SNS:一种快速无偏的分层图抽样算法

朱君鹏 李 晖 陈 梅 戴震宇
(贵州大学计算机科学与技术学院 贵阳 550025)
(贵州省先进计算与医疗信息服务工程实验室 贵阳 550025)

摘 要 抽样作为一种有效的统计分析方法,常被用于大规模图数据分析领域以提升性能。现有的图抽样算法大多存在高度节点或低度节点过度入样的问题,较大程度地影响了算法的性能。复杂网络具有无标度特性,即节点的度服从幂律分布,节点个体之间存在较大差异。在基于点选择策略的抽样方法的基础上,通过结合节点的近似度分布策略,设计并实现了高效无偏的分层图抽样算法 SNS。在 3 个真实的图数据集上的实验结果表明,SNS 算法比其他图抽样算法保留了更多的拓扑属性,且执行效率比 FFS 更高。SNS 算法在度的无偏性、抽样结果拓扑属性近似性方面的表现均优于现有算法。

关键词 有偏抽样,分层抽样,图抽样,向量聚类,性能评估
中图法分类号 TP393 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2018.11.039

SNS:A Fast and Unbiased Stratified Graph Sampling Algorithm

ZHU Jun-peng LI Hui CHEN Mei DAI Zhen-yu
(College of Computer Science and Technology,Guizhou University,Guiyang 550025,China)
(Guizhou Engineering Laboratory of Advance Computing and Medical Information Service,Guiyang 550025,China)

Abstract As an effective method of statistical analysis,sampling is commonly used in the field of analyzing the large-scale graph data to improve the performance. However,most of the existing graph sampling algorithms often have the problem of excessive sampling of high and low nodes,resulting in lower accuracy derived from the scale-free characteristic of complex networks. The scale-free characteristic means the degrees of different nodes follow a power law distribution,and the difference between nodes is huge. On the basis of the sampling method on node selection strategy,combining the approximate degree distribution strategy of nodes,this paper proposed and realized an efficient and unbiased stratified graph sampling algorithm named SNS. The experimental results show that SNS algorithm preserves more topological properties on three real data sets than other graph sampling algorithms,and consumes less time than FFS algorithm. Therefore,SNS algorithm is superior to the existing algorithms in terms of the unbiasedness of degree and the accuracy of sampling results.

Keywords Biased sampling,Stratified sampling,Graph sampling,Vector clustering,Performance estimation

1 引言

现有的图抽样算法可以分为以下 3 类:基于点选择策略的随机抽样算法、基于边选择策略的随机抽样算法、基于图拓扑结构的随机抽样算法。其中,典型的代表算法有 NS[1],ES[1],ES-i[2],FFS[1]等。复杂网络具有两个基本特性:小世界[3]和无标度[4]。小世界特性是指图具有较小的直径和较大的聚类系数;无标度特性是指不同节点的度服从幂律分布,节点个体之间存在较大的差异。复杂网络的上述特征导致了现有的图抽样算法普遍存在高度节点或低度节点过度入样的问题,文献[1-2,18-26]对此进行了探讨。

如何保证高度节点和低度节点在一次抽样过程中被"公平"抽到,是图分析领域的研究热点。为解决此问题,本文在研究图抽样算法的过程中考虑了节点的度分布,因此快速且高效地获取节点的度分布结果是本文的重点研究工作之一。通过对大量向量聚类算法(如 k-Means[6],EM[7],k-Medoids[8],DBSCAN[9],HDBSCAN[10])的实验分析发现,使用 k-Means 向量聚类算法对节点的度进行聚类能够更快、更好地得到节点的近似度分布结果,因此本文在基于 NS 算法设计的图抽样算法 SNS 中引入了 k-Means 聚类算法。此外,SNS 算法还引入了根据节点的近似度分布对节点分层的机制,结合 NS 算法,提升了抽样性能。

到稿日期:2017-10-08 返修日期:2017-12-28 本文受国家自然科学基金项目(61562010, 61462012,U1531246),贵州省重大应用基础研究项目(JZ20142001),贵州省数据分析云服务创新团队(黔科合人才团队字[2015]53),贵州大学研究生创新基金项目(研理工 2017078)资助。
朱君鹏(1989-),男,硕士生,主要研究方向为大规模数据管理与分析,E-mail:jpzhu.gm@gmail.com;李 晖(1982-),男,博士,硕士生导师,主要研究方向为大规模数据管理与分析,E-mail:cse.HuiLi@gzu.edu.cn(通信作者);陈 梅(1964-),女,硕士生导师,主要研究方向为数据库及其应用系统。

本文第 2 节简要介绍了经典的图抽样算法，并说明了存在的问题；第 3 节介绍了实验部分采用的评估图抽样算法的参数及其计算方法；第 4 节介绍了本文设计的 SNS 算法，并分析了其正确性和时空复杂度；第 5 节介绍了实验及其分析结果；最后对全文进行了总结与展望。

2 图抽样算法介绍

2.1 基于点选择策略的随机抽样算法

NS 算法是该类抽样算法的典型代表，其实现过程如算法 1 所示。首先，NS 算法随机、均匀地选择一个点（第 3－4 行），并将其放入样本点集 V_s 中；然后，迭代上述过程（第 2－4 行），直至点的数量达到抽样规模为止；最后，根据导出子图概念（第 5 行），把所有的源点、目标点在 V_s 中的边全部抽取出来并放入样本边集 E_s 中，得到抽样结果 $S=(V_s,E_s)$。该算法简单、直观，但抽样时不能很好地保留图的大量拓扑属性。其主要原因是：复杂网络具有无标度特征，使得 NS 算法在抽样时易于抽取更多的低度节点，算法误差较大。

