软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2023,34(10):4830−4850 [doi: 10.13328/j.cnki.jos.006654]
©中国科学院软件研究所版权所有.
E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

异质信息网络的复杂条件社区搜索*

王家龙, 杨 杰, 周丽华, 王丽珍, 王睿康

(云南大学 信息学院, 云南 昆明 650500)

通信作者: 周丽华, E-mail: lhzhou@ynu.edu.cn

摘 要: 社区是信息网络的重要属性, 社区搜索旨在寻找满足用户给定条件的节点集合, 是信息网络分析的重要研究内容. 异质信息网络由于包含更加全面、丰富的结构和语义信息, 所以异质信息网络的社区搜索近年来受到人们的广泛关注. 针对现有异质信息网络的社区搜索方法难以满足复杂条件社区搜索要求的不足, 定义了复杂条件社区搜索问题, 提出了考虑非对称元路径、受限元路径和禁止节点约束的搜索算法. 3 种算法分别通过元路径补全策略、调整带标签的批量搜索策略和拆分复杂搜索条件的方式搜索社区, 同时针对禁止节点约束的搜索算法设计了基于剪枝策略和近似策略的优化算法以提高搜索效率. 在真实数据集上进行了大量实验, 实验结果证明了所提算法的有效性和高效性.

关键词: 异质信息网络; 复杂条件社区搜索; 非对称元路径; 受限元路径; 禁止节点约束

中图法分类号: TP311

中文引用格式: 王家龙, 杨杰, 周丽华, 王丽珍, 王睿康. 异质信息网络的复杂条件社区搜索. 软件学报, 2023, 34(10): 4830–4850. http://www.jos.org.cn/1000-9825/6654.htm

英文引用格式: Wang JL, Yang J, Zhou LH, Wang LZ, Wang RK. Complex Conditional Community Search over Heterogeneous Information Networks. Ruan Jian Xue Bao/Journal of Software, 2023, 34(10): 4830–4850 (in Chinese). http://www.jos.org.cn/1000-9825/6654.htm

Complex Conditional Community Search over Heterogeneous Information Networks

WANG Jia-Long, YANG Jie, ZHOU Li-Hua, WANG Li-Zhen, WANG Rui-Kang

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

Abstract: Community is an important attribute of information networks. Community search, as an important content of information network analysis, aims to find a set of nodes that meet the conditions specified by the user. As heterogeneous information networks contain more comprehensive and richer structural and semantic information, community search in such networks has received extensive attention in recent years. However, the existing community search methods for heterogeneous information networks cannot be directly applied when the search conditions are complex. For this reason, this study defines community search under complex conditions and proposes search algorithms considering asymmetric meta-paths, constrained meta-paths, and prohibited node constraints. These three algorithms respectively use the meta-path completion strategy, the strategy of adjusting batch search with labeling, and the way of dividing complex search conditions to search communities. Moreover, two optimization algorithms respectively based on the pruning strategy and the approximate strategy are designed to improve the efficiency of the search algorithm with prohibited node constraints. A large number of experiments are performed on real datasets, and the experimental results verify the effectiveness and efficiency of the proposed algorithms.

Key words: heterogeneous information network; complex conditional community search; asymmetric meta-path; constrained meta-path; prohibited node constraint

---

* 基金项目: 国家自然科学基金 (62062066, 61762090, 61966036); 云南省基础研究计划重点项目 (202201AS070015); 云南省高校物联网技术及应用重点实验室; 云南大学研究生科研创新基金 (2021Y024)

收稿时间: 2021-07-31; 修改时间: 2021-09-18, 2021-12-17; 采用时间: 2022-02-16; jos 在线出版时间: 2022-12-08

CNKI 网络首发时间: 2022-12-10

© 中国科学院软件研究所 http://www.jos.org.cn

近年来, 随着信息技术的不断发展, 信息网络无处不在, 例如社会网络、学术网络、生物网络、道路网络等. 社区作为信息网络的基本属性, 本质上是由一组内部紧密连接的对象构成的集合, 这些集合的有效识别能够广泛地应用于现实生活中的诸多场景, 比如活动组织[1]、朋友推荐[2]、蛋白质识别[3]、基于位置的服务[4−7]等. 社区的研究按照分析任务不同可以分为社区检测 (community detection) 和社区搜索 (community search). 社区检测[8−10]一般采用统一的标准来对整个网络进行划分, 获取网络中的所有社区, 而社区搜索[11−13]的目标是在大型网络中针对用户给定的查询请求返回满足条件的社区, 为用户提供个性化的服务.

通常, 社区搜索的条件是用户给定的一个节点[14]或多个节点 (一个节点集)[1]. 然而现实生活中用户的需求复杂多样, 比如要求给定的节点不能出现在结果社区中, 或者至少有一个节点出现在结果社区中. 为了满足用户复杂多样的搜索要求, 文献 [15] 提出了复杂条件下的社区搜索方法, 但该方法针对的是同质信息网络, 即网络中只有一种类型的节点和一种类型的边, 所提方法并不适用于包含多种节点类型和边类型的异质信息网络[16]. 目前, 虽然已经提出了一些异质信息网络的社区搜索算法, 但这些算法的搜索条件较为简单, 比如指定的节点只有一个, 元路径是对称的[17,18], 或仅对查询节点和节点类型之间的度数进行约束[19]. 因此, 有必要研究异质信息网络中的复杂条件社区搜索方法, 以满足复杂的实际应用需求.

例如, 某期刊编辑部需要安排审稿人对收到的投稿论文进行评审. 为了公平起见, 所安排的审稿人必须与论文的作者没有合著关系并且审稿人的研究领域需要与该论文的研究领域相同. 图 1 所示是 DBLP 的网络实例, 包含作者、论文、会议、主题这 4 种对象类型, 以及论文与作者之间的撰写/被撰写关系、论文与会议之间的发表/被发表关系、论文与主题之间的包含/被包含关系. 假设编辑部收到的投稿论文为 p_x, 论文的作者是 a_4 和 a_5, 从图 1 中可以看出$\{a_1, a_2, a_3\}$是满足条件的候选审稿人. 因为 a_6 与 a_4、a_5 存在合著关系, 而 a_7、a_8 的研究主题与投稿论文的主题不符.

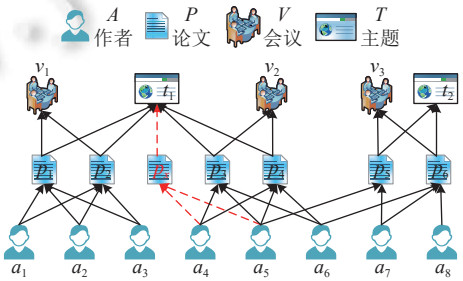

图 1 DBLP 的网络实例

经过对以上描述的分析, 该审稿人选择场景包括以下几个特点.

(1) 输入节点是投稿论文, 搜索结果是作者 (审稿人), 即搜索节点类型和搜索结果节点类型不一致.

(2) 输入节点是投稿论文, 审稿人的研究专长应该与投稿论文的主题一致, 即对搜索结果中节点的属性有特定的约束.

(3) 审稿人不能与投稿论文的任意一个作者存在合著关系, 即禁止某些特定的节点出现在搜索结果中.

显然, 审稿人选择场景涉及多种对象类型, 且对象之间包含多种关系类型, 符合异质信息网络的特点. 另外, 针对同一篇投稿论文, 编辑部需要安排多名审稿人, 且审稿人的研究专长应该尽可能相似且与投稿论文主题一致, 即审稿人之间存在较大的相似性, 该特点与社区相吻合. 由于审稿人选择需要满足一系列复杂条件, 所以可以采用异质信息网络中的复杂条件社区搜索解决. 实际上类似的应用场景还有很多, 例如项目评估中专家的遴选, 案件审理中法官的选择等. 因此异质信息网络的复杂条件社区搜索研究具有广泛的现实意义.

根据审稿人选择场景的特点, 异质信息网络的复杂条件社区搜索涉及如下几个问题.

P1. 搜索节点类型和结果社区节点类型不一致的问题.

P2. 搜索节点到结果社区节点的路径包含特定约束对象的问题.

© 中国科学院软件研究所 http://www.jos.org.cn

P3. 结果社区不能包含特定节点对象的问题.

