLF T Y. Sampling community structure[C]// Proceedings of the 19th International World Wide Web Conference. New York: Association for Computing Machinery, 2010: 701-710.

[7] WANG F, CHEUNG G, WANG Y C. Low-complexity Graph Sampling With Noise and Signal Reconstruction via Neumann Series[J]. IEEE Transactions on Signal Processing, 2019, 67(21): 5511-5526.

[8] JIAN B, SHI J M, ZHANG W S, et al. Graph sampling for Internet topologies using normalized Laplacian spectral features[J]. Information Sciences, 2019, 481: 574-603.

[9] ZHOU Z G, SHI C, SHEN X L, et al. Context-aware Sampling of Large Networks via Graph Representation Learning[J]. IEEE Transactions on Visualization and Computer Graphics, 2021, 27(2): 1709-1719.

[10] HU J B, DAI G H, WANG Y, et al. GraphSDH: A General Graph Sampling Framework with Distribution and Hierarchy [C]// 2020 IEEE High Performance Extreme Computing Conference (HPEC). Waltham: IEEE, 2020: 1-7.

[11] MALL R, LANGONE R, SUYKENS J A K. FURS: Fast and Unique Representative Subset selection retaining large-scale community structure[J]. Social Network Analysis and Mining, 2013, 3(4): 1075-1095.

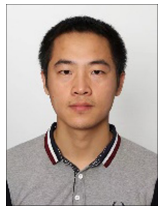
[12] ZHANG J P, CHEN H C, YU D J, et al. Cluster-preserving sampling algorithm for large-scale graphs[J]. Science China Information Sciences, 2022, 66: 112103.

[13] ZHANG J P, PEI Y L, GEORGE F, et al. Evaluation of the Sample Clustering Process on Graphs[J]. IEEE Transactions on Knowledge and Data Engineering, 2020, 32(7): 1333-1347.

[14] SANTO F, DARTO H. Community detection in networks: A user guide[J]. Physics Reports, 2016, 659: 1-44.

[15] MARK E J N. Modularity and community structure in networks [J]. Proceedings of the National Academy of Sciences, 2006, 103(23): 8577-8582.

HONG Yu, born in 1998, master. His main research interest is graph data mining.



ZHANG Jianpeng, born in 1988, Ph.D, research associate. His main research interest is big data analysis.

（责任编辑：何杨）