算法 1 NS 算法

输入：抽样比例 φ，点集合 N，边集合 E
输出：抽样结果集 S=(Vs,Es)
1. Vs=null set,Es=null set
2. While |Vs|<|N| * φ
3. Generate random uniformly number β
4. Get v_β→Vs
5. Get Es(sourceNode,targetNode) and sourceNode∈Vs and targetNode∈Vs
6. return S=(Vs,Es)

2.2 基于边选择策略的随机抽样算法

ES 算法和 ES-i 算法是该类抽样算法的典型代表，其中 ES-i 算法是 ES 算法的一种改进算法。这两种算法的介绍分别如下。

2.2.1 ES 算法

ES 算法的实现过程如算法 2 所示。其核心处理流程是：首先，随机、均匀地选择一条边（第 3－4 行），并将该边加入到样本边集 E_s 中；然后，将选出的边的两个端点加入到样本点集 V_s 中（第 5 行），迭代上述过程直到点的数量达到抽样规模为止（第 2－5 行）；最后，得到抽样结果 $S=(V_s,E_s)$。该算法在抽样时易于抽取高度节点，因为高度节点对应边的数量较大。此外，由第 5 节中的实验结果可知，ES 算法的抽样性能较差。

算法 2 ES 算法

输入：抽样比例 φ，点集合 N，边集合 E
输出：抽样结果集 S=(Vs,Es)
1. Vs= null set, Es=null set
2. While |Vs|<|N| * φ
3. Generate random uniformly number β
4. Get e_β→Es
5. Put two endpoints of e_β into Vs
6. return S=(Vs,Es)

2.2.2 ES-i 算法

ES-i 算法是 ES 算法的一种改进算法，其实现过程如算法 3 所示。ES-i 算法在 ES 算法的基础上（第 1－5 行）增加了导出子图的步骤（第 6 行）。由第 5 节中的实验结果可知，该算法的抽样性能远优于 ES 算法，但 ES-i 算法同样存在高度节点过度入样的问题。

算法 3 ES-i 算法

输入：抽样比例 φ，点集合 N，边集合 E
输出：抽样结果集 S=(Vs,Es)
1. Vs=null set,Es=null set
2. While |Vs|<|N| * φ
3. Generate random uniformly number β
4. Get e_β→Es
5. Put two endpoints of e_β into Vs
6. Get Es(sourceNode,targetNode) and sourceNode∈Vs and targetNode∈Vs
7. return S=(Vs,Es)

2.3 基于图拓扑结构的随机抽样算法

FFS 算法是该类抽样算法的典型代表，其实现过程如算法 4 所示。首先，FFS 算法随机、均匀地选择一个种子点（第 3－7 行），该种子点可能从队列中（第 4 行）顺序选择，亦可能通过随机选择（第 6－7 行）获取；随后，采用图的广度优先遍历技术获取该种子点的所有邻居点（第 9 行）；接着，生成一个服从几何分布的随机数（第 10 行），该随机数决定了加入到样本中的邻居点和相应边的数量（第 11 行）；然后，将选中的点作为种子，重复上述步骤（第 2－11 行），直到点的数量满足抽样规模为止；最后，通过导出子图（第 12 行）得到抽样结果集 $S=(V_s,E_s)$。大量实验证明，在 FFS 算法中，选择邻居点数量的最优经验值为 2.33 边/点[2]。FFS 虽然被认为是当前最优秀的抽样算法之一，但是仍然存在高度节点过度入样的问题[21]。

算法 4 FFS 算法

输入：抽样比例 φ，点集合 N，边集合 E
输出：抽样结果集 S=(Vs,Es)
1. Vs= null set,Es=null set
2. While |Vs|<|N| * φ
3. if Queue.isNotNull
4. Queue.get→v_β
5. else
6. Generate random number β
7. Get v_β→Vs
8. endif
9. BFS(v_β)→$V_{neighbor}$
10. Generate a geometric distribution number γ
11. Uniformly random select nodes by γ from $V_{neighbor}$ and put into Queue,and put nodes and corresponding edges into S=(Vs,Es)
12. Get Es(sourceNode,targetNode) and sourceNode∈Vs and targetNode∈Vs
13. return S=(Vs,Es)

3 图的基本定义与抽样结果的评价指标

3.1 图的基本定义

根据定义，将图表示为 $G=(V,E)$，其中节点集合 $V=\{v_1,v_2,v_3,\cdots,v_n\}$，边集合 $E=\{e_1,e_2,e_3,\cdots,e_m\}$，节点的度集合 $D=\{d_1,d_2,d_3,\cdots,d_n\}$，$|N|$ 表示节点集合中节点的数量，$|E|$ 表示边集合中边的数量。

用 $S=f(G)=(V_s,E_s)$ 来表示抽样结果集，且 $S_{nodes}=V_s\subset V$，$S_{edges}=E_s\subset E$。本文使用 φ 表示抽样比例，且 φ 满足 $0\leqslant\varphi\leqslant1$。大多数图抽样算法通过度量节点的数量来统计抽样是否达到要求，表示为 $|S|=\varphi*|N|$；也有少数算法通过度量

