软件学报 ISSN 1000-9825, CODEN RUXUEW
*Journal of Software*,2020,31(11):3540−3558 [doi: 10.13328/j.cnki.jos.005843]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

# 基于采样的在线大图数据收集和更新*

尹子都[1], 岳 昆[1], 张彬彬[1], 李 劲[2]

[1](云南大学 信息学院,云南 昆明 650500)

[2](云南大学 软件学院,云南 昆明 650500)

通讯作者: 岳昆, E-mail: kyue@ynu.edu.cn

**摘 要**: 互联网中,以网页、社交媒体和知识库等为载体呈现的大量非结构化数据可表示为在线大图.在线大图数据的获取包括数据收集和更新,是大数据分析与知识工程的重要基础,但面临着数据量大、分布广、异构和变化快速等挑战.基于采样技术,提出并行、自适应的在线大图数据收集和更新方法.首先,将分支限界方法与半蒙特卡罗采样技术相结合,提出能够自适应地收集在线大图数据的 HD-QMC 算法;然后,为了使收集的数据能反映实际中在线大图的动态变化,进一步基于信息熵及泊松过程,提出高效更新在线大图数据的 EPP 算法.从理论上分析了该算法的有效性,并将获取的各类在线大图数据统一表示为 RDF 三元组的形式,为在线大图数据分析及相关研究提供方便易用的数据基础.基于 Spark 实现了在线大图数据的收集和更新算法,人工生成数据和真实数据上的实验结果展示了该方法的有效性和高效性.

**关键词**: 在线大图;数据收集;数据更新;并行爬虫;Spark

**中图法分类号**: TP311

中文引用格式: 尹子都,岳昆,张彬彬,李劲.基于采样的在线大图数据收集和更新.软件学报,2020,31(11):3540−3558. http://www.jos.org.cn/1000-9825/5843.htm

英文引用格式: Yin ZD, Yue K, Zhang BB, Li J. Sampling-based collection and updating of online big graph data. Ruan Jian Xue Bao/Journal of Software, 2020,31(11):3540−3558 (in Chinese). http://www.jos.org.cn/1000-9825/5843.htm

## Sampling-based Collection and Updating of Online Big Graph Data

YIN Zi-Du[1], YUE Kun[1], ZHANG Bin-Bin[1], LI Jin[2]

[1](School of Information Science and Engineering, Yunnan University, Kunming 650500, China)

[2](School of Software, Yunnan University, Kunming 650500, China)

**Abstract**: The large volume of unstructured data obtained from Web pages, social media and knowledge bases on the Internet could be represented as an online big graph (OBG). Confronted with many challenges, such as its large-scale, widespread, heterogeneous, and fast-changing properties, OBG data acquisition includes data collection and updating, which is the basis of massive data analysis and knowledge engineering. In this study, the method for adaptive and parallel data collection and updating is proposed based on sampling techniques. First, the HD-QMC algorithm is given for adaptive data collection of OBG data by combining the branch-and-bound method and quasi-Monte Carlo sampling technique. Next, the EPP algorithm is given for efficient data updating based on entropy and Poisson process to make the collected data reflect the dynamic change of OBGs in real-world environments. Further, the effectiveness of the proposed algorithms is analyzed theoretically, and various kinds of collected OBG data are represented by triples universally to provide an easy-to-use data foundation for OBG analysis and relevant studies. Finally, the proposed algorithms for data collection and updating are

* 基金项目: 国家自然科学基金(U1802271, 62002311); 云南省基础研究计划杰出青年项目(2019FJ011); 云南省青年拔尖人才培养支持计划(C6193032); 云南大学东陆学者培育计划

Foundation item: National Natural Science Foundation of China (U1802271, 62002311); Science Foundation for Distinguished Young Scholars of Yunnan Province (2019FJ011); Young Talent Support Program of Yunnan Province(C6193032); Donglu Scholars Training Program of Yunnan University

收稿时间: 2018-10-25; 修改时间: 2018-12-08, 2019-01-16; 采用时间: 2019-03-26

implemented with Spark, and experimental results on simulated and real-world datasets show the effectiveness and efficiency of the proposed method.

**Key words**: online big graph; data collection; data updating; parallel crawler; Spark

互联网中存在着大量有价值的非结构化数据,这些数据以不同的载体呈现,是内容全面的信息资源.图结构是互联网中非结构化数据的一种主要组织形式,可表示为在线大图(online big graph,简称 OBG).典型的 OBG 包括网页、社交媒体和知识图库,如图 1 所示.OBG 数据的获取,包括数据收集和更新,是解决大数据分析、知识工程和决策支持等实际问题的重要前提和基础[1,2],在 Web 搜索与海量数据分析[3]、数据集成[4,5]、数据抽取与融合[6]等领域发挥着重要作用.一个 OBG 包括对象和连接,不同类型的 OBG 其对象和连接不同.

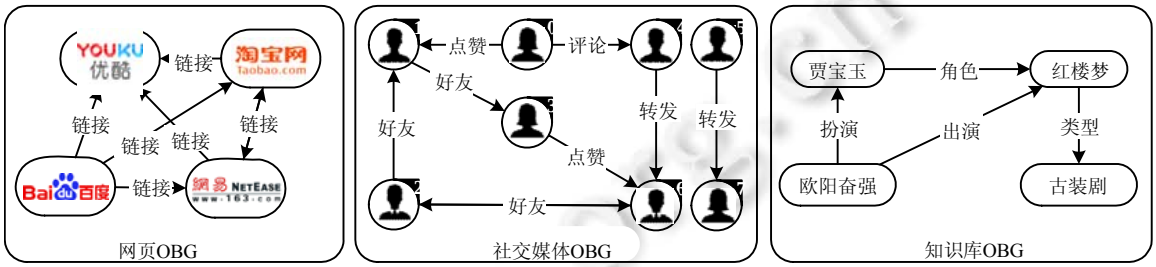



Fig.1 Typical OBGs

图 1 典型的 OBG

OBG 中的数据具有海量、分布、异构和快速变化等特点[7],并且在数据收集之前,其全局的图结构是未知的,这使得 OBG 数据的收集和更新都面临着数据量巨大、数据分布范围广、数据结构复杂且变化迅速等方面的挑战,因此无法获取 OBG 中全部对象的数据.如何优先获取 OBG 中有价值、重要的数据,是一个在线优化问题.解决这一问题主要面临如下难点和挑战:(1) OBG 数据的收集是在线(online)的,初始时,OBG 的结构完全未知,随着收集到的数据不断增加,OBG 的结构才逐渐被发掘,在线性使得数据收集中对象的选取变得更加困难;(2) 完全没有 OBG 数据的情况下,无法根据特定需求按照重要性由大到小的次序收集 OBG 中的对象,通过分析已收集的 OBG 数据,发现重要对象的分布规律,也是提升后续数据获取效率的保证.因此,本文围绕如何优先收集重要对象和利用已收集的 OBG 数据来优化其数据更新过程,讨论 OBG 数据的收集和更新方法,为 OBG 数据处理与分析的相关问题奠定基础.

以通用爬虫(universal crawler)和优先爬虫(preferential crawler)为代表的 OBG 数据收集方法主要关注 OBG 的局部结构,但是针对数据收集过程的在线性,不能从全局出发寻找重要的数据;同时,基于网页布局[8]或历史数据统计的 OBG 数据的更新方法[9]通过分析已收集的 OBG 数据得到变化规律,从而优化 OBG 数据的更新,但对于数据变化规律并未包含于历史数据的情形仍不能有效更新.采样技术被广泛用于统计机器学习[10],它能够方便地处理全局性问题.半蒙特卡洛采样(quasi Monte Carlo sampling,简称 QMC)与传统的蒙特卡洛采样(Monte Carlo sampling,简称 MC)技术不同,使用伪随机序列生成采样点,使得采样更加均匀和全面,避免了由于随机采样点过近而导致重要性评估效果下降的问题.因此,以 OBG 数据的有效获取为目标,本文借鉴优先爬虫的思想,基于 QMC 采样技术,提出自适应的 OBG 数据收集算法,并在其基础之上给出用于 OBG 数据更新的算法.针对 OBG 数据的海量性,我们利用 Spark 平台实现 OBG 数据的并行获取,保证 OBG 数据的高效获取.

首先,对于给定的 OBG,本文结合 QMC 采样技术和求解优化问题的分支限界方法,提出在线算法 HD-QMC,从全局角度提升 OBG 数据收集的效果.我们将 OBG 对象集合映射到高维空间,并对高维空间分割得到子空间,再利用 QMC 采样技术,先收集采样点对应的 OBG 对象,再由此评估各个子空间的重要性,对最重要的子空间递归地执行上述过程,直到得到 OBG 中所有对象的数据,从而完成 OBG 数据的收集.然后,将已收集的数据统一表示为 RDF(resource description framework)的形式[11],为后续研究提供统一的数据接口.进一步地,本文从理论上

给出 HD-QMC 算法的有效性、复杂度、标准误差、迭代次数和冲突率等指标的分析.

接着,借鉴基于历史数据统计的传统方法,本文结合信息熵、泊松过程[12]和前述的数据收集方法,提出数据更新算法 EPP.作为数据收集方法的扩展,EPP 算法首先通过信息熵计算各个子空间的信息量,并利用泊松过程预测每个子空间可能产生的增量,将 QMC 采样过程中得到的实际子空间增量的大小与预测增量的大小进行融合,由此优先对增量较多的子空间进行采样,得到采样对象的增量,从而完成 OBG 数据的更新.

最后,我们基于 Spark 平台实现本文提出的算法,在真实数据集和生成数据集上,将 HD-QMC 和 EPP 算法与现有方法在有效性和效率方面进行了对比,展示了本文方法针对实际应用中不同 OBG 数据的收集和更新效果.与其他方法相比,本文的方法能更快地发现大多数重要的对象,且具有良好的可扩展性.