针对上述的 3 个问题, 本文提出了相应的解决办法. 针对问题 P1, 定义了非对称元路径约束的社区搜索, 提出了元路径补全策略, 并结合对称元路径的社区搜索方法设计了非对称元路径社区搜索算法 (community search for asymmetric meta-path, CSAMP). 针对问题 P2, 定义了受限元路径约束的社区搜索, 提出了调整对称元路径的社区搜索方法中带标签的批量搜索策略, 在 CSAMP 算法的基础上设计了受限元路径社区搜索算法 (community search for constrained meta-path, CSCMP). 针对问题 P3, 定义了禁止节点约束的社区搜索, 结合非对称元路径约束和受限元路径约束, 提出了拆分复杂搜索条件的方式搜索社区, 在 CSCMP 算法的基础上设计了禁止节点约束社区搜索算法 (community search for prohibited node constraint, CSPNC).

本文对异质信息网络的复杂条件社区搜索进行深入研究, 主要贡献如下.

(1) 定义了异质信息网络的复杂条件社区搜索问题, 分析了复杂条件社区搜索和简单条件社区搜索的联系.

(2) 针对异质信息网络的复杂条件社区搜索中不同的约束条件, 给出了相应的解决办法, 设计了相应的搜索算法. 同时, 为了提高 CSPNC 算法的搜索效率, 设计了基于剪枝策略的优化算法 (CSPNC with pruning strategy, CSPNC-PS) 和基于近似策略的优化算法 (CSPNC with approximate strategy, CSPNC-AS). 这些算法很好地解决了本文所提出的问题, 能够搜索出满足用户个性化需求的社区.

(3) 在 4 个真实数据集上进行了大量的实验, 包括算法的搜索结果在有效性方面的对比实验, 基础算法与优化算法在效率上的对比实验. 实验结果验证了所提算法的有效性和高效性, 此外还给出了一个与所提问题相关的应用案例.

本文第 1 节介绍社区搜索的相关工作. 第 2 节陈述关于异质信息网络的基本概念. 第 3 节提出异质信息网络的复杂条件社区搜索问题, 包括非对称元路径约束的社区搜索, 受限元路径约束的社区搜索, 禁止节点约束的社区搜索. 第 4 节给出相应的搜索算法. 第 5 节进行实验评估. 第 6 节总结全文.

1 相关工作

当前, 社区搜索受到产业界和学术界的广泛关注, 大量的研究人员提出了各式各样的社区搜索方法. 2010 年 Sozio 等人[1]提出了针对简单无向图的社区搜索问题, 即给定一个节点集, 返回包含该节点集并且内部节点满足 *k*-Core 的社区. Sozio 等人[1]在解决该问题时采用的是全局搜索的策略, 所以搜索效率不高. 随后在 *k*-Core 社区结构的基础上, Cui 等人[14]提出了基于局部搜索策略的算法, Barbieri 等人[20]提出了基于索引的算法. 由于 *k*-Core 度量社区的凝聚性较弱, 为了提高社区的凝聚性, Huang 等人[21]于 2014 年提出了采用连通三角形 *k*-Truss 对社区的凝聚性进行度量, 并设计了相应的社区搜索算法. 在 *k*-Truss 社区结构的基础上, Akbas 等人[22]为了提高社区搜索效率, 又提出了一种新的索引技术. 然而, 在某些场景下, 社区除了需要包含给定的节点集, 可能还需要满足不能包含某些特定的节点, 于是 Wang 等人[23]提出了禁止节点感知的社区搜索问题. 2019 年, Fang 等人[24]将 *k*-Core 推广到简单有向图上, 提出了基于有向图的社区搜索问题, 其中社区结构采用的是 D-cores[25]. 2016 年 Fang 等人[26]在基于关键词的属性图上进行了社区搜索研究, 并提出了 CL-tree 的索引, 用于在分层结构中组织节点的关键词数据. 随后在此基础上又提出了维护 CL-tree 的索引结构[27], 大大加快了关键词的查找速度. 而 Huang 等人[28]则提出了一种属性驱动的 Truss 社区模型 ATC (attribute-driven Truss community), 用于查找内部节点紧密并且属性相似的社区. Zhang 等人[29]提出了以关键词为中心的社区搜索问题, 搜索条件仅包含关键词, 减轻了用户选择节点的负担. Liu 等人[30]提出了以节点为中心的属性社区搜索问题, 搜索结果考虑了多种属性. 在基于位置的属性图上, 研究人员针对不同的空间凝聚性提出了相应的社区搜索算法[31−36]. Li 等人[37]以及徐兰天等人[38]在带有时间信息的属性图上提出了持续社区搜索问题. 虽然, 社区搜索的研究取得了较多的成果, 但是无论是基于简单图的社区搜索或是基于属性图的社区搜索, 图的类型都是仅包含一种类型的节点和一种类型的边, 本质上都是基于同质信息网络的社区搜索.

然而, 现实世界中网络的节点类型和边类型复杂多样, 同质信息网络的建模方法容易丢失大量的信息, 导致得到的搜索结果不准确.于是, 部分研究人员又提出了异质信息网络的概念[16], 即网络中包含多种节点类型和边类

© 中国科学院软件研究所 http://www.jos.org.cn

型. 异质信息网络拥有更加复杂的网络结构和更为丰富的语义信息, 基于异质信息网络的社区搜索能够得到更为真实的社区, 为各种应用提供强有力的支撑. 目前, 异质信息网络中的社区搜索研究已经取得了一些进展. 其中, Fang 等人[17]最先开展在异质信息网络上的社区搜索研究, 目标是搜索与查询节点具有相同类型并且联系紧密的节点集合构成结果社区. 为了实现这个目标, Fang 等人[17]将 k-Core 与对称元路径$\mathcal{P}$相结合, 提出了 (k, $\mathcal{P}$)-Core 的概念 (即异质信息网络 G 中由$\mathcal{P}$-相连的节点构成的最大集合 H, 且 H 中每个节点都至少有 k 个$\mathcal{P}$-邻居), 并设计了一系列的社区搜索算法和索引结构加快搜索速度, 其中 FastBCore 算法采用了带标签的批量搜索 (batch search with labelling, BSL) 策略和带标签的深度优先搜索 (depth-first search with labelling, DSL) 策略, 具有良好的搜索性能. 在 FastBCore 算法[17]中, 首先, BSL 策略找出所有$\mathcal{P}$-相连到查询节点 q 的节点集合 S. 然后, 寻找 S 中的每一个节点 u 的路径实例, 将路径实例数小于 k 的节点加入队列 *Queue*. 接着, DSL 策略迭代地从队列 *Queue* 中取出节点 w, 验证 w 的每一个$\mathcal{P}$-邻居 v, 删除由 v 到 w 的路径, 并尽力寻找一条新的由节点 v 出发且终点在 S 中的路径实例, 如果存在这样的路径实例, 就将其添加到路径实例集合 $\psi[v]$ 中; 否则将节点 v 加入队列 *Queue*. 经过迭代之后, S 中的每个节点都有 k 个$\mathcal{P}$-邻居, 此时 S 即为搜索的结果社区. 随后, Wang 等人[39]为了充分利用异质社会网络中用户之间的关系信息达到有效搜索, 研究了一种在不同社交网络中在线匹配用户的技术, 提出了一种在有向社会网络中进行社区搜索的新框架 k-Dcore, 并构造了一种索引结构提高搜索效率. 由于 k-Truss 比 k-Core 具有更强的凝聚性, Yang 等人[18]基于对称元路径$\mathcal{P}$将 k-Truss 引入到异质信息网络的社区搜索问题中, 提出了 basic-triangle 和 (k, $\mathcal{P}$)-Btruss 的概念. 为了进一步加强结果社区的内聚性, 又提出了 circle-triangle 和 (k, $\mathcal{P}$)-Ctruss 的概念, 并基于 (k, $\mathcal{P}$)-Btruss 和 (k, $\mathcal{P}$)-Ctruss 设计了相应的社区搜索算法. Jian 等人[19]为了考虑节点之间度数的关系, 提出了关系社区的概念, 即允许用户指定节点间细粒度的连接要求, 并基于关系社区提出了相应的搜索方法. 为了加快搜索速度, 进一步设计了轮次索引 (round index), 并证明了该索引结构支持在动态异质信息网络上的社区搜索. Qiao 等人[40]提出了在异质信息网络上以关键词为中心的社区搜索问题, 在搜索过程中要求对称元路径的路径实例包含给定的关键词.