边的数量来统计抽样是否达到要求，表示为 $|S|=\varphi*|E|$。本文采用基于点的度量标准，即使用 $|S|=\varphi*|N|$ 表示抽样规模。

本文使用 $\theta(\cdot)$ 表示图的拓扑属性值，它既可以用来表示单值属性（如图的平均度、平均密度等）的取值，也可以用来表示分布属性（如节点的度分布、聚类系数分布等）的取值。下面介绍抽样结果的常见评价指标及其计算方法。

3.2　抽样结果的评价指标

抽样的目标是得到一个能够更好地保留原图拓扑属性的子图。因此，评价抽样结果时需要考查子图的拓扑属性与原图的近似程度，即需要满足：

$$\theta(G)\approx\theta(S) \tag{1}$$

其中，$\theta(S)$ 表示抽样结果图的拓扑属性取值，$\theta(G)$ 表示原始图的拓扑属性取值。常见的图的拓扑属性以及属性值的计算方法如下。

(1)平均度

$$deg_{avg}=\frac{1}{|N|}\sum_{v\in V}deg(G) \tag{2}$$

其中，$\sum_{v\in V}deg(G)$ 表示图中出现的所有节点的度。

(2)密度

$$density=\frac{link_{real}}{link_{possible}} \tag{3}$$

其中，$link_{real}$ 表示图中实际存在的边的数量，$link_{possible}$ 表示图中可能存在的边的最大数量。对于有向图，有：

$$link_{possible}=|N|*(|N|-1) \tag{4}$$

对于无向图，有：

$$link_{possible}=\frac{|N|*(|N|-1)}{2} \tag{5}$$

由上述定义可知，当一幅图中的节点数量一定时，有 $density\propto link_{real}$。

(3)直径

$$diameter=\max_{v_i,v_j}d(v_i,v_j) \tag{6}$$

图的直径即图中最短路径的最大值。

(4)平均聚类系数

$$C(v_i)=\frac{2N_{v_i}}{K_{v_i}*(K_{v_i}-1)} \tag{7}$$

其中，k_{v_i} 表示节点 v_i 的度，N_{v_i} 表示所有邻居点的数量，$C(v_i)$ 表示点 v_i 的聚类系数。因此，平均聚类系数可以定义为：

$$C_{avg}=\frac{\sum_{i=1}^{|N|}C(v_i)}{|N|} \tag{8}$$

(5)节点的度分布

图的度分布定义为度为 k 的节点的比例，因此，

$$p_k=\frac{|\{v\in V|deg(v)=k\}|}{|N|} \tag{9}$$

(6)聚类系数分布

$$p_c=\frac{|\{v\in V'|C(v)=c\}|}{|V'|} \tag{10}$$

其中，$V'=\{v\in V|deg(v)>1\}$，某个点的聚类系数用来表示以节点 v 为中心的三角形的数量。在图中，节点倾向于聚成簇，因此聚类系数是非常重要的度量指标。

4　SNS 算法

SNS 算法是在 NS 算法的基础上，结合节点的近似度分布策略优化得到的快速无偏的分层图抽样算法，其实现过程如算法 5 所示。该算法包含 4 个阶段：在第 1 阶段（第 1－2 行），使用 k-Means 算法聚类节点的度，得到节点的近似度分布，从第 2 行可知，聚类结果的簇数目设置为 3（详见第 5 节）；在第 2 阶段（第 3 行），根据节点的近似度分布对节点进行分层，得到 $N_{high\text{-}degree}$，$N_{medium\text{-}degree}$ 和 $N_{low\text{-}degree}$ 3 层；在第 3 阶段（第 4－7 行），使用 NS 算法分别对 $N_{high\text{-}degree}$，$N_{medium\text{-}degree}$ 和 $N_{low\text{-}degree}$ 进行抽样，得到样本点集 V_s；在第 4 阶段（第 8 行），采用导出子图概念得到抽样结果集中边的集合，最终得到抽样子集 $S=(V_s,E_s)$。

算法 5　SNS 算法

输入：抽样比例 φ，点集合 N，边集合 E

输出：抽样结果集 $S=(V_s,E_s)$

1. V_s = null set，E_s = null set
2. k-Means = k-Means(distance,3)
3. $(N_{high\text{-}degree},N_{medium\text{-}degree},N_{low\text{-}degree})$ = k-Means.run(D)
4. $V_{high\text{-}degree}=NS(\varphi,N_{high\text{-}degree})$
5. $V_{medium\text{-}degree}=NS(\varphi,N_{medium\text{-}degree})$
6. $V_{low\text{-}degree}=NS(\varphi,N_{low\text{-}degree})$
7. $V_s=V_{high\text{-}degree}\cup V_{medium\text{-}degree}\cup V_{low\text{-}degree}$
8. Get E_s(sourceNode,targetNode) and sourceNode$\in V_s$ and targetNode$\in V_s$
9. return $S=(V_s,E_s)$