## 1 相关工作

- OBG 数据收集

通用爬虫和优先爬虫是主要的 OBG 数据收集工具.通用爬虫利用经典的广度优先[13]和深度优先[14]方法,仅根据局部图结构进行数据收集.但 OBG 中各个部分的重要性在实际情况下并不相同,通用爬虫没有考虑到不同部分重要性的差异.优先爬虫主要包括主题爬虫(topic crawler)和聚焦爬虫(focused crawler)[15]:主题爬虫根据特定的主题从 Web 上搜索信息;聚焦爬虫通过使用基于应用、链接[16]和语义[17]的方法,在数据收集过程中能够优先收集用户关注的内容,关注的内容可以是关键词或用户定义的其他需求.但是现有的优先爬虫大部分同样由于受到 OBG 的在线性限制,主要关注局部图结构,为了快速发现 OBG 中重要的部分,还需要进一步扩展.与广度优先和深度优先方法不同,本文方法并不是从 OBG 中的某个对象开始逐渐扩大收集范围,而是从 OBG 的全局出发,优先收集重要部分中的数据.

- 基于采样技术的数据收集

使用这类方法可得到全局的图结构信息[18],采样技术的引入,使得数据收集过程能够区别对待重要性不同的部分,且采样技术与 OBG 数据收集还有很多结合点,例如,图流采样方法[19]从海量的图流数据中发现并收集重要的图数据.但与本文方法不同,这种方法需要对所有图流数据进行分析,并不能在部分数据未知的情况下完成采样.Yin 等人[20]提出一种基于 QMC 采样技术的聚焦爬虫,将 OBG 中的对象映射为一维向量,在此向量上,通过 QMC 采样技术估算不同区域的重要性并收集数据.但是对于结构复杂的 OBG,一维映射会导致信息丢失,影响重要性评估.另外,此方法对重要性的度量依据只有连接,对用户关注的其他信息表达不足.与上述方法不同,本文通过将 OBG 中的对象映射到高维空间,并利用高维 Halton 序列生成采样点,提高了子空间重要性评估的准确性,同时使用户可以选择需要的信息作为重要性评估的依据.

- OBG 数据更新

早期 OBG 数据更新的方法使用 Revisit[21]策略,直接重新收集对象数据来替换本地已有数据.但是随着 OBG 数据的快速增长,导致每次数据更新会产生巨大的开销,并且不能满足数据更新的速度要求.为此,很多研究专注于缩小收集范围来降低数据更新的成本.例如:Xi 等人[8]提出一种基于网页布局模式的方法,能够适用于复杂的真实数据环境,借助网页布局模式来判断某一页面变化的可能性,但对于页面布局模式相似的对象来说,并不能准确地为每个对象给出预测结果;Pavai 等人[9]根据历史数据计算每个对象发生变化的概率,但这种方法无法预测历史数据中没有出现过的对象;Radinsky 等人[22]提出一种考虑了网页间连接结构的更新预测方法,通过分析不同网页之间的结构关系,得出数据更新策略,但是随着网页间连接关系变得复杂,预测的准确性会受到一定的影响;Cho 等人[23]提出基于采样的数据更新方法,通过对 OBG 某个部分进行少量的采样,判断其变化的可能性,但该方法还不能用于自适应的数据更新.本文的数据更新方法建立在提出的数据收集方法之上,同时结合了统计方法与 OBG 中的真实变化两个方面,能够快速且有效地更新已收集数据.

## 2 基于采样的 OBG 数据收集

**定义 1**. 一个 OBG 是一个有向图 $G_{ON}=\{O,E,R\}$,其中,$O$,$E$ 和 $R$ 分别是对象、连接和关系的集合.$G_{ON}$ 中的每

个连接都有一种连接关系类型并连接两个不同对象,即 $e=(o_i,o_j,r_k),e\in E,o_i,o_j\in O,i\neq j,r_k\in R$.

**定义 2**. $G_{ON}$ 中 $o_i$ 的对象数据 $\Psi_{o_i}$ 是通过访问互联网获取的 OBG 数据,包括对象本身和此对象产生的所有连接,可表示为 $o_i$ 以及 $e_i=(o_i,o_j,r_k)$,其中,$e_i\in E,o_i,o_j\in O,i\neq j,r_k\in R$,全体对象的对象数据集合记为 $\Psi$.

$\Psi$包含 $G_{ON}$中的所有信息,所以获取 $G_{ON}$等价于获取 $\Psi$.鉴于$\Psi$对不同对象是独立可分的,因此可将 OBG 的对象集合映射到高维空间,再对其进行子空间划分,这样就能在高维子空间中使用 QMC 采样来量化不同子空间的重要性,从而进行自适应的 OBG 数据收集.

### 2.1 OBG高维空间映射与子空间划分

我们将 $G_{ON}$ 的对象集合 $O=\{o_1,o_2,\ldots,o_{|O|}\}$映射到一个高维空间 $A$,$|O|$为 $G_{ON}$ 中对象的数量.映射后,第 $i$ 个对象的高维空间坐标可表示为

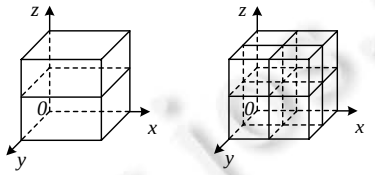
$$\left(\mathrm{mod}\left(\left\lfloor\frac{i}{L^0}\right\rfloor,L\right),\mathrm{mod}\left(\left\lfloor\frac{i}{L^1}\right\rfloor,L\right),\ldots,\mathrm{mod}\left(\left\lfloor\frac{i}{L^{h-2}}\right\rfloor,L\right),\left\lfloor\frac{i}{L^{h-1}}\right\rfloor\right) \tag{1}$$

其中,$L$ 为 $\sqrt[h]{|O|}$,$h$ 代表空间维度.

QMC 采样使用高维伪随机序列进行采样,高维空间 $A$ 的维度 $h$ 越高,QMC 采样的伪随机性和子空间覆盖性越好,子空间分割也能够越精细,重要性计算的结果也越准确.对于 OBG 而言,恰当的 $h$ 就足以表达其内部的信息,多余的维度会造成计算资源的浪费.因此,对于复杂的 OBG,适当提高维度 $h$ 将能够更快发现重要的子空间,提升数据获取的效率.

$A$ 中不同的子空间重要性不同,且这些子空间内不同的子空间重要性也不相同.为了计算子空间的重要性,可将 $A$ 分割为 $K$ 个等大小的子空间,记为$\{A_1,\ldots,A_s,\ldots,A_K\}$,每个子空间也可继续划分,直到只包含单个对象.第 $J$ 次划分结果表示为划分向量 $\vec{k}_J$,其每个维度取值为各个子空间该维度上等分分割份数的倒数.子空间的递归细分将会产生一棵高度为 $\log_K|O|$的 $K$ 叉树,根节点对应整个高维空间,第 1 层节点分别对应分割后的 $K$ 个子空间,其余层依此类推.例 1 展示了 OBG 的高维空间映射与子空间分割过程.基于此的分割方法,下一节将讨论如何利用 QMC 采样递归的获取子空间重要性并进行自适应的数据收集.

例 1:设图 1 中社交媒体 OBG 对应的有向图为 $G_{ON}=\{O,E,R\}$,$O=\{o_0,\ldots,o_7\}$,$E=\{e_0=(o_2,o_1,r_0),\ldots,e_8\}$,且 $R=\{r_0=$好友,$r_1=$评论,$r_2=$转发,$r_3=$点赞$\}$.若 $h=3$,根据公式(1)对 $G_{ON}$ 进行空间映射可得,$o_0$~$o_7$ 对应坐标分别为(0,0,0),(1,0,0),(0,1,0),(1,1,0),(0,0,1),(1,0.1),(0,1,1),(1,1,1).此时,OBG 将会被映射到一个三维的立方体内.对这个立方体进行子空间分割,$K=2$ 时先对 $z$ 轴分割,则立方体在 $x$,$y$ 和 $z$ 这 3 个维度上分别被等分为 1 份、1 份和 2 份,对应的分割向量为 $\vec{k}_1=(1,1,0.5)$,如图 2(a)所示.接着,用 $\vec{k}_2=(0.5,1,1)$和 $\vec{k}_3=(1,0.5,1)$对 $x$ 轴和 $y$ 轴划分,得到 8 个子空间,如图 2(b)所示.后续分割过程类似,直到空间不再可分.




(a) 仅 $z$ 轴分割 (b) $x$,$y$ 和 $z$ 轴分割

Fig.2 Example of OBG projection and split

图 2 OBG 的映射与分割举例

### 2.2 基于采样的OBG数据收集算法

结合前述 OBG 的高维空间映射与划分方法,基于高维空间的 QMC 采样技术和分支限界方法,本节给出自适应的 OBG 数据收集方法,包括子空间重要性评估和子空间选择两个阶段.

在子空间重要性评估阶段,我们首先将高维空间$A$按照子空间划分方法进行分割,并由QMC采样得到每个子空间的重要性.将需要收集的信息视为目标信息,例如网页OBG中特定主题的信息,或社交媒体OBG中的用户信息和关注、转发、评论及点赞等社交行为.子空间中目标信息量又决定了子空间的重要性,我们使用目标信息密度来度量子空间的重要性,目标信息密度与聚焦爬虫中数据收集目标的概念类似,反映了OBG数据收集时子空间中的目标信息量.下面给出子空间$A_s$中目标信息密度的概念.

$$\rho_{A_s}=\frac{\sum_{o\in A_s}D(\Psi_o)}{|A_s|} \tag{2}$$

其中,$D(\Psi_o)$代表一个OBG中对象$o$所包含的目标信息量.若将社交媒体OBG中的社交行为作为目标信息,则子空间的目标信息密度为该子空间中社交行为的数量与子空间对象总数的比值.