目前社区搜索的研究虽然取得了一定的成果, 但是针对异质信息网络的复杂条件社区搜索并未见报道. 本文在杨杰[41]工作的基础上进行扩展, 对异质信息网络的复杂条件社区搜索问题进行研究. 在异质信息网络中, 社区搜索常用的拓扑结构主要有 k-Core 和 k-Truss. 其中, k-Core 保证了每个节点都能很好地参与到社区中, 且 k-Core 具有更好的计算效率, 所以该结构使用最为广泛. 因此, 本文主要研究基于 k-Core 的变体 (k, $\mathcal{P}$)-Core[17]在异质信息网络复杂条件下的社区搜索. 当然, 其他凝聚性结构也同样适用.

2 基本概念

本节首先介绍异质信息网络的相关概念.

异质信息网络[42]. 异质信息网络通常被定义为一个由节点集$\mathcal{N}$和边集$\mathcal{E}$组成的有向图$G=\{\mathcal{N},\mathcal{E}\}$, 其中$\mathcal{N}$中的节点和$\mathcal{E}$中的边属于多种类型. 节点类型的集合和映射函数分别为$\mathcal{A}$和$\phi:\mathcal{N}\to\mathcal{A}$, 每个节点$u\in\mathcal{N}$属于$\mathcal{A}$中的一种特定类型$\phi(u)=A_u\in\mathcal{A}$, 边类型的集合和映射函数分别为$\mathcal{R}$和$\varphi:\mathcal{E}\to\mathcal{R}$, 每条边$e\in\mathcal{E}$属于$\mathcal{R}$中的一种特定类型$\varphi(e)=R_e\in\mathcal{R}$, 其中$|\mathcal{A}|+|\mathcal{R}|>2$. 节点和边具体化的网络称为网络实例.

例 1: 图 1 所示是 DBLP 的网络实例, 包含的节点类型集合$\mathcal{A}=\{A, P, V, T\}$, 节点集$\mathcal{N}=\{a_1, a_2, \ldots, a_8, p_1, p_2, \ldots, p_6, p_x, t_1, t_2, v_1, v_2\}$, 包含的边类型集合$\mathcal{R}=\{<A, P>, <P, A>, <V, P>, <P, V>, <T, P>, <P, T>\}$, 边集$\mathcal{E}=\{<a_1, p_1>, \ldots, <p_1, v_1>, \ldots, <p_1, t_1>, \ldots\}$, 由于$|\mathcal{A}|+|\mathcal{R}|>2$, 所以图 1 是一个异质信息网络.

● 网络模式[42]. 网络模式是定义在节点类型集合$\mathcal{A}$和边类型集合$\mathcal{R}$上的一个有向图, 记为$S_G=\{\mathcal{A},\mathcal{R}\}$, 图中的节点表示对象类型, 边表示关系类型. 网络模式是异质信息网络的元描述, 用于限定对象集合以及对象间关系的类型约束, 这些约束使得异质信息网络具有半结构化的特点, 引导网络语义的探究.

例 2: 图 2 所示是 DBLP 的网络模式, 表达了 4 种节点类型: 作者、论文、会议、主题和 6 种关系类型: 论文与作者之间的撰写/被撰写关系、论文与会议之间的发表/被发表关系、论文与主题之间的包含/被包含关系. 从图 2 中可以发现, 如果从节点类型 A 到节点类型 P 存在一种撰写关系 R, 则必定也存在从 P 到 A 的逆关系 R^{-1} 表示被撰写.

© 中国科学院软件研究所 http://www.jos.org.cn

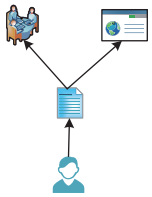

图 2 DBLP 的网络模式

● 元路径[42]. 元路径$\mathcal{P}$定义为在网络模式$S_G=\{\mathcal{A},\mathcal{R}\}$上通过边类型序列$R_1R_2\ldots R_L$连接的节点类型序列$A_1A_2\ldots A_{L+1}$, 表示为$\mathcal{P}=A_1\xrightarrow{R_1}A_2\xrightarrow{R_2}\ldots\xrightarrow{R_L}A_{L+1}$, 其中$L$表示元路径$\mathcal{P}$的长度. 当$\mathcal{P}$与其反向路径$\mathcal{P}^{-1}$相同时, 称$\mathcal{P}$为对称元路径, 否则为非对称元路径. 如果在同一对节点类型之间不存在多种类型的边, 则元路径可以简写为$A_1A_2\ldots A_{L+1}$. 在异质信息网络中, 当且仅当$\phi(u_i)=A_i\,(i\in[1,L+1])$, $\varphi(e_i)=R_i\,(i\in[1,L])$, 才能称$P=(u_1u_2\ldots u_{L+1})$为对象$x$和$y$间关于元路径$\mathcal{P}$的路径实例, 记为$P\in\mathcal{P}$, 其中$e_i=<u_i,u_{i+1}>$, $x=u_1$, $y=u_{L+1}$, 通常对象x和y间关于元路径$\mathcal{P}$的路径实例可能不止一条.

例 3: 在图 2 所示 DBLP 的网络模式中, 表示作者之间合著关系的元路径A (作者)$\xrightarrow{\text{撰写}}P$ (论文)$\xrightarrow{\text{被撰写}}A$ (作者), 由于 DBLP 中同一对节点类型之间不存在多种类型的关系, 因此可缩写为APA. 显然, APA的反向路径是其自身, 即满足$\mathcal{P}=\mathcal{P}^{-1}$, 所以称元路径$APA$为对称元路径. 图 1 所示网络中的路径$P=(a_1p_1a_2)$便是对称元路径$APA$的一条路径实例.

● 受限元路径[43]. 受限元路径定义为具有某种特定约束对象的元路径, 可以表示为$C\mathcal{P}=\mathcal{P}|C$. 其中$\mathcal{P}=A_1A_2\ldots A_{L+1}$表示元路径, C表示元路径$\mathcal{P}$中的约束对象.

例 4: 以图 2 为例, 当对元路径APA施加约束条件P="p_x", 就得到一条受限元路径$APA|P$="p_x", 该受限路径表示作者的合著关系并且合著的论文为p_x, 即论文p_x的所有作者. 类似地, 受限元路径$APTPA|T$="t_1"表示在同一主题t_1下发表论文的作者.

显然, 与元路径相比, 受限元路径在不同约束条件下细分元路径, 传递了更加丰富的语义[43].

● $\mathcal{P}$-邻居[17]. 在异质信息网络中, 可以通过元路径$\mathcal{P}$的路径实例与节点u连接的节点, 都是u的$\mathcal{P}$-邻居.

假设存在路径实例$P_1=(x_1x_2\ldots x_L)$和$P_2=(y_1y_2\ldots y_L)$, 当且仅当$x_L=y_1$或$y_L=x_1$时, 路径实例P_1和P_2是可连接的. 多条可连接的路径实例相连构成路径实例链.

● $\mathcal{P}$-相连[17]. 在异质信息网络中, 如果节点u可以通过元路径$\mathcal{P}$的路径实例链连接到节点v, 则称v是u的$\mathcal{P}$-相连节点.

● $\mathcal{P}$-连通[17]. 由异质信息网络诱导的一个子网络H, 若H中所有节点都是$\mathcal{P}$-相连的, 则称该子网络H是$\mathcal{P}$-连通的.

例 5: 在图 1 中, 当元路径$\mathcal{P}=APA$, 则a_4的$\mathcal{P}$-邻居为a_5和a_6, 此时a_4与a_5、a_6、a_7、a_8都是$\mathcal{P}$-相连的, 而$\{a_4,a_5,a_6,a_7,a_8\}$构成的子网络是$\mathcal{P}$-连通的. 同样, $\mathcal{P}$-邻居、$\mathcal{P}$-相连和$\mathcal{P}$-连通也适用于受限元路径. 当受限元路径$C\mathcal{P}=APA|P$="p_3"时, a_4的$C\mathcal{P}$邻居为a_5和a_6, 此时a_4与a_5、a_6都是$C\mathcal{P}$-相连的, 而$\{a_4,a_5,a_6\}$构成的子网络是$C\mathcal{P}$-连通的.

● 禁止节点[15]. 社区中不能出现的节点称为禁止节点.

例 6: 在投稿论文的审稿人选择场景中, 审稿人不能与投稿论文的任意一个作者存在合著关系, 即与投稿论文的任意一个作者存在合著关系的作者 (包括投稿论文的作者) 不能出现在审稿人候选名单构成的社区中, 此时这些作者就被称为禁止节点.

3 问题定义

根据异质信息网络的相关概念, 可以定义异质信息网络在简单条件下的社区搜索和复杂条件下的社区搜索.

3.1 异质信息网络中简单条件社区搜索

定义 1. 异质信息网络在简单条件下的社区搜索.