SNS 算法基于点选择抽样策略的主要原因在于点选择策略的思路简洁、易扩展，并且具有较强的理论基础性，但该类算法的抽样效果差，结合分层策略可使性能得到显著提升。本文提出的近似度分布策略能够将不同层内抽样规模作为参数，合理地分配不同度的点的入样比例，从而解决基于边选择和基于图拓扑结构的抽样策略中的高度节点过度入样问题。这部分工作的理论分析和实验验证将在后续工作中进行探讨。

SNS 算法的创新之处在于该算法的设计过程考虑了节点的度分布对抽样性能的影响。复杂网络的无标度特性会导致节点个体之间的差异较大，从统计学的角度看，采用分层抽样的思想能够有效地降低抽样误差。因此，如何分层成为了 SNS 算法的核心。根据节点的度分布对节点进行分层能够从根本上解决节点个体之间存在的差异问题，但得到节点的度分布或者近似度分布，使算法具有强的可操作性，是算法设计的难点。大量实验表明，向量聚类技术能够解决上述问题。通过分析大量聚类算法发现，k-Means 算法简单高效且时间复杂度（线性时间复杂度）低，得到的近似度分布结果的层内差异小，层间差异大。

4.1　算法的正确性分析

假设原始图有 N 个节点，使用 k-Means 算法将节点分成 i 层，这 i 层包含的节点个数分别为 $|N_1|,|N_2|,\cdots,|N_i|$，且满足：

$$\begin{aligned}&\forall N_p\cap\forall N_q=\text{null set},p,q\in[1,i]\\&p\neq q,N_1\cup N_2\cup\cdots\cup N_i=N\end{aligned} \tag{11}$$

根据式(11)可知，任意两层之间的交集为空，所有 i 层的并集为原始图，因此，必然存在：

$$|N|=(|N_1|+|N_2|+\cdots+|N_i|) \tag{12}$$

所以有：

$$|N| * \varphi = (|N_1| + |N_2| + \cdots + |N_i|) * \varphi \quad (13)$$

$$= |N_1| * \varphi + |N_2| * \varphi + \cdots + |N_i| * \varphi \quad (14)$$

其中，$|N| * \varphi$ 为 NS 算法的抽样结果。式(14)为 SNS 算法的抽样结果，可见 NS 算法和 SNS 算法的结果具有等价性，从而证明了 SNS 算法运行结果的正确性。

4.2 算法的复杂性分析

SNS 算法在 NS 算法的基础上增加了 k-Means 向量聚类步骤，因此其时间复杂度变为 $O(kt|N|)+O(|N|+|E|)$。其中，k 表示聚类结果数，t 表示迭代次数，$|N|$ 表示节点的个数，$|E|$ 表示边的个数，并且存在 $k \ll |N|$，$t \ll |N|$。可以得出，SNS 算法的渐进时间复杂度没有发生变化，SNS 算法的空间复杂度仍为 $O(|N|+|E|)$。

5 实验分析

本节将使用 3 个代表性的图数据集来评估 SNS，NS，ES，ES-i，FFS 算法的抽样性能，并比较 SNS 算法和 FFS 算法的运行时间。

5.1 数据集和软硬件环境配置

实验中使用的 3 个数据集[11]分别为 Facebook 朋友列表网络、Condmat 合作者网络、Amazon 消费者网络。表 1 汇总了 3 个网络图数据集的全部统计参数。3 个数据集的数据规模(点数量、边数量)依次增大，网络依次变得更稀疏。每个数据集的抽样区间均为[5%，25%]。对于每种算法、每个抽样比例，实验重复进行了至少 200 次，以确保结果的准确性。

表 1 网络图数据集的特征参数

Table 1 Feature parameters of network graph dataset

Graph	Nodes	Edges	Clustering coefficient	Diameter	Average density	Average degree
Facebook	4039	88234	0.6055	8	1.08×10^{-2}	46.6910
Condmat	23133	186936	0.6334	15	7×10^{-4}	16.1618
Amazon	334863	925872	0.2050	47	1.65×10^{-5}	5.5299

实验中，设置 k-Means 算法的聚类结果簇数目 k 为 3。在 SNS 算法设计之初，设置 k 值为 2，这意味着只需将高度节点和低度节点分开即可，但是经过大量实验表明，当 k 为 2 时，高度节点分离得不彻底，即不能更加准确地得到高度节点集合 $N_{high\text{-}degree}$。设置 k 值为 3 后，不同节点的度分布结果表明，层内节点的度差异较小，层间节点的度差异较大(轮廓系数大)，近似度分布结果的准确度得到了极大提升，因此可以有效分离出高度节点。经过大量实验分析得出，k 值继续增大，只会导致低度节点集合反复分割，这显然是没有必要的，因此最终将 k 值设置为 3。

表 2 汇总了实验使用的软硬件环境。为了保证算法的可扩展性，本文将 3 个数据集的数据全部存储在数据库中，按照文献[15]的方式对数据进行统一处理和存储，现有的大多数图分析引擎[16-17]都采用了这种存储方式。