为了快速计算子空间的重要性,我们利用QMC采样技术,通过少量采样点尽可能精确地计算不同子空间的目标信息密度,由此找到最重要的子空间.QMC使用确定性的方法生成低差异的伪随机序列(low-discrepancy sequence)[24],并将其作为采样点,这些采样点对子空间的覆盖性优于随机点.QMC通常将Halton序列作为其伪随机序列进行采样,令$b$为一个素数,则任何一个正整数$Z^+$可表示为$d_jb^j+\ldots+d_ib^i+\ldots+d_1b+d_0$,其中,$d_i\in\{0,1,\ldots,b-1\}$,且$i=0,1,\ldots,j$.在一个Halton序列$W$中,第$w$个元素$\phi_b(w)$为$\frac{d_0}{b^1}+\frac{d_1}{b^2}+\ldots+\frac{d_j}{b^{j+1}}$.同时,对于任何$w>0$,有$\phi_b(w)\in[0,1]$.对于$h$个不同的素数$b_1,\ldots,b_d$,$n$个$h$维的Halton序列为$\{\vec{x}_1,\ldots,\vec{x}_n\}$,$\vec{x}_i$可表示为

$$\vec{x}_i=[\phi_{b_1}(i-1),\ldots,\phi_{b_n}(i-1)]^T,i=1,\ldots,n \tag{3}$$

因此,第$J$次子空间划分后,第$i$个采样点在每个子空间中的相对坐标为$\vec{x}_i\circ\vec{k}_1\circ\ldots\circ\vec{k}_J\times\sqrt[h]{|O|}$,其中,"$\circ$"代表哈达马积(Hadamard product).令采样点数$n=|A_s|\cdot R_{samp}$,其中,$R_{samp}$为采样比率,$\{o_1,\ldots,o_i,\ldots o_n\}\in A_s$为子空间$A_s$中用Halton序列生成的$n$个采样点对应的采样对象,则$\rho_{A_s}$可作如下近似计算.

$$\rho_{A_s}\approx\frac{1}{n}\sum_{i=1}^{n}D(\Psi_{o_i}) \tag{4}$$

值得说明的是,以上采样过程不仅可得到子空间的重要性,还可得到采样点的对象数据$\Psi_{o_i}$.因此,本文的数据收集方法通过采样过程完成,不仅并行地对每次分割得到的各个子空间进行采样,而且并行地收集每个子空间内部的采样对象$o_i$,从而得到$\Psi_{o_i}$,已收集的对象数据集合表示为$\Psi^c=\{\Psi_1^c,\Psi_2^c,\ldots,\Psi_{|O|}^c\}$.为了使得相关应用能通过统一的接口访问已收集数据,我们将已收集的数据统一表示为RDF形式,记为〈对象1,关系,对象2〉的三元组形式,其中,"关系"可以表示网页OBG中的链接、社交媒体OBG中的社交操作以及知识库OBG中实体间的关系类型.

在子空间选择阶段,我们根据QMC采样得到的子空间目标信息密度找出下一次迭代的子空间.所有本次及以往迭代中访问过的子空间的目标信息密度都被记录在一个候选子空间集合$P_{cand}$中,$P_{cand}=\{\rho_{A_1},\rho_{A_2},\ldots,\rho_{A_K},\rho_x\}$,其中,$\rho_x$是之前迭代访问过的子空间的目标信息密度.为了能够自适应地收集数据,始终从$P_{cand}$中选取目标信息密度最大的子空间作为下一次迭代的高维空间.

由于存在某些目标信息较少且无需收集的子空间,如分散孤立的部分,使得数据收集效率较低.对此,下面给出算法的结束条件.

$$\bar{\rho}<\rho_{\min} \tag{5}$$

其中,$\bar{\rho}$是所有采样的平均目标信息密度,$\rho_{\min}$是能够接受的最小子空间目标信息密度.随着大多数高密度子空间逐渐收集完成,剩下的子空间中的目标信息将越来越少,$\bar{\rho}$相应下降.最终,当$\bar{\rho}<\rho_{\min}$时,整个收集过程结束.

上述思想由算法1描述.

**算法 1**. HD-QMC.

输入:$A,K,R_{samp},P_{cand}$.

输出:$\Psi^c$.

步骤:

**IF** $|A|>1$ AND $\bar{\rho} \geqslant \rho_{\min}$ **THEN**
  $S \leftarrow Divide(A,K)$ //分为 $K$ 个子空间,得到集合 $S$
  **FOR EACH** $s$ IN $S$ **DO** //$s$ 为子空间
    $mass \leftarrow 0$
    $W \leftarrow HaltonSeq(s,R_{samp})$ //生成采样点集合 $W$
    **FOR EACH** $w$ IN $W$ **DO**
      **IF** $w \notin \Psi^c$ **THEN**
        $obj \leftarrow Collect(w)$
        $\Psi^c \leftarrow \Psi^c \cup obj$
      **END IF**
      $mass \leftarrow mass+NumInfo(obj)$ //加入新增的目标信息数量
    **END FOR**
    **IF** $|s|>1$ **THEN**
      $density \leftarrow mass/|W|$
      $P_{cand}.Add([s,density])$
    **END IF**
  **END FOR**
  $A \leftarrow P_{cand}.PickMax(\cdot)$ //取出密度最大的子空间
  $HD\text{-}QMC(\cdot)$ //下一次迭代
**END IF**

算法 1 中的 FOR 循环可并行完成.算法迭代执行,对子空间进行分割并产生一棵 $K$ 叉树,每个非叶子节点代表一次迭代并对应一个子空间,每次迭代将从一个非叶子节点产生 $K$ 个分支,对原来子空间进一步划分,并产生新的子空间.子空间不可分,则为叶子节点且不产生新的分支.例 2 给出了基于算法 1 的 OBG 数据收集过程.

例 2:基于例 1 得到的高维空间 $A$ 及其子空间,以社交行为“关注”“转发”“评论”“点赞”作为目标信息,则子空间的重要性取决于对象的出度,整个数据收集过程将进行多次 QMC 采样迭代并生成一棵二叉树,其上部为子空间内的所有对象,下部表示子空间的目标信息密度,分支上标注了 QMC 采样点对应的对象,如图 3 所示.二叉树中的第 1 层~第 3 层节点对应的子空间,分别由上层节点对应的子空间经过分割向量 $\vec{k}_1,\vec{k}_2,\vec{k}_3$ 分割得到.算法优先选择目标信息密度最大的节点进行迭代,执行顺序为图中序号,最终得到所有对象数据集合 $\Psi^c$.

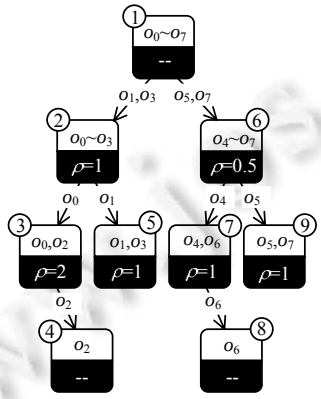



Fig.3 Example of data collection

图 3 数据收集过程举例

### 2.3 分 析

(1) 有效性

给定高维空间 $A$,将对象数据收集过程中目标信息的增长曲线与坐标轴所构成区域的“面积”,即收集过程中目标信息的积分,作为算法 1 的有效性,用 $S_A$ 描述.

$$S_A=\sum_{i=1}^{|A|}\sum_{j=1}^{i}D(\Psi_j^c) \tag{6}$$

其中,$|A|$是 $A$ 中的对象总数,$\Psi_j^c$ 表示第 $j$ 个收集到的对象数据.越重要的对象越早被收集,$S_A$ 越大.

(2) 复杂度

算法 1 的时间复杂度取决于 $K$ 叉树的高度$\lfloor\log_K|O|\rfloor$和采样比 $R_{samp}$.$K$ 叉树中,下一层采样是对当前这一层子空间分割并评估的过程,每生成一层,都会对$|O|$个对象按照 $R_{samp}$ 重新采样,并评估子空间的目标信息密度.因此,算法 1 的时间复杂度可表示为 $O(|O|\cdot R_{samp}\cdot\lfloor\log_K|O|\rfloor)=O(|O|\cdot\log|O|)$.

(3) 标准误差

标准误差是对子空间目标信息密度的估计误差.算法 1 的标准误差在不同采样点 $n$ 下不同.

标准误差的估计为 $|A_s|\frac{\sigma_n}{\sqrt{n}}$,其中,$\sigma_n=\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\left(D(\Psi_{o_i})-\frac{1}{n}\sum_{i=1}^{n}D(\Psi_{o_i})\right)^2}$.

算法 1 的 $|A_s|\frac{\sigma_n}{\sqrt{n}}$ 随着采样点数的上升以非线性趋势下降.

(4) 迭代次数

迭代次数是算法执行过程中对子空间分割并评估其重要性的次数,记为 $N_{iter}$.算法 1 的迭代次数取决于子空间的分割数 $K$ 和对象总数$|O|$,迭代次数由定理 1 给出.

**定理 1**. 给定一个有$|O|$个对象和 $K$ 个子空间的 OBG($|O|>0$,$K>1$),可以得到 $N_{iter}$ 为

$$\begin{cases}1, & 0<|O|<K\\ |O|-K^{m-1}+\sum_{j=0}^{m-1}K^j, & K^m\leqslant|O|\leqslant 2K^m\\ \sum_{j=0}^{m}K^j, & 2K^m<|O|<K^{m+1}\end{cases} \tag{7}$$

其中,$m=\lfloor\log_K|O|\rfloor$,$m\in\mathbf{Z}^+$.

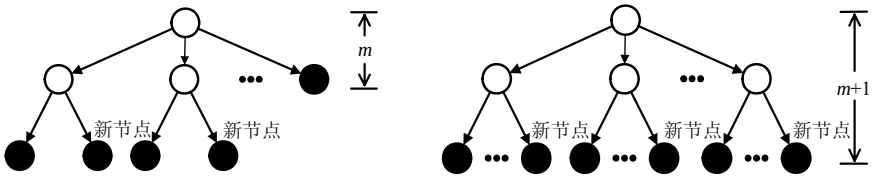
证明:根据对象数的不同可分为 3 种情形,下面依次进行说明.