对于异质信息网络$G=\{\mathcal{N},\mathcal{E}\}$, 用户给定属于$\mathcal{N}$的查询节点$q$和对称元路径$\mathcal{P}$, 寻找至少一个满足如下条件

的节点集 H.

(1) H 属于 $\mathcal{N}$, 且 H 中节点的类型与查询节点 q 的类型相同.

(2) $G[H]$ 表示由 H 导出的关于对称元路径 $\mathcal{P}$ 的 $\mathcal{P}$-连通子图.

(3) H 包含查询节点 q.

(4) H 满足搜索问题的社区定义.

Fang 等人[17]提出的 $(k, \mathcal{P})$-Core 结构的社区搜索和 Yang 等人[18]提出的 $(k, \mathcal{P})$-Btruss 结构的社区搜索都属于异质信息网络在简单条件下的社区搜索.

3.2 异质信息网络中复杂条件社区搜索

定义 2. 异质信息网络在复杂条件下的社区搜索.

对于异质信息网络 $G=\{\mathcal{N},\mathcal{E}\}$, 用户给定属于 $\mathcal{N}$ 的查询节点 q 和元路径 $\mathcal{P}$, 寻找至少一个满足如下条件的节点集 H.

(1) H 属于 $\mathcal{N}$, 且 H 中节点的类型相同.

(2) $G[H]$ 表示由 H 导出的关于元路径 $\mathcal{P}'$ 的 $\mathcal{P}'$-连通子图 (若 $\mathcal{P}$ 是对称元路径, 则 $\mathcal{P}'=\mathcal{P}$;否则 $\mathcal{P}'=\mathcal{P}$ 补全后的元路径).

(3) H 满足定义在查询节点 q 上的搜索条件 F.

(4) H 满足搜索问题的社区定义.

当复杂条件的社区搜索问题的元路径 $\mathcal{P}$ 为对称元路径且节点集 H 包含查询节点 q 时, 则复杂条件的社区搜索问题退化为简单条件的社区搜索问题. 所以, 简单条件的社区搜索问题是复杂条件的社区搜索问题的特例.

本文需要解决以下 3 个问题.

Q1. 搜索节点类型和结果社区节点类型不一致的问题.

Q2. 搜索节点到结果社区节点的路径包含特定约束对象的问题.

Q3. 结果社区不能包含特定节点对象的问题.

所以本文考虑的复杂条件主要有非对称元路径约束 (由问题 Q1 概括得到)、受限元路径约束 (由问题 Q2 概括得到) 和禁止节点约束 (由问题 Q3 概括得到), 关于这 3 个复杂条件的社区搜索的定义如下.

定义 3. 非对称元路径约束的社区搜索.

当社区搜索问题的元路径 $\mathcal{P}$ 为非对称元路径时, 称搜索问题为非对称元路径约束的社区搜索.

定义 4. 受限元路径约束的社区搜索.

当社区搜索问题的元路径 $\mathcal{P}$ 为受限元路径时, 称搜索问题为受限元路径约束的社区搜索.

定义 5. 禁止节点约束的社区搜索.

当社区搜索问题的搜索条件 F 包含禁止节点约束时, 称搜索问题为禁止节点约束的社区搜索.

显然, 当搜索问题同时满足定义 3–5 时, 该搜索问题依旧是一个复杂条件的社区搜索问题, 比如投稿论文的审稿人选择便是符合这些特性的社区搜索问题.

4 异质信息网络中复杂条件社区搜索方法

根据异质信息网络的复杂条件社区搜索问题的定义, 本节给出了相应的解决办法, 并设计了相应的搜索算法.

4.1 非对称元路径约束的社区搜索

在异质信息网络中, 对称元路径表达了其路径实例的首尾对象之间的相似性[42], 而目标社区内节点的类型相同, 且社区表达了社区内节点之间的相似性, 所以针对非对称元路径约束的社区搜索, 一个直观的方法是首先补全非对称元路径使其成为对称元路径, 然后采用对称元路径的社区搜索方法进行社区搜索.

4.1.1 非对称元路径补全策略

将非对称元路径补全为对称元路径时, 可能会产生多条对称元路径, 然而研究表明, 元路径越长, 由元路径相

© 中国科学院软件研究所 http://www.jos.org.cn

连的节点间的相似度越低[44]. 以图 2 中的元路径 APT 为例, 当用户给定查询节点 q ($\phi(q)=A$) 和元路径 APT, 表明用户想要获取与作者 q 的研究相似的研究主题. 元路径 APT 可以补全为 $TPAPT$ 或 TPT, 其中 $TPAPT$ 表示通过不同作者的论文连接相似的研究主题, 而 TPT 表示通过不同的论文连接相似的研究主题, 对比 $TPAPT$ 和 TPT 所表达的语义信息, 可以发现 TPT 与 APT 更加吻合, 因为用户给定的查询节点 q 已经确定了节点类型 A 的值, 所以 $TPAPT$ 连接的研究主题的相似程度较 TPT 更弱. 因此补全元路径采用最短原则, 即保证补全后的元路径是所有可能的补全元路径中最短的一条, 所以 APT 补全后为 TPT.

通常, 目标社区的节点类型与元路径的最后一个节点类型相同, 所以补全元路径采用从后向前的方式. 假设需要补全的元路径 $\mathcal{P}=A_1A_2\ldots A_L$, 为了尽量保留结果社区与查询节点的相关性, 提出的补全策略为:

(1) 判断元路径 $\mathcal{P}$ 中是否有重复的节点类型 $A\in\Pi(\{A_1, A_2, \ldots, A_L\})$($\Pi$ 表示投影操作), 若有则执行 (2), 否则执行 (3).

(2) 判断元路径 $\mathcal{P}$ 是否关于节点类型 A 存在部分路径对称, 若存在则令 $\mathcal{P}$=(元路径 $\mathcal{P}$ 中从 A 的下一个节点开始向后截取的子串), 并以子串的第 1 个节点为对称点向前补全为对称元路径; 否则执行 (3).

(3) 采用最短原则将元路径 $\mathcal{P}$ 补全为对称元路径.

例 7: 对于元路径 $\mathcal{P}_1=PTPA$, 存在重复节点类型 P, 故执行 (2); 因为存在关于节点类型 P 的对称部分 PT 和 TP, 所以 $\mathcal{P}_1=TPA$ ($\mathcal{P}_1$ 中从 P 的下一个节点开始向后截取的子串), 并以子串的第 1 个节点 T 为对称点补全元路径得到 $\mathcal{P}_1=APTPA$. 对于元路径 $\mathcal{P}_2=TPV$, 不存在重复节点类型, 故执行 (3); 采用最短原则补全 $\mathcal{P}_2$ 得到 VPV. 显然, 按照元路径补全策略得到的元路径 $\mathcal{P}_1$ 和 $\mathcal{P}_2$ 与预期的一致.

当非对称元路径补全为对称元路径之后, 就可以采用对称元路径的社区搜索方法进行社区搜索. 本文将非对称元路径补全策略与 FastBCore 算法[17]结合, 提出了解决非对称元路径约束的社区搜索算法 (community search for asymmetric meta-path, CSAMP).

4.1.2 非对称元路径约束的社区搜索算法 CSAMP

非对称元路径约束的社区搜索算法 CSAMP 如算法 1 所示.

算法 1. CSAMP 算法.

输入: 异质信息网络 G, 查询节点 q, 正整数 k, 元路径 $\mathcal{P}$;
输出: 满足条件的社区 C.