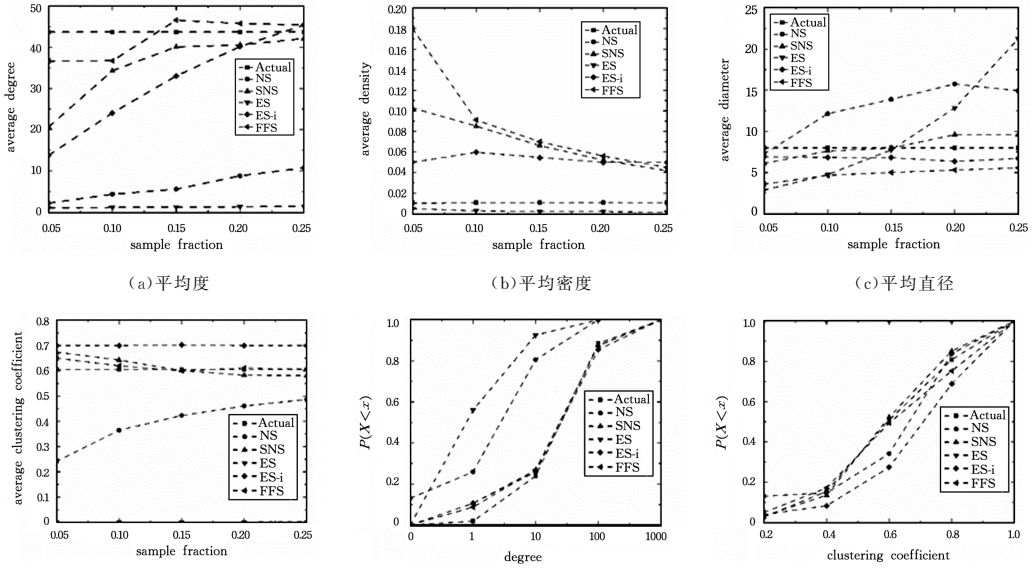
表 2 软硬件环境配置

Table 2 Configuration of software and hardware

CPU	Inter(R)-Xeon-CPU-E5-2630@2.36GHz 2 cores
内存	1GB
数据库	PostgreSQL 9.5.2

5.2 度评估

本节通过实验比较了未抽样与使用 SNS，NS，ES，ES-i，FFS 算法进行抽样时的度指标接近程度。图 1(a)、图 2(a)、图 3(a)分别呈现了不同抽样比例下不同数据集的平均度属性评估结果，图 1(e)、图 2(e)、图 3(e)分别呈现了 15%[1)]抽样规模下不同数据集的度分布情况。