情形 1:对象数在 0 到 $K$ 之间,即 $0<|O|<K$.由于每个子空间至少会有一个对象被收集,所以这种情况下所有的对象将在一次迭代内收集完成,因此 $N_{iter}=1$.

情形 2:对象数在 $K^m$ 到 $2K^m$ 之间,即 $K^m\leqslant|O|\leqslant 2K^m$.首先用叶子节点代表对象,所有非叶子节点代表一次迭代,称为迭代节点.若 $K^m=|O|$,则为一棵完全 $K$ 叉树有 $\sum_{j=0}^{m-1}K^j$ 个叶子节点.当一个对象被加入时,就会有新的迭代节点形成,该位置上原来的叶子节点和新加入的叶子节点将作为该迭代节点的孩子节点.因此,一共有 $|O|-K^{m-1}$ 个新加入的节点,叶子节点按照同样的方法加入到这个 $K$ 叉树中,如图 4(a)所示.最终,$N_{iter}=|O|-K^{m-1}+\sum_{j=0}^{m-1}K^j$.

情形 3:对象数在 $2K^m$ 到 $K^{m+1}$ 之间,即 $2K^m<|O|<K^{m+1}$.当先前的叶子节点都变成迭代节点时,新加入的节点将不再产生新的迭代节点,直接加入现有的迭代节点中,直到形成完全 $K$ 叉树.这种情况下,迭代次数将一直保持为 $N_{iter}=\sum_{j=0}^{m}K^j$,且所有的叶节点都在 $K$ 叉树的同一层,如图 4(b)所示.

证毕. □




(a) 叶子节点在不同层 (b) 叶子节点在同一层

Fig.4 Iteration under different object numbers (white: iteration node, black: leaf node)

图 4 不同对象数下的迭代情况(白色节点为迭代节点,黑色节点为叶子节点)

(5) 冲突率

算法 1 在不同迭代节点对同一对象采样,则产生冲突,冲突越少,算法 1 执行的效率越高.冲突用冲突率表示,记为

$$E_{conf}=\frac{N_{conf}}{N_{iter}\cdot K\cdot|A_s|\cdot R_{samp}} \tag{8}$$

其中,$|A_s|\cdot R_{samp}$ 是每次迭代过程中各个子空间的采样点数,$N_{conf}$ 是整个数据收集过程中的冲突总数.

定理 2 定量地描述了所有的冲突.

**定理 2**. 给定一个包含$|O|$个对象的 OBG,冲突总数为

$$N_{conf}=\sum_{i=1}^{N_{iter}}\sum_{j=1}^{m_i}P_{ij} \tag{9}$$

其中,$N_{iter}$ 由定理 1 得出,$m_i=\lfloor\log_K|A^i|\rfloor$,$A^i$ 是第 $i$ 次迭代的采样子空间,$P_{ij}$ 是子空间 $A^i$ 对应的 $K$ 叉树中第 $j$ 层分割得到的子空间中不重叠的冲突数.

证明:当算法 1 从 $K$ 叉树自顶向下进行采样时,冲突最初从第 1 层迭代节点开始产生.

通过对子空间 $A^i$ 中各级分割子空间中的非重叠冲突进行求和,得到第 $i$ 次迭代的冲突数为 $\sum_{j=1}^{m_i}P_{ij}$ ,那么该过程的冲突总数为 $\sum_{i=1}^{N_{iter}}\sum_{j=1}^{m_i}P_{ij}$ .证毕. □

假设 $K$ 叉树对应的各级采样子空间中的冲突数为一个固定的值,则 $N_{conf}$ 的近似值为 $N_{iter}\cdot\sum_{j=1}^{m_i}P_{ij}$ .因此,冲突数以及 $E_{conf}$ 可通过调整 $K$ 和 $R_{samp}$ 来降低.

## 3 基于泊松过程的 OBG 数据更新

### 3.1 数据更新概述

**定义 3**. 令 $G_{ON}=\{O,E,R\}$为一个 $T$ 时刻的 OBG,在 $T'(T'>T)$时为 $G'_{ON}=\{O',E',R'\}$ .$G_{ON}$ 从 $T$ 到 $T'$的增量可表示为

$$\Delta G_{ON}=\{DIFF(O',O),DIFF(E',E),DIFF(R',R)\} \tag{10}$$

$DIFF(O',O)$可表示为 $[O'\setminus(O'\cap O)]\cup\overline{O\setminus(O'\cap O)}$ ,用来描述新加入与删除对象的集合,并以此作为 $G'_{ON}$ 以及 $G_{ON}$ 之间的差异.$\Delta G_{ON}$ 中,$E$ 和 $R$ 的增量可用类似方式表达.

根据定义 3,增量可以更具体表示为 $O$,$E$ 和 $R$ 集合中元素的添加和删除操作,因此增量可由定义 4 表示.

**定义 4**. 更新操作集合 $U$ 包括添加和删除,分别记为 $u$ 和 $\bar{u}$ .$O$,$E$ 和 $R$ 的更新操作集合分别记为 $U_O$,$U_E$ 和 $U_R$,$\Delta G_{ON}=\{U_O,U_E,U_R\}$.

**定义 5**. 从 $T_0$ 到 $T$ 时刻收集的 OBG 数据中,统计窗口是 $T_0$ 到 $T$ 时刻内且结束于 $T$ 时刻的时间区间,定义为 $(T-\alpha(T-T_0),T]$,如图 5 所示.其中,$\alpha$称为窗口因子($0<\alpha\leqslant 1$).

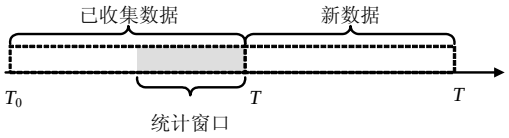



Fig.5 Statistic window of data updating

图 5 数据更新的统计窗口

数据更新方法建立在数据收集方法之上,包括增量预测和增量搜索两个阶段.数据更新方法将定义 4 表示的增量作为目标信息,子空间重要性的度量依据仍采用目标信息密度,在数据更新中为增量密度.因此,数据更新方法可看作是对第 2 节中数据收集方法的扩展.

增量预测阶段利用 HD-QMC 算法输出的已收集数据 $\Psi^c$,通过使用信息熵和泊松过程对统计窗口内 $G_{ON}$ 各子空间变化的可能性进行量化,预测各个子空间的增量密度,得到各个子空间的初始增量密度.增量搜索阶段以增量作为收集目标,基于并扩展 HD-QMC 算法,给出 QMC 采样得到的增量密度与增量预测阶段预测得到的增量密度进行合理融合的方法,并使用融合后的增量密度作为子空间重要性的度量依据,以此帮助发现更多的增量,并用增量来更新本地数据,从而优化数据更新过程.

### 3.2 基于信息熵的数据更新算法

在增量预测阶段,假设当前已收集数据的统计窗口中对应数据的 OBG 为 $G_{ON}^B$,我们将 $G_{ON}^B$ 映射到高维空间并分割为 $K$ 个子空间,即 $G_{ON}^B=\{B_1,B_2,...,B_K\}$.泊松过程是累计随机事件发生次数的独立增量过程[12],同时,$B_j$ 内增量的产生并没有确定的规律且相互独立,是一个随机事件,因此某段时间内,$B_j$ 中增量产生的大小可用泊松过程来描述和预测.从 $T$ 到 $T'$时刻,用信息熵表示子空间的信息量,则可根据信息熵运用泊松过程来预测未来一段时间内增量产生的可能性,以此对这些子空间进行独立的更新.

**定义 6**. 设一个 OBG 中从 $T_0$ 到 $T$ 时刻 $O$,$E$ 和 $R$ 的出现概率分别为 $P(o_i)$,$P(e_i)$和 $P(r_i)$,其信息熵分别为

$$H(O)=E[I(o_i)],H(E)=E[I(e_i)],H(R)=E[I(r_i)].$$

其中,$I(o_i)=-\log P(o_i)$,$I(e_i)=-\log P(e_i)$,$I(r_i)=-\log P(r_i)$.子空间 $B_j$ 的信息量即为

$$Y_j=|O_j|\cdot H(O)+|E_j|\cdot H(E)+|E_j|\cdot H(R).$$

其中,$O_j$ 和 $E_j$ 即为对象集合和连接集合,$o_i,e_i\in B_j,j=1,2,\ldots,K$.

**定义 7**. 令 $\lambda=\dfrac{Y_j}{\alpha(T-T_0)}$ 为泊松分布的均值,若在$\Delta T$ 内增量的大小为 $\tau$,则增量产生的概率为

$$P\{X(\Delta T+T)-X(T)=\tau\}=\mathrm{e}^{-\lambda\Delta T}\frac{(\lambda\Delta T)^{\tau}}{\tau!} \tag{11}$$

其中,$X$ 是泊松过程,且 $\tau=0,1,\ldots$.

令 $U_j$ 为 $B_j$ 增量大小的预测,满足 $P\{X(\Delta T+T)-X(T)=U_j\}=\mathrm{Max}\{P\{X(\Delta T+T)-X(T)=\tau\}\}$.

因此,可以得到 $U=\{U_1,U_2,\ldots,U_j,\ldots,U_K\}$.

最后,将 $U_j/|B_j|$作为子空间的初始增量密度,并加入到 $P_{cand}$ 中.