1. 初始化 $C\leftarrow\varnothing$, $X_q\leftarrow\{q\}$;
2. **if** 元路径 $\mathcal{P}$ 是非对称的 **then**
3. **for** $i\leftarrow 1$ 到 L **do** //L 为元路径 $\mathcal{P}$ 的长度
4. $Y\leftarrow\varnothing$;
5. **for** X_q 中的每一个节点 u **do**
6. **for** u 的每一个邻居 v **do**
7. **if** 边<u, v>匹配元路径 $\mathcal{P}$ 的第 i 条边 **then**
8. **if** 边<u, v>的标签不为 i **then**
9. 将 v 添加到集合 Y 并将边<u, v>的标签标记为 i;
10. $X_q\leftarrow Y$;
11. $\mathcal{P}'\leftarrow\mathcal{P}$ 补全后的对称元路径;
12. **else**
13. $\mathcal{P}'\leftarrow\mathcal{P}$;
14. **return** $C\leftarrow$ FastBCore(G, X_q, $\mathcal{P}'$, k); //将原始算法仅支持输入单个查询节点 q 调整为节点集合 X_q

CSAMP 算法首先初始化社区 C 为空, 查询节点集 X_q 为 $\{q\}$(第 1 行), 然后根据元路径 $\mathcal{P}$ 是否对称来确定 X_q

© 中国科学院软件研究所 http://www.jos.org.cn

和$\mathcal{P}'$的值. 当元路径$\mathcal{P}$为非对称元路径时, 则从节点 q 开始采用 BSL 策略搜索查询节点 q 的所有$\mathcal{P}$-邻居, 并将其赋值给 X_q (第 2–10 行), 同时将$\mathcal{P}$ 补全后的对称元路径赋值给$\mathcal{P}'$ (第 11 行). 否则 $X_q=\{q\}$, $\mathcal{P}'$ 被赋值为$\mathcal{P}$ (第 12,13 行). 最后, 将 X_q 和$\mathcal{P}'$输入到 FastBCore 算法进行社区搜索并返回结果 (第 14 行). CSAMP 算法采用 BSL 策略搜索节点 q 的所有$\mathcal{P}$-邻居, 加快了搜索速度. 该算法即适用于对称元路径的社区搜索, 也适用于非对称元路径的社区搜索, 算法的时间复杂度为$\mathrm{O}\left(n_1\times d_{1,2}+n_1\sum_{i=2}^{L}n_i\times d_{i,i+1}\right)$, 其中 n_i 表示对称元路径$\mathcal{P}'$的第 i 个节点类型对应的节点数, $d_{i,i+1}$ 表示元路径$\mathcal{P}'$的第 i+1 个节点类型对应的节点连接到第 i 个节点类型的某一节点的最大节点数.

4.2 受限元路径约束的社区搜索

在 CSAMP 算法中, BSL 策略用于寻找查询节点 q 的所有$\mathcal{P}$-邻居, 根据受限元路径的定义, 为了处理受限元路径约束的社区搜索, 本节对 CSAMP 算法的 BSL 策略进行改进, 设计了解决受限元路径约束的社区搜索算法 (community search for constrained meta-path, CSCMP).

4.2.1 受限元路径约束的 BSL 策略

假设受限元路径$\mathcal{P}=A_1A_2\ldots A_L$, 约束对象节点集 O_c 表示所有约束对象构成的集合, O_i 表示 O_c 中所有节点类型为 A_i 的对象构成的集合, 其中 A_i 表示元路径$\mathcal{P}$中第 i 个位置的节点类型, $1\leqslant i\leqslant L$. 比如对于受限元路径 $PTP|T=$ "t_1", i 从 1 开始编号, 则 T 的位置是 2, 因此得到 $O_2=\{t_1\}$, 而 O_1、O_3 为空.

具体而言, 改进的 BSL 策略在判断边<u, v>与元路径$\mathcal{P}$的第 i 条边是否匹配之前, 首先判断 O_i 是否为空, 只有当 O_i 为空或者 O_i 不为空且 v 在 O_i 中时, 才能继续往下执行. 如果 O_i 不为空且 v 不在 O_i 中, 则跳过 (过程如算法 2 第 3–10 行所示).

4.2.2 受限元路径约束的社区搜索算法 CSCMP

受限元路径约束的社区搜索算法 CSCMP 如算法 2 所示.

算法 2. CSCMP 算法.

输入: 异质信息网络 G, 查询节点 q, 正整数 k, 元路径$\mathcal{P}$, 约束对象节点集 O_c;
输出: 满足条件的社区 C.

1. 初始化 $C\leftarrow\varnothing$, $X_q\leftarrow\{q\}$, $O_i\leftarrow\{O_c$ 中所有节点类型为 A_i 的对象};
2. **if** 元路径$\mathcal{P}$是非对称的 **then**
3. **for** $i\leftarrow 1$ 到 L **do** //L 为元路径$\mathcal{P}$的长度
4. $Y\leftarrow\varnothing$;
5. **for** X_q 中的每一个节点 u **do**
6. **for** u 的每一个邻居 v **do**
7. **if** O_i 为空或者 O_i 不为空且 v 在 O_i 中 **then**
8. **if** 边<u, v>匹配元路径$\mathcal{P}$的第 i 条边 **then**
9. **if** 边<u, v>的标签不为 i **then**
10. 将 v 添加到集合 Y 并将边<u, v>的标签标记为 i;
11. $X_q\leftarrow Y$;
12. $\mathcal{P}'\leftarrow\mathcal{P}$ 补全后的对称元路径;
13. **else**
14. $\mathcal{P}'\leftarrow\mathcal{P}$;
15. **return** $C\leftarrow$ FastBCore(G, X_q, $\mathcal{P}'$, k, O_c); //将原始算法仅支持输入单个查询节点 q 调整为节点集合 X_q, 加入约束对象节点集 O_c, 并调整 BSL 策略使其适应受限元路径的条件

CSCMP 算法首先初始化社区 C 为空, 查询节点集 X_q 为$\{q\}$, O_i 为集合$\{O_c$ 中所有节点类型为 A_i 的对象}(第 1 行), 然后根据元路径$\mathcal{P}$是否对称来确定 X_q 和$\mathcal{P}'$的值. 当元路径$\mathcal{P}$为非对称元路径时, 则从节点 q 开始采用调整

© 中国科学院软件研究所 http://www.jos.org.cn

后的 BSL 策略搜索查询节点 q 的所有$\mathcal{P}$-邻居, 并赋值给 X_q(第 2–11 行), 同时赋值$\mathcal{P}'$为$\mathcal{P}$补全后的对称元路径 (第 12 行). 否则 $X_q=\{q\}$, $\mathcal{P}'$被赋值为$\mathcal{P}$(第 13–14 行). 最后, 将 X_q、$\mathcal{P}'$以及 O_c 输入到调整后的 FastBCore 算法进行社区搜索并返回结果 (第 15 行). 由于 CSCMP 算法是基于 CSAMP 算法改进得到的, 所以同样适用于对称元路径的社区搜索和非对称元路径的社区搜索, 算法的时间复杂度为 $\mathrm{O}\left(n_1\times d_{1,2}+n_1\sum_{i=2}^{L}n_i\times d_{i,i+1}\right)$(当 O_i 不为空时, $n_i=O_i$).

4.3 禁止节点约束的社区搜索

本文所考虑的禁止节点约束是在受限元路径之上进行的, 为了便于描述, 将复杂条件的社区搜索问题中用户给定的查询节点 q 称为必要查询节点, 将元路径$\mathcal{P}$称为搜索元路径, 将禁止节点对应的搜索元路径称为禁止元路径.

求解禁止节点约束的社区搜索的一种直观的想法是先对必要查询节点进行社区搜索获得一个社区 (称为必要社区), 然后再检查该社区中是否存在禁止节点. 若存在禁止节点则删除, 最后检查该社区剩余的节点集是否依旧满足社区定义[15]. 该方法在异质信息网络中可能存在问题, 以图 1 所示的网络为例, 当禁止节点为 a_4 时, 根据语义环境应删除与节点 a_4 具有合著关系的所有节点, 所以只删除节点 a_4 显然是不恰当的. 因此, 针对禁止节点, 首先需要搜索禁止节点对应的社区 (称为禁止社区), 然后从必要社区中删除禁止社区包含的节点.

4.3.1 禁止节点约束的社区搜索算法 CSPNC

基于上述的思想, 提出了针对禁止节点约束的社区搜索算法 (community search for prohibited node constraint, CSPNC), 见算法 3. 由于在禁止节点约束下的条件复杂多样, 所以算法首先将搜索条件拆分为多个不可拆分的搜索条件, 然后对每个搜索条件进行社区搜索, 最后将结果融合. 例如本文一开始提出的由投稿论文安排审稿人对应的复杂搜索条件可表示为 $(p_x; PTPA|T=\text{“}t_1\text{”})\cap\neg(a_4; APA)\cap\neg(a_5; APA)$(“;”前的值 p_x 为必要查询节点, $PTPA$ 为搜索元路径, a_4 和 a_5 为禁止节点, APA 为禁止元路径), 该条件可拆分为 $(p_x; PTPA|T=\text{“}t_1\text{”})$、$(a_4; APA)$ 和 $(a_5; APA)$, 条件 $(p_x; PTPA|T=\text{“}t_1\text{”})$ 的搜索结果为 $C_1=\{a_1, a_2, a_3, a_4, a_5, a_6\}$, 条件 $(a_4; APA)$ 和 $(a_5; APA)$ 的搜索结果都为 $C_2=\{a_4, a_5, a_6, a_7, a_8\}$, 显然融合之后的结果为 $C=\{a_1, a_2, a_3\}$.