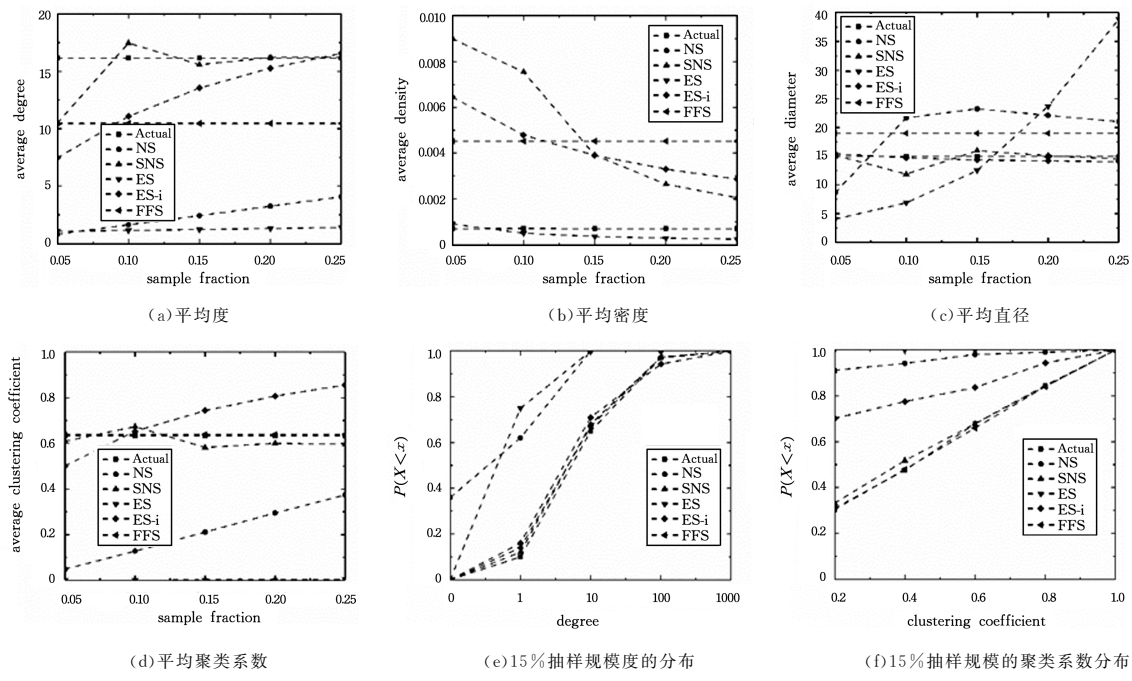
(a)平均度 (b)平均密度 (c)平均直径

(d)平均聚类系数 (e)15%抽样规模的度分布 (f)15%抽样规模的聚类系数分布

图 1 Facebook 网络图的参数评估

Fig. 1 Parameter estimation of Facebook

1) 文献[1]中指出 15%是抽样比例的最优经验值

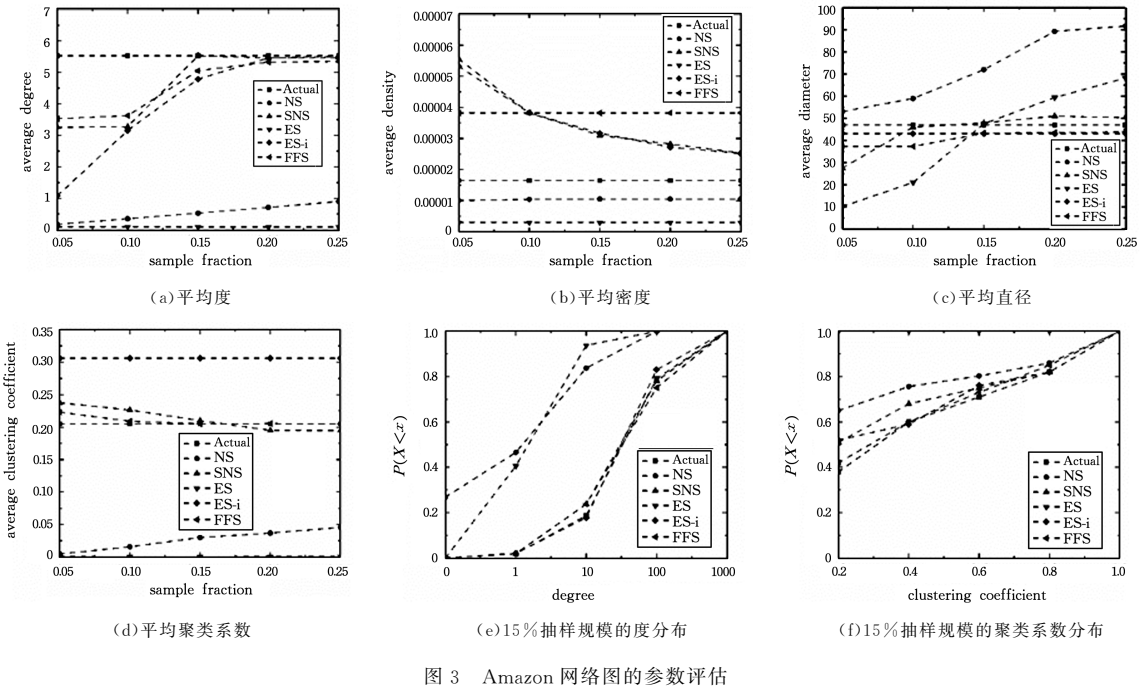


(a)平均度　(b)平均密度　(c)平均直径

(d)平均聚类系数　(e)15%抽样规模度的分布　(f)15%抽样规模的聚类系数分布

图 2　Condmat 网络图的参数评估

Fig. 2　Parameter estimation of Condmat



(a)平均度　(b)平均密度　(c)平均直径

(d)平均聚类系数　(e)15%抽样规模的度分布　(f)15%抽样规模的聚类系数分布

图 3　Amazon 网络图的参数评估

Fig. 3　Parameter estimation of Amazon

可以看到，在平均度的实验结果中，SNS 算法除了在 Facebook 数据集上次于 FFS 算法位居第二之外，在其他两个数据集上都优于 FFS 算法，并且相对于其他几种算法具有更大的优势。这主要是由于 SNS 算法采用了分层思想，能够更加“公平”地抽取高度节点和低度节点，并且分层抽样本身能够使抽样结果的误差更小。其次，在 Facebook 数据集上，SNS 算法略次于 FFS 算法位居第二，主要是由于 Facebook 数据集的数据量较小，$N_{high\text{-}degree}$ 中仅有 4 个节点，在[5%，25%]的抽样规模下，SNS 算法在实现时保证了至少能抽取到一个高度节点，而 FFS 算法本身偏向于抽取更多的高度节点，因此，当抽样比例小于 15%时，FFS 算法略优于 SNS 算法。同时可以看到，图 1(a)中随着抽样规模的增大(大于 15%)，SNS 算法与 FFS 算法的性能都与未抽样时趋于接近，而 FFS 算法因为抽取了更多的高度节点导致平均度略高于未抽样时的值。

从度分布实验结果中可以看到，SNS 算法在[100，1000]区间，即在高度节点区间上，抽取的点比 NS 算法多。这说明 SNS 算法很好地解决了 NS 算法存在的低度节点过度入样的问题。

NS 算法和 ES 算法的性能最差。对于 NS 算法而言，其在

抽样过程中存在低度节点过度入样的问题，导致实验结果差。ES 算法采用随机、均匀地抽取边的策略，丝毫没有考虑节点度之间存在的差异性，因此抽样误差最大，度指标的实验结果最差。ES-i 由于采用了导出子图概念，使得其抽样性能优于 ES 算法。

FFS 算法基于图的广度优先遍历策略(BFS)。文献[21]证明了 BFS 算法的抽样结果的准确度为 85%～95%，并且证明了 BFS 算法存在高度节点过度入样的问题。但是 FFS 算法没有将某个种子点的所有邻居点全部抽取出来，而是通过生成服从几何分布的随机数来限制每次抽取的邻居点数量，在一定程度上修正了高度节点过度入样的问题，适当“平衡”了高度节点与低度节点的抽样比例，因此度指标的实验评估结果较好。

5.3 平均密度评估

本节对比了 SNS 算法与未抽样、NS、ES、ES-i、FFS 之间的平均密度。图 1(b)、图 2(b)、图 3(b)显示了 3 个数据集的平均密度的评估结果，实验结果表明，SNS，FFS，ES-i 算法的平均密度高于其他算法。