在增量搜索阶段,由于从 $G_{ON}^B$ 的子空间 $B_j$ 中得到的 $U_j$ 不一定能很好地反映真实的增量大小,但在 OBG 的采样过程中能对真实的增量大小进行评估,因此我们引入融合比率$\beta$,将 $U_j/|B_j|$与采样得到的增量密度进行合理融合,以便从 $K$ 叉树的第 2 层开始指导子空间的选取,并快速发现 OBG 中真实的增量.实际采样过程中,对于 $K$ 叉树中任意层的迭代节点,第 $j$ 个子空间的增量密度表示为 $V_{jl}$,通过 QMC 采样方法计算:

$$V_{jl}\approx\frac{1}{n}\sum_{\varepsilon=1}^{n}\left(\left\lceil N_\varepsilon\frac{H(O)}{N_\varepsilon+1}\right\rceil+N_\varepsilon\cdot H(E)+N_\varepsilon\cdot H(R)\right) \tag{12}$$

其中,$N_\varepsilon$和 $n$ 分别为采样得到的新连接数和采样点数,$l$ 为 $K$ 叉树的层数.$\left\lceil N_\varepsilon\dfrac{H(O)}{N_\varepsilon+1}\right\rceil+N_\varepsilon\cdot H(E)+N_\varepsilon\cdot H(R)$ 代表

第$\varepsilon$个采样点的增量大小,则融合后的 $K$ 叉树中第 $l$ 层第 $j$ 个子空间的增量密度表示为

$$V'_{jl}=(1-\beta)V_{jl}+\beta\left(\frac{V'_{j(l-1)}}{K}\right)(0\leqslant\beta\leqslant 1,l>1) \tag{13}$$

$V'_{j(l-1)}$ 为包含当前子空间的上一级子空间增量密度,即 $K$ 叉树中当前迭代节点的父节点对应子空间的增量密度.特别地,$V'_{j1}=U_j/|B_j|$.

上述思想见算法 2.

**算法 2**. EPP.

输入:初始增量密度最大的子空间 $A_{\max}$,$K$,$R_{samp}$, $\Psi^c$,包含所有子空间增量密度初始值的 $P_{cand}$.

输出:更新后的 $\Psi^c$.

步骤:

```
IF |A_max|>1 AND ρ̄ ≥ ρ_min THEN
  S←Divide(A_max,K)  //K 个子空间与已收集数据相同
  FOR EACH s IN S DO
    vol←0
    W←HaltonSeq(s,R_samp)  //生成采样点集合 W
    FOR EACH w IN W DO
      obj←Collect(w)
      temp←Find(Ψ^c,w)  //在 Ψ^c 中查找 w
      IF obj≠temp THEN
        vol←vol+IncVol(obj,temp)  //计算增量大小
        Update(Ψ^c,w,obj)  //用 obj 更新 Ψ^c 中的 w
      END IF
    END FOR
    IF |s|>1 THEN
      density←(1−β)·vol/|W|+β·HistoryDensity(·)
      P_cand.Add([s,density])
    END IF
  END FOR
  A_max←P_cand.PickMax(·)  //取出下一个子空间
  EPP(·)  //下一次迭代
END IF
```

与算法 1 类似,同样使用 QMC 采样技术来发现全部增量,包括新的对象、连接和连接类型.算法 2 中的 *IncVol* 计算当前采样点的增量大小.*HistoryDensity* 用于得到当前子空间的 $V'_{j(l-1)}/K$.算法 2 与算法 1 的时间复杂度相同,都为 $O(|O|\cdot\log|O|)$.算法 2 的执行过程可由例 3 表示.

例 3:基于例 2 得到的 $\Psi^c$,令 $\alpha$=1,$\beta$=0.2.若新产生的数据为 $e_9$=($o_3$,$o_4$,$r_2$),$e_{10}$=($o_4$,$o_5$,$r_3$),$e_{11}$=($o_5$,$o_6$,$r_0$),根据定义 6, $G^B_{ON}$ 中 $O$,$E$ 和 $R$ 的信息熵分别为 $H(O)$=0.903,$H(E)$=0.954,$H(R)$=0.553.

数据更新过程如图 6 所示,子空间划分方式及采样序列与例 2 一致,图中节点下部左侧与右侧各代表 $V'_{jl}$ 和 $V_{jl}$.算法优先选择融合后增量密度最大的节点进行迭代,顺序由图中的标号给出,其中,对 $o_4$,$o_5$ 和 $o_3$ 的采样过程中发现了新数据 $e_{10}$,$e_{11}$ 和 $e_9$,并将其加入到已收集数据中,已收集数据与当前 OBG 上的数据保持同步.

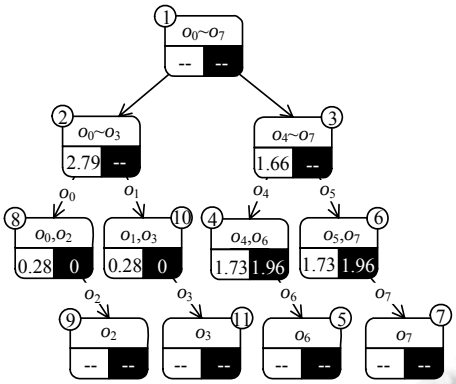



Fig.6 Example of data updating

图 6 数据更新过程举例

## 4 实验结果

### 4.1 实验设置

为了测试本文提出方法的性能,我们基于 Spark 平台实现本文的算法,分别测试了算法 1(HD-QMC)和算法 2(EPP)的有效性和效率.Spark 集群包括 6 计算节点,每个计算节点拥有 2 个 10 核心/20 线程/3.6GHz 的 CPU 和 128GB 内存.各计算节点共用一个千兆交换机,且网络带宽限制为 2Mb/s.Spark 和 HDFS 版本分别为 1.6.1 和 2.5.2.实验使用的测试数据集见表 1.

**Table 1** Test datasets

**表 1** 测试数据集

| 测试类型 | 数据集 | $\|O\|(\times10^3)$ | $\|E\|(\times10^4)$ | $\|R\|$ |
|---|---|---|---|---|
| 收集 | Ber-Stan[25] | 15 | 150 | 1 |
| | Facebook[26] | 4 | 88 | 4 |
| | Wikidata(https://www.wikidata.org/wiki/Wikidata:Database_download,2018) | 10 | 49 | 2 900 |
| 收集/更新 | 微博(Microblog)[27] | 40 | 1 000 | 4 |
| 更新 | LFR | 15 | 16 | 4 |

### 4.2 有效性测试

(1) HD-QMC 算法有效性测试

为了测试 HD-QMC 数据收集的有效性,本文选用 Ber-Stan、Facebook、微博和 Wikidata 分别作为典型的网页 OBG、社交媒体 OBG 和知识库 OBG.需要说明的是,实验所使用的 Wikidata 数据集是存储在维基网站上的数据,实验中需要通过网络访问,使用前 10 000 个实体作为测试数据集.Ber-Stan、Facebook 和微博数据集预先下载到本地磁盘,测试时模拟了数据收集的真实环境,仿真了数据收集过程中网络访问的开销.

将 OBG 中对象之间的连接作为目标信息,并以目标信息覆盖度来度量 OBG 中目标信息的收集进度,由 $\sum_{\Psi_i\in\Psi^c}D(\Psi_i)\Big/\sum_{\Psi_j\in\Psi}D(\Psi_j)$ 计算得到,其中,$\Psi^c$ 和 $\Psi$ 分别代表已收集数据的集合与全体数据的集合.针对目标信息覆盖度对本文提出的 HD-QMC 进行测试,并与顺序收集(sequence)、深度优先策略(DFS)、Snowball 采样(snowballing)[13]、随机选择(random)和基于 QMC 采样技术的方法(BB-QMC)[20]进行了对比,分别如图 7(a)~图 7(d)所示.其中,

- Sequence、DFS 和 Snowballing 是传统的顺序、深度优先和 Snowball 采样方法.
- Random 方法随机选取对象进行数据收集.
- BB-QMC 是一种先进和新颖的数据收集方法,其依照不同子空间的重要性自适应的收集 OBG 数据.但

与本文方法不同的是,BB-QMC 将 OBG 中的对象映射为一维向量,在此一维向量空间上,通过 QMC 采样技术估算不同子空间的重要性并收集数据.对于复杂的 OBG,由于对高维空间的投影往往会造成信息的丢失,一维空间在信息表示上与高维空间相比存在劣势.

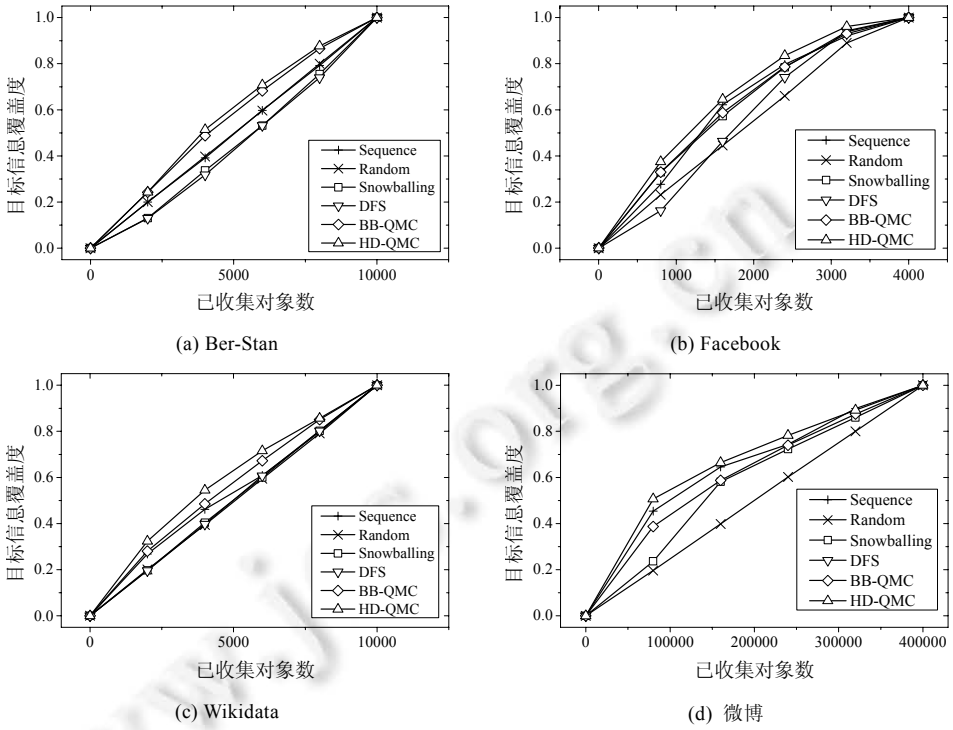



Fig.7 Target information coverage of different methods for data collection

图 7 不同数据收集方法的目标信息覆盖度

从图 7 可以看出,随着收集对象的增加,由于对不同子空间的重要性考虑不足,Snowballing、Sequence 和 DFS 的目标信息覆盖度与 Random 相近;而 HD-QMC 和 BB-QMC 的目标信息覆盖度显著增加,且 HD-QMC 在大多数情况下增加较快.这是由于 HD-QMC 将对象映射到更高维度的空间中,从而能更细致地分割和计算子空间的重要性,因此能优先收集目标信息较多的子空间.同时,给定$\rho_{\min}$,由于 HD-QMC 能够更快地收集重要的对象,所以相比其他方法能够更早地结束且$|\Psi^t|$最小,所消耗的成本也最低.由此可见,HD-QMC 能够高效地收集 3 类典型的 OBG 数据.