算法 3. CSPNC 算法.

输入: 异质信息网络 G, 复杂搜索条件 Q;
输出: 满足条件的社区 C.

1. 初始化 $C\leftarrow\varnothing$;
2. 将复杂条件 Q 拆分为多个不可拆分的搜索条件 Q_i;
3. **for** 每一个搜索条件 Q_i **do**
4. 　　对条件 Q_i 采用 CSCMP 算法进行社区搜索得到 C_i;
5. **return** $C\leftarrow$ 融合所有的 C_i;

CSPNC 算法首先初始化社区 C 为空, 然后将复杂搜索条件 Q 拆分为多个不可拆分的搜索条件 Q_i(第 1–2 行); 接着循环对每一个搜索条件 Q_i 采用算法 2 (CSCMP 算法) 进行社区搜索得到社区 C_i(第 3–4 行), 最后融合所有的社区 C_i 得到结果社区 C 并返回结果 (第 5 行). CSPNC 算法的时间复杂度为 $\mathrm{O}\left(m\times n_1\times d_{1,2}+m\times n_1\sum_{i=2}^{L}n_i\times d_{i,i+1}\right)$, 其中 m 表示复杂条件 Q 拆分出的搜索条件个数.

4.3.2 基于剪枝策略的社区搜索算法 CSPNC-PS

禁止节点约束的社区搜索算法 CSPNC 通常需要对一个必要查询节点进行社区搜索得到必要社区, 对一个或多个禁止节点进行社区搜索得到禁止社区, 然后将必要社区和禁止社区融合得到结果社区. 在对多个禁止节点进行社区搜索时, 有可能得到相同的禁止社区, 即禁止社区的搜索存在同一性.

定理 1. 同一性.

给定一个信息网络 G 和一个从 G 中搜索得到的社区 H, 对于任意一个查询节点 $q\in H$, 在其他搜索条件不变的情况下, 从 G 中搜索得到的结果社区都是 H 本身, 即将社区中的节点作为查询节点搜索社区存在同一性.

© 中国科学院软件研究所 http://www.jos.org.cn

证明: 假设由查询节点 a 从信息网络 G 中搜索得到社区 A, 由节点 $b\in A$ 搜索得到社区 B. 若 $A\neq B$, 则由 $(k, \mathcal{P})$-Core 的定义可知, 在信息网络 G 中, 节点 a 和 b 通过元路径的任意路径实例链都不相连, 这与由查询节点 a 搜索得到社区 A 并且节点 $b\in A$ 相矛盾, 故定理 1 得证.

根据定理 1, 假设现有两个禁止节点 a 和 b, 对 a 进行社区搜索得到 C_1, C_1 包含 b, 对 b 进行社区搜索得到 C_2, $C_2=C_1$, 此时对 b 的社区搜索就属于冗余搜索. 为了解决冗余搜索问题, 提出了相应的剪枝策略: 首先将复杂搜索条件 Q 拆分为一个必要查询节点对应的搜索条件和多个禁止节点的搜索条件. 针对多个禁止节点的搜索, 在将禁止节点作为查询节点进行社区搜索前, 需要事先判断查询节点是否存在于禁止社区中, 若存在则跳过该查询节点的搜索. 例如对于复杂搜索条件 $(p_x; PTPA|T=\text{“}t_1\text{”})\cap\neg(a_4; APA)\cap\neg(a_5; APA)$, 可拆分为必要查询节点 p_x 对应的搜索条件 $(p_x; PTPA|T=\text{“}t_1\text{”})$ 和两个禁止节点 a_4 和 a_5 的搜索条件 $(a_4; APA)$ 和 $(a_5; APA)$. 针对禁止节点 a_4 和 a_5, 首先判断当前的禁止社区是否包含 a_4, 由于当前的禁止社区为空, 所以将 a_4 作为查询节点进行社区搜索得到禁止社区 $\{a_4, a_5, a_6, a_7, a_8\}$. 接下来判断当前的禁止社区是否包含 a_5, 显然 a_5 在禁止社区中, 所以直接跳过 a_5 的社区搜索. 基于剪枝策略的社区搜索算法 (CSPNC with pruning strategy, CSPNC-PS) 如算法 4 所示.

算法 4. CSPNC-PS 算法.

输入: 异质信息网络 G, 复杂搜索条件 Q;
输出: 满足条件的社区 C.

1. 初始化 $C\leftarrow\varnothing$, $C_q\leftarrow\varnothing$, $C_f\leftarrow\varnothing$;
2. 将复杂条件 Q 拆分为一个必要查询节点 q 的搜索条件 Q_q 和多个禁止节点的搜索条件 Q_i; //Q_i 表示第 i 个禁止节点的搜索条件
3. $C_q\leftarrow$ 对搜索条件 Q_q 采用 CSCMP 算法进行社区搜索;
4. **for** 每一个搜索条件 Q_i **do**
5. **if** Q_i 的查询节点 $q_i\notin C_f$ **then**
6. 对搜索条件 Q_i 采用 CSCMP 算法进行社区搜索得到 C_i 并合并到 C_f;
7. **return** $C\leftarrow C_q\backslash C_f$;

CSPNC-PS 算法首先初始化社区 C、必要社区 C_q 和禁止社区 C_f 为空 (第 1 行), 然后将复杂条件 Q 拆分为一个必要查询节点 q 的搜索条件 Q_q 和多个禁止节点的搜索条件 Q_i (第 2 行), 并对搜索条件 Q_q 采用算法 2 (CSCMP 算法) 进行社区搜索得到必要社区 C_q (第 3 行), 接着循环对每一个搜索条件 Q_i 采用算法 2 进行社区搜索 (第 4–6 行). 为了避免冗余搜索, 在对搜索条件 Q_i 进行社区搜索之前, 算法需要事先判断 Q_i 的查询节点 q_i 是否已经存在于禁止社区 C_f 中 (第 5 行), 若已存在, 则跳过当前的搜索条件, 否则采用算法 2 对当前的搜索条件进行社区搜索得到社区 C_i 并合并到禁止社区 C_f 中 (第 6 行). 最后从必要社区 C_q 中剔除禁止社区 C_f 得到结果社区 C 并返回结果 (第 7 行). 在 CSPNC-PS 算法中, 当搜索条件 Q_i 的查询节点 q_i 始终不在禁止社区 C_f 中, 即禁止节点之间不存在连通关系, 算法的运行效率最低, 在最坏情况下的时间复杂度为 $\mathrm{O}\left(m\times n_1\times d_{1,2}+m\times n_1\sum_{i=2}^{L}n_i\times d_{i,i+1}\right)$, 此时 CSPNC-PS 算法的时间复杂度与 CSPNC 算法相同.

4.3.3 基于近似策略的社区搜索算法 CSPNC-AS

为了进一步提高 CSPNC 算法的搜索效率, 针对禁止节点约束的搜索条件, 提出了获取近似禁止社区的方法, 即将禁止节点的所有 $\mathcal{P}$-相连的节点中满足至少存在 k 个 $\mathcal{P}$-邻居的节点构成近似禁止社区. 显然, 近似禁止社区与通过 $(k, \mathcal{P})$-Core 得到的禁止社区之间存在包含性.

定理 2. 包含性.

给定一个信息网络 G 和一个查询节点 q, 假设由节点 q 根据 $(k, \mathcal{P})$-Core 得到禁止社区 H, 由节点 q 得到近似禁止社区 H', 则必定存在 $H\subseteq H'$.

证明: 假设节点 q 的所有 $\mathcal{P}$-相连的节点构成集合 S_1, S_1 中 $\mathcal{P}$-邻居的节点数小于 k 的节点构成集合 S_2, 则由节

© 中国科学院软件研究所 http://www.jos.org.cn

点 q 根据 $(k, \mathcal{P})$-Core 得到禁止社区 $H=S_1-S_2-S_3$, S_3 表示根据 $(k, \mathcal{P})$-Core 进行社区搜索时迭代地删除 S_1 中 $\mathcal{P}$-邻居的节点数小于 k 的节点时所删除的不在 S_2 中的其他节点构成的集合, $|S_3|\geqslant 0$; 由节点 q 得到近似禁止社区 $H'=S_1-S_2$. 显然, $H=H'-S_3$, 所以必定存在 $H\subseteq H'$ (当$|S_3|=0$ 时, 存在 $H=H'$).