根据 3.2 节平均密度的定义可以得出 $density_{avg} \propto link_{real}$，在抽取节点数量相同的情况下，3 种算法都能抽取更多的边。在图中，高度节点往往是“枢纽点”，这些点抽取得越多，网络中实际存在的边数就越多，网络的连通性就越好，密度就越大。由于 SNS 算法采用分层抽样策略，在一次抽样过程中能够抽取比 NS 算法更多的高度节点，因此密度较大。

ES-i 算法由于使用了导出子图概念，在抽取节点数量相同的情况下，相比于 ES 算法能够抽取到更多的边，使得其密度变大，网络连通性更好。

FFS 算法基于 BFS 策略，BFS 算法本身的抽样准确率高，抽样密度大，FFS 算法继承了该优点，抽样密度也较大。

5.4 平均直径评估

各算法在 3 个不同的数据上的平均直径实验评估结果如图 1(c)、图 2(c)、图 3(c)所示。

实验结果表明，SNS 算法的性能远胜于其他算法(FFS 算法与 SNS 算法的结果接近)，这主要得益于 SNS 算法采用的分层抽样技术解决了 NS 算法存在的低度节点过度入样的问题。根据 3.2 节中的平均直径定义可知，平均直径是最短路径的最大值，通常任何两点之间的最短路径会经过这些高度节点。SNS 算法在每次抽样时都能抽取到一定比例的高度节点，“平衡”了高度节点和低度节点的抽取比例，因此其网络直径和原图非常接近。

在一次抽样中，FFS 算法采用的抽样机制适当调整了高度节点与低度节点的抽样比例，这是其平均直径评估结果较好的主要原因。NS 算法和 ES 算法的性能最差，ES-i 算法的性能优于 ES 算法，原因同 5.2 节。

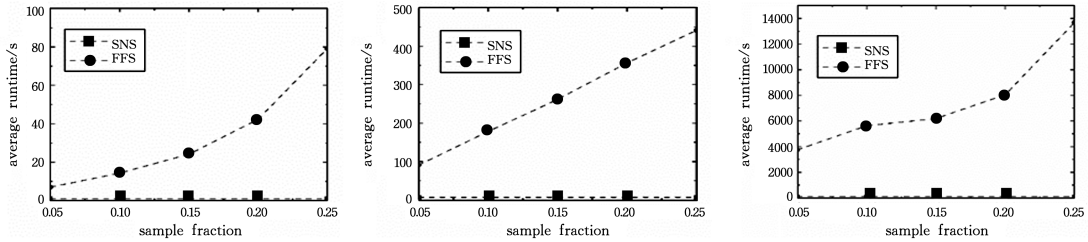
5.5 聚类系数评估

本节给出了 5 种算法在不同数据集上的平均聚类系数的实验结果及其在 15%抽样规模下不同数据集聚类系数的分布情况，如图 1(d)和图 1(f)、图 2(d)和图 2(f)、图 3(d)和图 3(f)所示。

根据实验结果可知，SNS 算法和 FFS 算法结果相近，并且二者都优于其他算法。其原因在于 SNS 算法采用了分层抽样策略，使得算法误差最小。FFS 算法的评估结果较好的原因同5.2 节。总体而言，使用 SNS 算法和 FFS 算法得到的抽样结果的结构保留得最完整。NS 算法和 ES 算法的实验结果较差，原因同 5.2 节。

5.6 SNS 算法与 FFS 算法的运行时间比较

由以上实验分析可知，SNS 算法和 FFS 算法的抽样性能比较接近，但运行时间却相差较大。图 4 显示了在不同数据集以及不同抽样比例下两种算法运行时间的对比情况。显然，SNS 算法的运行时间几乎不会随着抽样比例的增大而增大，并且其平均时间远小于 FFS 算法，而 FFS 算法的运行时间随着抽样规模的增大而增加。



(a)Facebook 数据集 (b)Condmat 数据集 (c)Amazon 数据集

图 4 在不同数据集以及不同抽样比例下，SNS 算法和 FFS 算法的运行时间比较

Fig. 4 Comparison of running time of SNS algorithm and FFS algorithm under different datasets and sampling rates

产生上述时间开销差异的主要原因在于：FFS 算法每次迭代各个种子点时都需要通过广度优先遍历算法寻找其邻居点，需要多次扫描边表，导致 FFS 算法的时间复杂度为多项式时间；而 SNS 算法由于只是增加了 k-Means 聚类步骤，k-Means 算法的时间复杂度为线性，这使得 SNS 算法也能够在线性时间内完成。

实验结论：1)NS 算法和 ES 算法的思路简单，但抽样性能最差，在抽样时没有考虑节点本身的差异性；2)SNS 算法由于采用了基于近似度分布的分层策略，算法的抽样误差小，抽样性能好；3)使用 k-Means 算法获取节点的近似度分布，这使得 SNS 算法能在线性时间内完成。

结束语 本文在基于点选择策略的均匀随机抽样算法的基础上，通过结合节点的近似度分布策略，设计并实现了高效无偏的分层图抽样算法 SNS，并且通过实验验证了该算法的

高效性和准确性。后续工作会研究在不同层内采用不同抽样策略、不同抽样比例对 SNS 算法抽样性能的影响，也会考虑将分层策略应用于基于边选择和基于图拓扑结构抽样算法中，并通过实验分析抽样效果。

致　谢　感谢西安电子科技大学计算机学院博士生导师高琳教授、西北农林科技大学信息工程学院贾敏对本文提出的宝贵意见。

参 考 文 献

[1] LESKOVEC J，FALOUTSOS C. Sampling from large graphs [C]// ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM，2006：631-636.