接着,本文测试了 HD-QMC 执行过程中不同 $R_{samp}$($R$)对目标信息覆盖度的影响,如图 8(a)~图 8(d)所示.可以看出,当 $R_{samp}$ 大于 0.05 时,在前 3 个数据集上都可得到较好的数据收集结果;而 $R_{samp}$ 为 0.01 时,在微博数据集上的数据收集效果最好.由此可知,不同数据集需要选取合适的 $R_{samp}$ 来提升数据收集效果.

同时,本文测试了子空间分割数 $K$ 对目标信息覆盖度的影响,如图 9(a)~图 9(d)所示.在 Ber-Stan、Wikidata 和微博数据集上,$K$ 分别为 30,15 和 45 时,可得到最佳目标信息覆盖度.在 Facebook 数据集上,初始 $K$ 为 30 时得到了较好的结果;但当已收集的对象数量大于 1 500 之后,$K$ 为 45 时得到最好的效果.由此可知,针对不同的 OBG 数据可设置适当的 $K$ 值,以达到较好的数据收集效果.

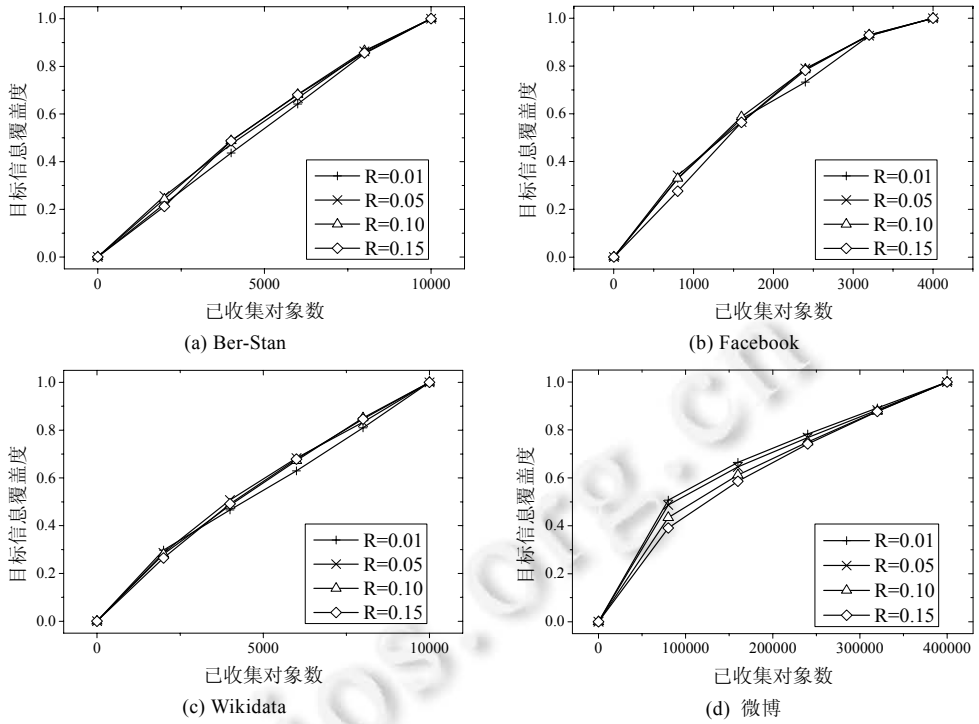



Fig.8 Target information coverage of HD-QMC with different $R_{samp}$

图 8 HD-QMC 针对不同 $R_{samp}$ 的目标信息覆盖度

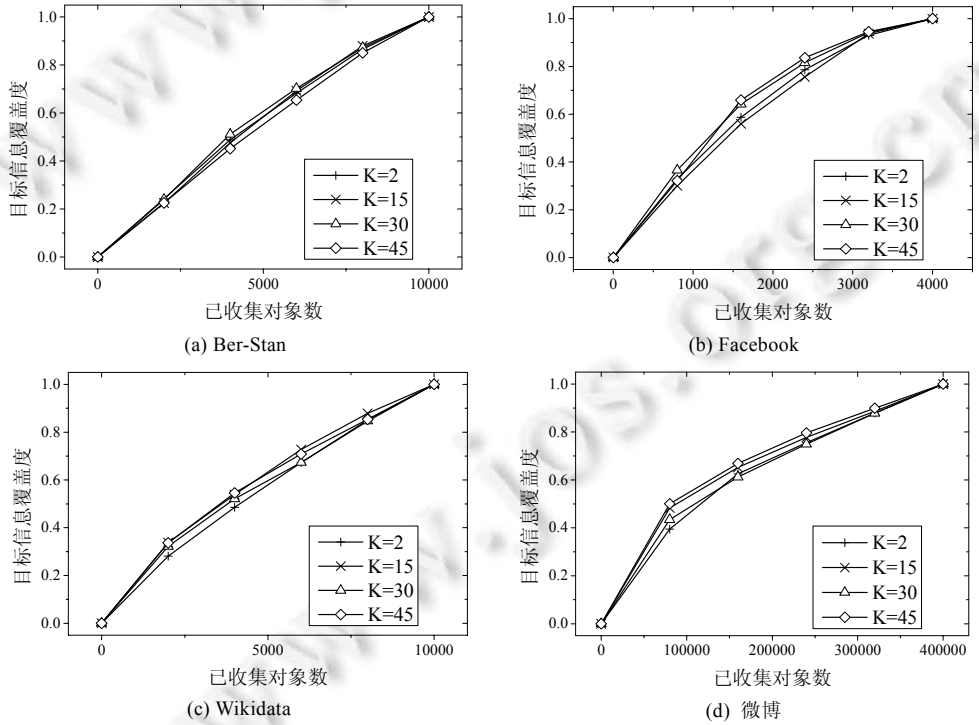



Fig.9 Target information coverage of HD-QMC with different $K$ values

图 9 HD-QMC 针对不同 $K$ 值的目标信息覆盖度

进一步地,我们根据公式(6)计算了上述对比方法与 HD-QMC 在各个数据集下的有效性指标($S_A$),见表 2.不难看出,由于 HD-QMC 能够更精确地计算子空间的重要性,因此所有数据集上获得了最高的 $S_A$,并取得了最好的数据收集效果.DFS 方法由于无法在微博数据集上运行,表 2 中对应值未列出.

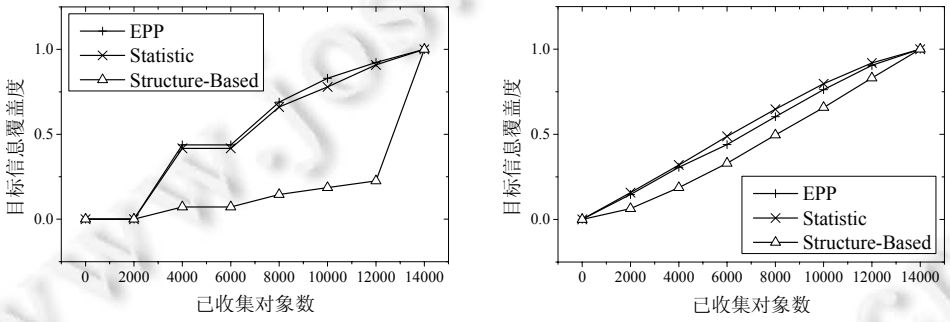
**Table 2** $S_A$ of different methods for data collection

**表 2** 不同数据收集方法的 $S_A$

| 数据集 | Sequence | Random | Snowballing | DFS | BB-QMC | HD-QMC |
|---|---|---|---|---|---|---|
| Wiki-data | 53.45 | 50.02 | 50.62 | 50.48 | 56.45 | 59.27 |
| Ber-stan | 49.99 | 51.06 | 45.12 | 44.36 | 56.44 | 57.50 |
| Facebook | 21.51 | 18.23 | 21.30 | 19.13 | 22.83 | 22.99 |
| 微博 | 505.54 | 401.04 | 471.53 | N/A | 488.31 | 550.73 |

(2) EPP 算法有效性测试

我们用 2 个典型的数据集测试 EPP 算法的有效性:第 1 个是时间跨度 2 周、约 11 000 个用户的真实微博数据集,第 1 周数据作为已收集数据,第 2 周数据作为新数据;第 2 个是由扩展的 LFR 方法[28]生成的 LFR 数据集,以此来测试当社交媒体中存在极端变化时,EPP 的数据更新效果.与 HD-QMC 的测试相似,我们测试了不同方法的目标信息覆盖度,包括本文提出的 EPP、基于统计的方法(Statistic)[9]和基于结构的方法(structure-based)[22],如图 10(a)、图 10(b)所示.其中,基于统计的方法通过统计并计算历史数据中对象变化的概率进行数据更新,基于结构的方法通过计算对象间不同连接结构的变化概率进行数据更新.




(a) 微博 (b) LFR

Fig.10 Target information coverage of different methods for data updating

图 10 不同数据更新方法的目标信息覆盖度

根据公式(6),以上 3 种方法的有效性指标($S_A$)见表 3.