根据定理 2 可知, 由禁止节点的搜索条件得到的禁止社区是近似策略得到的近似禁止社区的真子集, 因此近似算法最终所得社区是 CSPNC 算法所得社区的真子集, 这在一定程度上保证了结果的准确性. 基于近似策略的社区搜索算法 (CSPNC with approximate strategy, CSPNC-AS) 如算法 5 所示.

算法 5. CSPNC-AS 算法.

输入: 异质信息网络 G, 复杂搜索条件 Q;
输出: 满足条件的社区 C.

1. 初始化 $C\leftarrow\varnothing$, $C_q\leftarrow\varnothing$, $C_f\leftarrow\varnothing$;
2. 将复杂条件 Q 拆分为一个必要查询节点 q 的搜索条件 Q_q 和多个禁止节点的搜索条件 Q_i; //Q_i 表示第 i 个禁止节点的搜索条件
3. $C_q\leftarrow$ 对搜索条件 Q_q 采用 CSCMP 算法进行社区搜索;
4. **for** 每一个搜索条件 Q_i **do**
5. **if** Q_i 的查询节点 $q_i\notin C_f$ **then**
6. 将节点 q_i 的 $\mathcal{P}$-相连的节点中满足至少存在 k 个 $\mathcal{P}$-邻居的节点合并到 C_f;
7. **return** $C\leftarrow C_q\backslash C_f$;

CSPNC-AS 算法第 6 行根据近似策略将节点 q_i 的 $\mathcal{P}$-相连的节点中满足至少存在 k 个 $\mathcal{P}$-邻居的节点直接合并到禁止社区 C_f, 而其他步骤与 CSPNC-PS 算法类似. CSPNC-AS 算法对搜索条件 Q_q 采用 CSCMP 算法进行社区搜索的时间复杂度为 $\mathrm{O}\left(n_1\times d_{1,2}+n_1\sum_{i=2}^{L}n_i\times d_{i,i+1}\right)$; 当搜索条件 Q_i 的查询节点 q_i 始终不在禁止社区 C_f 中, 最坏情况下寻找所有禁止节点的 $\mathcal{P}$-相连节点中满足至少存在 k 个 $\mathcal{P}$-邻居节点的时间复杂度为 $\mathrm{O}\left((m-1)\times d_{1,2}+(m-1)\times\sum_{i=2}^{L}n_i\times d_{i,i+1}\right)$, 其中 $m-1$ 表示禁止节点的搜索条件个数. 所以在最坏情况下, CSPNC-AS 算法总的时间复杂度为 $\mathrm{O}\left(n_1\times d_{1,2}+n_1\sum_{i=2}^{L}n_i\times d_{i,i+1}+(m-1)\times d_{1,2}+(m-1)\times\sum_{i=2}^{L}n_i\times d_{i,i+1}\right)$.

5 实验评估

为了验证各个复杂条件的社区搜索算法的有效性和效率, 本节采用多种评价指标在 4 个真实的数据集上进行社区搜索实验. 首先在 S-DBLP 数据集上评估了 CSAMP 和 CSCMP 算法的搜索结果, 其次在其余 3 个数据集上比较了 CSAMP、CSCMP、CSPNC、CSPNC-PS 和 CSPNC-AS 算法的搜索结果的直径、密度、PathSim 和 HeteSim, 接着在 S-DBLP 数据集上比较了不同算法搜索结果的 F_1 分数值, 并展示了 CSPNC-AS 算法的平均相对误差 (mean relative error, *MRE*), 然后研究了参数变化和数据集大小对算法运行时间的影响, 最后给出了一个针对本文所提问题的应用案例.

5.1 实验环境和数据集

本文实验所用的软硬件环境: 操作系统 Ubuntu 18.04, 处理器 Intel(R) Core(TM) i7-7820X CPU @ 3.60 GHz, 内存 64 GB.

本文实验使用了 4 个真实的数据集, 相关统计信息如表 1 所示. Foursquare 是由用户在美国的 34 713 条签到记录 (record) 组成, 记录包含城市 (city)、类别 (category)、用户 (user) 和兴趣点 (POI), 其中记录类型的节点与其他 4 种类型的节点之间有边. DBLP 包括计算机领域的出版记录, 拥有 239 679 名作者 (author)、398 472 篇论文 (paper)、2 558 个会议或期刊 (venue) 和 42 020 种主题 (topic), 其中论文与作者、会议、主题之间有边. IMDB 由自 2000 年以来的电影评价记录组成, 包含演员 (actor)、导演 (director)、作家 (writer) 和电影 (movie) 这 4 种类型,

© 中国科学院软件研究所 http://www.jos.org.cn

其中电影与其他 3 种类型的节点之间有边. S-DBLP 是从 DBLP 中抽取的包含真实社区的小数据集, 包括数据库、数据挖掘、信息检索和人工智能这 4 个领域的 14 475 名作者、14 276 篇论文、20 个会议或期刊和 8 920 种主题. 针对每个数据集, 为了与现有研究一致[17], 本文收集了一组对称元路径, 元路径的种类数如表 1 的最后一列所示, 这些元路径的最大长度为 4.

表 1 实验所使用的数据集

数据集	节点数	边数	节点种类	边种类	元路径种类
Foursquare	43 199	405 476	5	4	52
DBLP	682 729	1 951 209	4	3	30
IMDB	4 467 806	7 597 591	4	3	30
S-DBLP	37 691	341 588	4	3	30

为了进行相应的实验评估, 本文为每个数据集生成了 200 条搜索记录, 每一条记录的格式为: {必要查询节点, 搜索元路径, 约束对象节点集, k_q, 禁止节点集, 禁止元路径, k_f}, 其中 k_q 是必要社区的 k 值, 默认取值为 6, 禁止节点集包含 3 个节点, k_f 是禁止社区的 k 值, 默认取值也为 6. 在后续的实验中, 除了特别说明之外, 实验结果默认都是这 200 条搜索记录的平均值. 为了生成一条搜索记录, 本文首先随机选择一条对称元路径, 删除起始节点得到一条非对称元路径作为搜索元路径. 然后选择一个节点类型与搜索元路径起始节点相同且度大于等于 6 的节点作为必要查询节点, 以此保证结果社区是有意义的. 最后再根据必要查询节点和搜索元路径生成约束对象节点集、禁止节点集和禁止元路径. 针对图 1 的网络, 假设现有对称元路径 $APTPA$, 删除其起始节点 A 得到非对称元路径 $PTPA$ 作为搜索元路径, 然后寻找一个 P 类型且度大于等于 6 的节点 p_x(假设 p_x 的度大于等于 6) 作为必要查询节点, 再将 p_x 的主题 t_1 作为搜索元路径的约束, 最后将 p_x 的所有作者 a_4、a_5 作为禁止节点, 将对称元路径 APA 作为禁止元路径, 形成一条搜索记录$\{p_x, PTPA, \{t_1\}, k_q, \{a_4, a_5\}, APA, k_f\}$.

5.2 算法的有效性评估

5.2.1 CSAMP 和 CSCMP 算法的有效性

由于 S-DBLP 包含真实存在的社区, 所以首先基于 S-DBLP 在不同的搜索条件下得到的搜索结果评估 CSAMP 和 CSCMP 算法的有效性, 对比算法为 FastBCore[17]和 Basic-*kKP*-core[40]及其变体 Basic-*kKP*-core*、Basic-*kKP*-core**. 其中 Basic-*kKP*-core 是以关键词为中心的社区搜索算法, 仅支持对称元路径的搜索条件, 在非对称元路径的搜索条件下, 有无关键词对结果并无影响, 所以实验中输入的关键词为空; Basic-*kKP*-core*表示支持对称元路径且关键词集合为空的搜索条件; Basic-*kKP*-core**表示支持对称元路径且关键词集合不为空的搜索条件, 为了与 CSCMP 算法进行对比, 实验中输入的关键词与 CSCMP 的约束对象相同. 6 个算法的社区搜索结果如表 2 所示, 每一行的内容表示在相同的元路径$\mathcal{P}$条件下的搜索结果 (Basic-*kKP*-core*和 Basic-*kKP*-core**表示在元路径$\mathcal{P}$补全后的对称元路径下的搜索结果), 在不同的元路径条件下查询节点 q 和正整数 k 相同, 即 q="论文 High Performance Index Build Algorithms for Intranet Search Engines", k=10. 从搜索结果可以看出, 在相同的元路径$\mathcal{P}$条件下, FastBCore 和 Basic-*kKP*-core 算法对应的结果社区为空, 因为 FastBCore 和 Basic-*kKP*-core 不支持非对称元路径的搜索条件; Basic-*kKP*-core*算法对应的搜索结果始终包含 CSAMP 算法的搜索结果, Basic-*kKP*-core**算法的搜索结果也始终包含 CSCMP 算法的搜索结果, 这是因为 Basic-*kKP*-core*和 Basic-*kKP*-core**找出的结果社区仅需满足任意节点与社区中其他节点存在$\mathcal{P}$-邻居关系的路径实例数大于等于 k, 而 CSAMP 和 CSCMP 找出的结果社区需要满足任意节点与社区中其他节点存在$\mathcal{P}$-邻居关系的节点数大于等于 k, 较 Basic-*kKP*-core*和 Basic-*kKP*-core**的条件更为严格, 所以在同等条件下, CSAMP 和 CSCMP 的搜索结果一定会分别出现在 Basic-*kKP*-core*和 Basic-*kKP*-core**的搜索结果中, 该结论同时也验证了 CSAMP 和 CSCMP 算法的有效性; CSCMP 算法由于支持约束条件, 所以得到的结果较 CSAMP 算法搜索到的节点更少, 并且该结果是 CSAMP 算法所得结果的子集. 以元路径$\mathcal{P}$=$PTPA$ 为例, CSAMP 由于没有主题约束, 结果社区包含 14 464 名作者 (数据集中总共包含 14 475