[2] AHMED N K，NEVILLE J，KOMPELLA R. Network Sampling：From Static to Streaming Graphs[J]. Acm Transactions on Knowledge Discovery from Data，2013，8(2)：1-56.

[3] WATTS D J，STROGATZ S H. Collective dynamics of 'small-world' networks[J]. Nature，1998：393(6684)：440-442.

[4] FALOUTSOS M，FALOUTSOS P，FALOUTSOS C. On power-law relationships of the Internet topology[J]. Acm Sigcomm Computer Communication Review，1999：29(4)：251-262.

[5] HAN J. Data Mining：Concepts and Techniques[M]. San Francisco，CA，USA：Morgan Kaufmann Publishers Inc，2005.

[6] HAMERLY G. Making k -means even faster[C] // SIAM International Conference on Data Mining 2010. 2010：130-140.

[7] JORGENSEN M. EM Algorithm[M]// Encyclopedia of Environmetrics. John Wiley & Sons，Ltd，2006：267-292.

[8] PARK H S，JUN C H. A simple and fast algorithm for K-Medoids clustering[J]. Expert Systems with Applications，2009，36(2)：3336-3341.

[9] DASZYKOWSKI M，WALCZAK B. Density-Based Clustering Methods[M]// Comprehensive Chemometrics. Elsevier，2010.

[10] CAMPELLO R J G B，MOULAVI D，SANDER J. Density-Based Clustering Based on Hierarchical Density Estimates[C]// Pacific-Asia Conference on Knowledge Discovery & Data Mining. 2013：160-172.

[11] SNAP[OL]. http：//snap. stanford. edu/data/index. html.

[12] DASGUPTA A，KUMAR R，SIVAKUMAR D. Social sampling [C]// ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM，2012：235-243.

[13] GJOKA M，KURANT M，BUTTS C T，et al. Walking in Facebook：A Case Study of Unbiased Sampling of OSNs[C]// INFOCOM，2010 Proceedings IEEE. IEEE，2010：1-9.

[14] PALMER C R，FALOUTSOS C. Density biased sampling：an improved method for data mining and clustering[J]. Acm Sigmod Record，2000，29(2)：82-92.

[15] CALLAGHAN M，CALLAGHAN M，CALLAGHAN M，et al. LinkBench：a database benchmark based on the Facebook social graph[C]// ACM SIGMOD International Conference on Management of Data. ACM，2013：1185-1196.

[16] FLESCA S，GRECO S. Querying graph databases[C]// International Conference on Extending Database Technology：Advances in Database Technology. Springer-Verlag，2000：510-524.

[17] MAASS S，MIN C，KASHYAP S，et al. Mosaic：Processing a Trillion-Edge Graph on a Single Machine[C]// Twelfth European Conference on Computer Systems. ACM，2017：527-543.

[18] ZHAO J，WANG P，LUI J C S，et al. Sampling Online Social Networks by Random Walk with Indirect Jumps[OL]. http：//www. researchgate. net/publication/319391447 _ Sampling _ Online_Social_Networks_by_Random_Walk_with_Indirect_Jumps.

[19] WAGNER C，SINGER P，KARIMI F，et al. Sampling from Social Networks with Attributes[C]// Conference www'17 Proceedings of the 26th International Conference on World Wide Wep. 2017：1181-1190.

[20] HASAN M A. Methods and Applications of Network Sampling [C]// SIAM Conference on Data Mining. 2016.

[21] YU L. Sampling and characterizing online social networks[C]// IISA 2016. 2016：245-249.

[22] REZVANIAN A，MEYBODI M R. Sampling algorithms for weighted networks[J]. Social Network Analysis & Mining，2016，6(1)：1-22.

[23] VOUDIGARI E，SALAMANOS N，PAPAGEORGIOU T，et al. Rank degree：An efficient algorithm for graph sampling[C]// IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining. IEEE，2016：120-129.

[24] ZHAO J，LUI J C S，TOWSLEY D，et al. A tale of three graphs：Sampling design on hybrid social-affiliation networks[C]// IEEE，International Conference on Data Engineering. IEEE，2015：939-950.

[25] CUI Y A，LI X，WANG Z X，et al. A Comparison on Methodologies of Sampling Online Social Media[J]. Chinese Journal of Computers，2014，37(8)：1859-1876. (in Chinese)
崔颖安，李雪，王志晓，等. 在线社交媒体数据抽样方法的比较研究[J]. 计算机学报，2014，37(8)：1859-1876.

[26] TANG J T，WANG T，WANG J. Shortest Path Approximate Algorithm for Complex Network Analysis[J]. Journal of Software，2011，22(10)：2279-2290. (in Chinese)
唐晋韬，王挺，王戟. 适合复杂网络分析的最短路径近似算法[J]. 软件学报，2011，22(10)：2279-2290.