**Table 3** $S_A$ of different methods for data updating

**表 3** 不同数据更新方法的 $S_A$

| 数据集 | Statistic | Structure-based | EPP |
|---|---|---|---|
| 微博 | 41.03 | 10.99 | 42.17 |
| LFR | 115.62 | 90.49 | 109.78 |

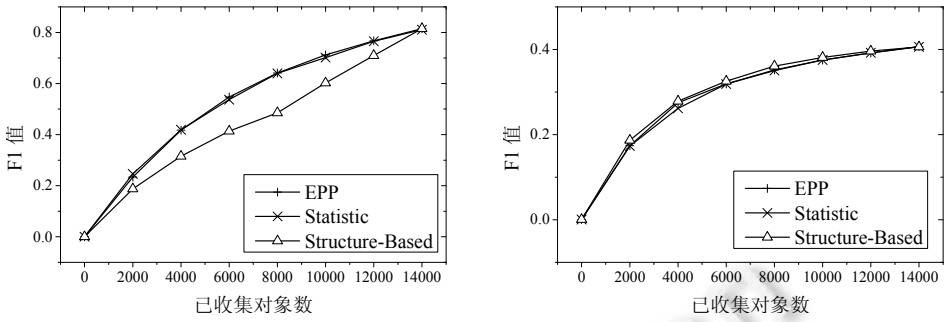
结合图 10 和表 3 可以看出,对于真实的微博数据集,EPP 比其他方法能够更好地发现增量,但是基于统计的方法在 LFR 数据集上取得了更好的数据更新效果.

我们进一步用 $F1$ 值测试 EPP 针对数据更新的有效性.

$$F1=\frac{2\cdot Pr\cdot Re}{Pr+Re} \tag{14}$$

其中,$Pr$ 和 $Re$ 分别表示 *Precision* 值和 *Recall* 值,*Precision* 是 EPP 发现的增量与其预测增量的比值,*Recall* 是 EPP 发现的增量与真实增量的比值.测试结果如图 11(a)、图 11(b)所示,可以看出,EPP 在微博和 LFR 数据集上优于

基于统计和基于结构的方法.

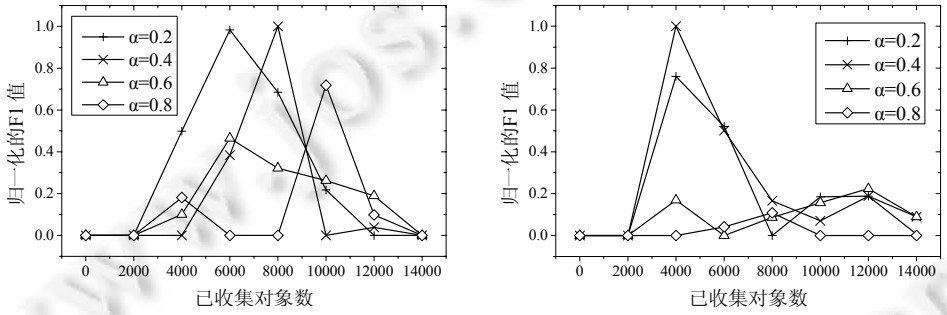



(a) 微博 (b) LFR

Fig.11 $F1$ values of different methods for data updating

图 11 不同数据更新方法的 $F1$ 值

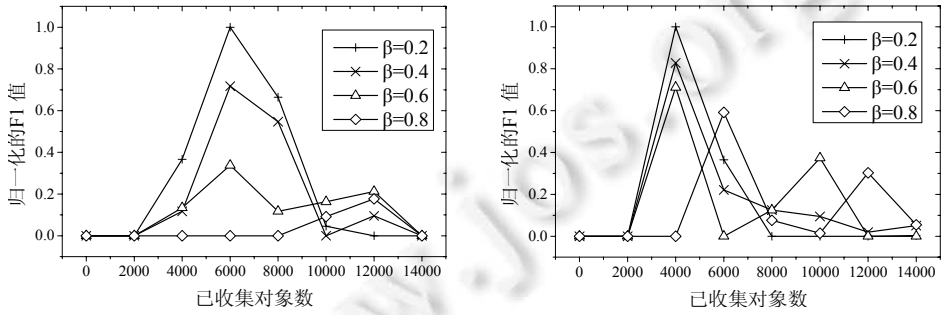
为了测试 EPP 算法中参数对数据更新效果的影响,本文比较了微博和 LFR 数据集上不同窗口因子($\alpha$)和融合比率($\beta$)时的 $F1$ 值.为了便于观察,我们对测试结果进行归一化,分别如图 12 和图 13 所示.




(a) 微博 (b) LFR

Fig.12 Normalized $F1$ value with different window factors

图 12 不同窗口因子下归一化的 $F1$ 值




(a) 微博 (b) LFR

Fig.13 Normalized $F1$ value with different fusion ratios

图 13 不同融合比率下归一化的 $F1$ 值

在微博和 LRF 数据集上,$\alpha$分别为 0.2 和 0.4 时 EPP 算法取得最好的效果.在微博上,$\beta$为 0.2 时算法效果最好.而 LFR 数据集上,$\beta$为 0.2,0.4 和 0.6 的情况较为接近:访问对象小于 8 000 时,$\beta$为 0.2 最好;大于 8 000 时,$\beta$为

0.6 的效果最好.总体上,$\beta$为 0.2 时效果最好,对应曲线与坐标轴之间所构成区域的面积最大.由此可知,对于不同的数据集,应选取合适的$\alpha$和$\beta$.

### 4.3 效率测试

EPP 算法与 HD-QMC 算法执行效率接近,实验通过网络访问 Wikidata 数据集,测试了 HD-QMC 的执行时间、加速比和并行效率,分别如图 14~图 16 所示.

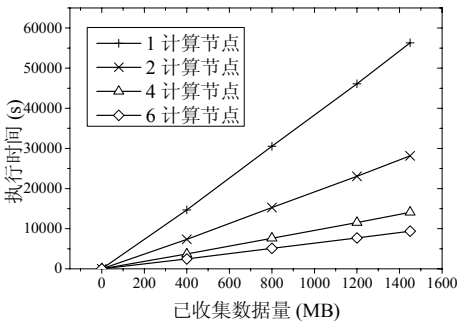



Fig.14 Execution time

图 14 执行时间

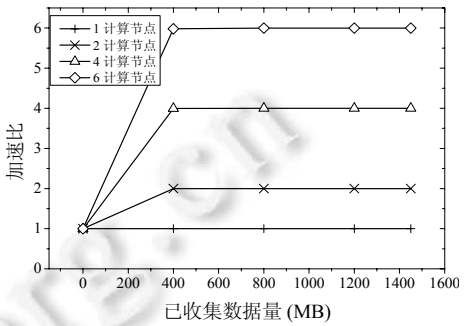



Fig.15 Speedup

图 15 加速比

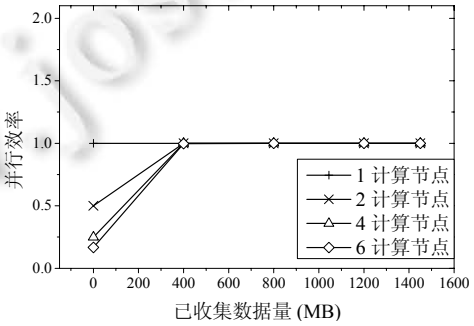



Fig.16 Parallel efficiency

图 16 并行效率

由图 14 可知,HD-QMC 数据收集的执行时间随着 OBG 中已访问对象数的增加基本呈线性增长,且 Spark 平台的计算节点越多执行时间越少,6 个节点时的总执行时间与 1 个节点时相比明显减少,为 1 节点下执行时间的 1/6.图 15 中的加速比和图 16 中的并行效率在 HD-QMC 开始执行时便趋于稳定,加速比接近计算节点数,同时,并行效率也接近理论最优值 1.以上情况产生的原因是由于 HD-QMC 的执行时间很大程度上依赖于网络带宽,而多计算节点下的网络带宽会随着计算节点数同步增长.以上实验结果与理论分析得出的结论一致,进一步验证了本文方法的高效性.

为了对比 HD-QMC 与传统算法的执行效率,实验测试了 HD-QMC 与 Sequence、Random、HD-QMC、Snowballing 在执行时间与可扩展性上的差异.我们首先在 1 个计算节点情形下测试了不同算法的执行时间,得到不同算法单线程执行效率.接着在 6 节点情形下再次测试算法执行时间,获取不同算法的可扩展性表现,分别如图 17 和图 18 所示.

从图 17 可以看出,1 个计算节点情形下,Random 和 Snowballing 算法执行较慢,HD-QMC 和 BB-QMC 算法稍快,Sequence 算法由于直接从对象列表中依次选取对象进行获取,所以最快.1 节点下单线程的 HD-QMC 没有充分发挥可扩展性上的优势,但在 6 个计算节点情形下多线程测试中执行效率提升明显.从图 18 中可以看出,由于无法并行执行,因此 6 个计算节点情形下 Snowballing 和 Sequence 算法执行时间与 1 个计算节点下一致.

Random、HD-QMC 与 BB-QMC 算法在 6 个节点下的算法执行效率提升明显,由于这 3 种算法都使用了分割的思想同时对不同的子空间进行采样,执行效率都随节点数增加而提升,因此 HD-QMC 可扩展性很好,适合 OBG 数据的收集与更新.

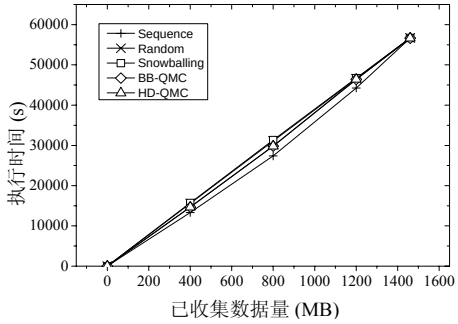



Fig.17 Execution time on 1 node

图 17 1 节点下算法执行时间

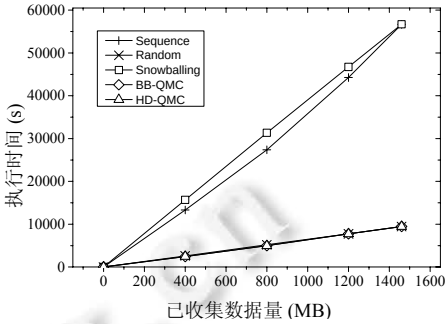



Fig.18 Execution time on 6 nodes

图 18 6 节点下算法执行时间

## 5 结论与展望

本文首先讨论了基于采样技术的自适应 OBG 数据收集,该方法能够通过 QMC 采样找到重要的子空间并尽可能好地给出 OBG 中对象数据的收集顺序,以实现 OBG 数据有效收集.同时给出数据的统一表示方法,降低数据集成和使用的成本.进一步地,我们扩展数据收集方法,给出高效的数据更新算法,既能利用历史数据中的数据变化规律,又能利用增量中新的数据变化规律,共同指导后续的数据更新过程.即使数据不断变化,本文的算法总能快速发现增量,完成高效的数据更新.实验结果表明,在大多数情况下,本文方法能够取得较好的效果,可为大数据分析和知识工程提供方便易用的数据基础.但是,对部分社交媒体 OBG 数据的获取,还需针对其本身的特点进一步研究,后续工作将针对社交媒体数据中社交行为与社区演化的具体过程,探索相应的 OBG 数据获取方法,且保证其有效性和高效性.

**References**:

[1] Wang JM. Key technologies in big data applications development and runtime support platform. Ruan Jian Xue Bao/Journal of Software, 2017,28(6):1516−1528 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5231.htm [doi: 10.13328/j.cnki.jos.005231]

[2] Wu XD, Chen HH, Wu GQ, Liu J, Zheng QH, He XF, Zhou AY, Zhao ZQ, Wei BF, Li Y, Zhang QP, Zhang SC. Knowledge engineering with big data. IEEE Intelligent Systems, 2015,30(5):46−55. [doi: 10.1109/MIS.2015.56]

[3] Zhang JZ, Meng XF. Mobile Web search. Ruan Jian Xue Bao/Journal of Software, 2012,23(1):46−64 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4120.htm [doi: 10.3724/SP.J.1001.2012.04120]

[4] Wang GL, Han YB, Zhang ZM, Zhu ML. Could-based integration and service of streaming data. Chinese Journal of Computers, 2017,2017(1):107−125 (in Chinese with English abstract). [doi: 10.11897/SP.J.1016.2017.00107]

[5] Xia D, Wang YS, Zhao ZP, Cui D. Incremental and interactive data integration approach for hierarchical data in domain of intelligent livelihood. Journal of Computer Research and Development, 2017,54(3):586−596 (in Chinese with English abstract). [doi: 10.7544/issn1000-1239.2017.20151048]

[6] Lin HL, Wang YZ, Jia YT, Zhang P, Wang WP. Network big data oriented knowledge fusion methods: A survey. Chinese Journal of Computers, 2017, 2017(1):1−27 (in Chinese with English abstract). [doi: 10.11897/SP.J.1016.2017.00001]

[7] Surendran S, Prasad DC, Kaimal MR. A scalable geometric algorithm for community detection from social networks with incremental update. Social Network Analysis and Mining, 2016,6(1):Article No.90. [doi: 10.1007/s13278-016-0399-9]

[8] Xi SJ, Sun FC, Wang JM. A cognitive crawler using structure pattern for incremental crawling and content extraction. In: Proc. of the IEEE Int'l Conf. on Cognitive Informatics. 2010. 238−244. [doi: 10.1109/COGINF.2010.5599733]

[9] Pavai G, Geetha TV. Improving the freshness of the search engines by a probabilistic approach based incremental crawler. Information Systems Frontiers, 2017,19(5):1013−1028. [doi: 10.1007/s10796-016-9701-7]

[10] Matteo R, Fabio V. MiSoSouP: Mining interesting subgroups with sampling and pseudodimension. In: Proc. of the 24th ACM Int'l Conf. on Knowledge Discovery & Data Mining. 2018. 2130−2139. [doi: 10.1145/3219819.3219989]

[11] Nikolov A, Haase P, Herzig DM, Trame J, Kozlov A. Combining RDF graph data and embedding models for an augmented knowledge graph. In: Proc. of the Companion of the Web Conf. 2018. 977−980. [doi: 10.1145/3184558.3191527]

[12] Andreou AS, Chatzis SP. Software defect prediction using doubly stochastic Poisson processes driven by stochastic belief networks. Journal of Systems and Software, 2016,122:72−82. [doi: 10.1016/j.jss.2016.09.001]

[13] Stivala AD, Koskinen JH, Rolls DA, Wang P, Robins G. Snowball sampling for estimating exponential random graph models for large networks. Social Networks, 2016,47:167−188. [doi: 10.1016/j.socnet.2015.11.003]

[14] Tao J, Zhao QQ, Cao PF, Wang Z, Zhang Y. APK-DFS: An automatic interaction system based on depth-first-search for APK. In: Proc. of the Int'l Conf. on Algorithms and Architectures for Parallel Processing. 2017. 420−430. [doi: 10.1007/978-3-319-65482-9_29]

[15] Khan A, Sharma DK. Self-Adaptive ontology based focused crawler for social bookmarking sites. Int'l Journal of Information Retrieval Research, 2017,7(2):51−67. [doi: 10.4018/IJIRR.2017040104]

[16] Wu CS, Hou W, Shi YQ, Liu T. A Web search contextual crawler using ontology relation mining. In: Proc. of the Int'l Conf. on Computational Intelligence and Software Engineering. 2009. 1−4. [doi: 10.1109/CISE.2009.5365842]

[17] Batzios A, Dimou C, Symeonidis AL, Mitkas PA. BioCrawler: An intelligent crawler for the semantic Web. Expert Systems with Applications, 2008,35(1-2):524−530. [doi: 10.1016/j.eswa.2007.07.054]

[18] Arulampalam MS, Evans RJ, Letaief KB. Importance sampling for error event analysis of HMM frequency line trackers. IEEE Trans. on Signal Processing, 2002,50(2):411−424. [doi: 10.1109/78.978395]

[19] Ahmed NK, Duffield N, Willke TL, Rossi RA. On sampling from massive graph streams. Proc. of the VLDB Endowment, 2017, 10(11):1430−1441. [doi: 10.14778/3137628.3137651]

[20] Yin ZD, Yue K, Wu H, Su YJ. Adaptive and parallel data acquisition from online big graphs. In: Proc. of the Int'l Conf. on Database Systems for Advanced Applications. LNCS 10827, Gold Coast: Springer-Verlag, 2018. 223−331. [doi: 10.1007/978-3-319-91452-7_21]

[21] Sharma V, Kumar M, Vig R. A hybrid revisit policy for web search. Journal of Advances in Information Technology, 2012,3(1): 36−47. [doi: 10.4304/jait.3.1.36-47]

[22] Radinsky K, Bennett PN. Predicting content change on the Web. In: Proc. of the 6th ACM Int'l Conf. on Web Search and Data Mining. 2013. 415−424. [doi: 10.1145/2433396.2433448]

[23] Cho J, Ntoulas A. Effective change detection using sampling. In: Proc. of the Very Large Data Bases Conf. 2002. 514−525. [doi: 10.1016/B978-155860869-6/50052-4]

[24] Faure H, Lemieux C. Improved Halton sequences and discrepancy bounds. Monte Carlo Methods Applications, 2010,16(3):1−18. [doi: 10.1515/mcma.2010.008]

[25] Leskovec J, Lang K, Dasgupta A, Mahoney M. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. Internet Mathematics, 2009,6(1):29−123. [doi: 10.1080/15427951.2009.10129177]

[26] McAuley J, Leskovec J. Learning to discover social circles in ego networks. In: Proc. of the Int'l Conf. on Neural Information Processing Systems. 2012. 539−547.

[27] Fu KW, Chan CH, Chau M. Assessing censorship on microblogs in China: Discriminatory keyword analysis and impact evaluation of the 'real name registration' policy. IEEE Internet Computing, 2013,17(3):42−50. [doi: 10.1109/MIC.2013.28]

[28] Le BD, Nguyen HX, Shen H, Falkner N. GLFR: A generalized LFR benchmark for testing community detection algorithms. In: Proc. of the Int'l Conf. on Computer Communication and Networks. 2017. 1−9. [doi: 10.1109/ICCCN.2017.8038442]

**附中文参考文献**:

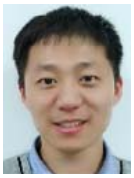
[1] 王建民.领域大数据应用开发与运行平台技术研究.软件学报,2017,28(6):1516−1528. http://www.jos.org.cn/1000-9825/5231.htm [doi: 10.13328/j.cnki.jos.005231]

[3] 张金增,孟小峰.移动 Web 搜索研究.软件学报,2012,23(1):46−64. http://www.jos.org.cn/1000-9825/4120.htm [doi: 10.3724/SP.J.1001.2012.04120]

[4] 王桂玲,韩燕波,张仲妹,朱美玲.基于云计算的流数据集成与服务.计算机学报,2017,2017(1):107−125. [doi: 10.11897/SP.J.1016.2017.00107]

[5] 夏丁,王亚沙,赵梓棚,崔达.面向智慧民生领域的增量交互式数据集成方法.计算机研究与发展,2017,54(3):586−596. [doi: 10.7544/issn1000-1239.2017.20151048]

[6] 林海伦,王元卓,贾岩涛,张鹏,王伟平.面向网络大数据的知识融合方法综述.计算机学报,2017,2017(1):1−27. [doi: 10.11897/SP.J.1016.2017.00001]



**尹子都**(1990－),男,博士生,主要研究领域为海量数据处理与分析,知识融合.

**张彬彬**(1982－),女,博士,讲师,CCF 专业会员,主要研究领域为云计算和知识发现.

**岳昆**(1979－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为海量数据处理与分析,大数据知识工程.

**李劲**(1975－),男,博士,副教授,CCF 专业会员,主要研究领域为海量数据处理和机器学习.