© 中国科学院软件研究所 http://www.jos.org.cn

名作者), 而 CSCMP 由于存在 T="Engines"的约束条件, 得到的结果社区仅包含 87 名作者, 该结果社区可以很好地为用户提供便利, 极大地减少了用户选择的负担. 同样, 在元路径$\mathcal{P}=PTPV$ 条件下, CSAMP 得到的结果社区包含 20 个会议 (数据集中总共包含 20 个会议), 除了包括数据库和数据挖掘领域的会议外, 还包括 AAAI 和 CVPR 等人工智能和计算机视觉领域的会议, 而 CSCMP 得到的结果社区仅包含 11 个会议, 主要集中在数据库和数据挖掘领域, 从中可以发现 CSCMP 得到的结果社区更为合理.

表 2 FastBCore、Basic-*kKP*-core、Basic-*kKP*-core*、Basic-*kKP*-core**、CSAMP 和 CSCMP 算法的社区搜索结果

元路径	FastBCore	Basic-*kKP*-core	Basic-*kKP*-core*	CSAMP	Basic-*kKP*-core**	CSCMP
$\mathcal{P}=PTPA$	—	—	George D. Hadden, Kenneth Bloom, Jun Xu, Paul Brown, Barry Smyth, Nivio Ziviani, Kristie Seymore等14 466人	Jun Xu, Paul Brown, Barry Smyth, Nivio Ziviani, Kristie Seymore, Enrico Zimuel, Ying Zhang等14 464人	{"Engines"} Barry Smyth, Nivio Ziviani, Kristie Seymore, Enrico Zimuel, Ying Zhang等87人	T="Engines" Barry Smyth, Nivio Ziviani, Kristie Seymore, Enrico Zimuel, Ying Zhang等87人
$\mathcal{P}=PVPA$	—	—	Amal Perera, Masum Serazi, Enrico Zimuel, Ying Zhang, Xiang Zhang, Stefano Lonardi, Jorge Lobo等14 475人	Amal Perera, Masum Serazi, Enrico Zimuel, Ying Zhang, Xiang Zhang, Stefano Lonardi, Jorge Lobo等14 475人	{"VLDB"} Enrico Zimuel, Ying Zhang, Xiang Zhang, Stefano Lonardi, Jorge Lobo等2 507人	V="VLDB" Enrico Zimuel, Ying Zhang, Xiang Zhang, Stefano Lonardi, Jorge Lobo等2 507人
$\mathcal{P}=PTPV$	—	—	AAAI, CVPR, VLDB, SIGMOD, KDD, ICDE, CIKM等20个	AAAI, CVPR, VLDB, SIGMOD, KDD, ICDE, CIKM等20个	{"Engines"} VLDB, SIGMOD, KDD, ICDE, CIKM等11个	T="Engines" VLDB, SIGMOD, KDD, ICDE, CIKM等11个

5.2.2 CSPNC、CSPNC-PS 和 CSPNC-AS 算法的有效性

接下来通过不同的社区评价指标来评估 CSPNC、CSPNC-PS 和 CSPNC-AS 算法的有效性, 选用的对比算法为 CSAMP、CSCMP, 默认情况下对比算法的 k 值与其他 3 种算法的 k_q 值相等.

直径: 首先使用直径评估 3 个算法所得社区的紧密度. 直径是同质信息网络中常用的社区紧密度评价指标, 描述了社区内任意一对节点之间的最大最短距离. 为了适应异质信息网络中的社区, 本文重新定义距离为连接两个实例的元路径实例链的实例数[17], 其中元路径为补全后的对称元路径. 直径的值越小, 紧密度越高. 图 3 所示是 5 种算法在不同数据集上通过对 200 条搜索记录进行社区搜索所得结果社区的平均直径. 其中 CSAMP 所得社区的节点数较 CSCMP 更多, 在计算社区内任意节点对之间的最大最短距离时能够得到更大的值, 所以 CSAMP 所得社区的直径最大. 另外, 对比 CSCMP 与 CSPNC、CSPNC-PS、CSPNC-AS 这 3 种算法的结果, 发现 CSCMP 所得社区的直径略小于后 3 种算法, 这是因为后 3 种算法剔除了禁止社区节点, 使得结果社区中连接两个实例的元路径实例链的实例数变大, 导致最终得到结果社区的直径较 CSCMP 更大.

密度: 接着使用密度评估 3 个算法所得的社区结果. 通常, 社区的密度定义为边数量除以节点数量, 但是异质信息网络中社区的节点间并没有直接的边相连, 所以为了适应异质信息网络, 本文将密度重新定义为社区中满足$\mathcal{P}$-连通的节点对数量除以所有节点数量[17], 其中$\mathcal{P}$为补全后的对称元路径. 图 4 所示是 5 种算法在不同数据集上通过对 200 条搜索记录进行社区搜索所得结果社区的平均密度, 显然社区的直径越大对应的密度越小. 对比 CSAMP、CSCMP 两种算法与 CSPNC、CSPNC-PS、CSPNC-AS 这 3 种算法的结果, 发现后 3 种算法所得社区的密度始终小于前两种算法, 这是因为后 3 种算法剔除了密度较高的禁止社区节点, 使得在计算结果社区密度的过程中得到$\mathcal{P}$-连通的节点对数量大大减少, 导致最终得到了较低的密度, 该结果与本文所研究的问题背景相吻合, 即审稿人之间的关系不会太过于紧密.

PathSim: 然后为了评价社区内节点间的相似性, 由于社区内节点间是$\mathcal{P}$-连通的, 所以采用 Sun 等人[44]提出

© 中国科学院软件研究所 http://www.jos.org.cn

的 PathSim 方法进行评估. PathSim 值越接近 1, 节点间的相似度越高. 图 5 所示是各个算法在不同数据集上通过对 200 条搜索记录进行社区搜索所得结果社区的平均 PathSim 值, 其中 CSAMP、CSCMP 两种算法所得社区的 PathSim 值始终小于 CSPNC、CSPNC-PS、CSPNC-AS 这 3 种算法, 这是因为后 3 种算法的结果社区是在更为严格的条件下得到的, 较前两种算法节点间的相似度更高. CSPNC 和 CSPNC-PS 所得社区的相似度相同, 而 CSPNC-AS 所得社区的相似度略高, 但是实际上相差不大, 这说明 CSPNC-AS 的近似策略能够在一定程度上保证结果的准确性.

HeteSim: 为了评价查询节点与社区内节点间的相似性, 由于搜索元路径是非对称的, 所以采用 Shi 等人[45]提出的 HeteSim 方法进行评估. HeteSim 值越接近 1, 相似度越高. 图 6 所示是各个算法在不同数据集上通过对 200 条搜索记录进行社区搜索所得结果社区的平均 HeteSim 值, 其中 CSAMP、CSCMP 两种算法所得社区的 HeteSim 值大于 CSPNC、CSPNC-PS、CSPNC-AS 这 3 种算法, 这是因为后 3 种算法剔除了与查询节点相似度高的禁止社区节点, 导致结果社区的 HeteSim 值较小, 这种情况显然是合理的. 因为根据本文所研究问题的背景, 结果社区与查询节点之间的相似性应该弱, 即投稿论文与审稿人之间仅需在研究主题上相似, 而不应该在其他诸如合著等方面有关联关系.

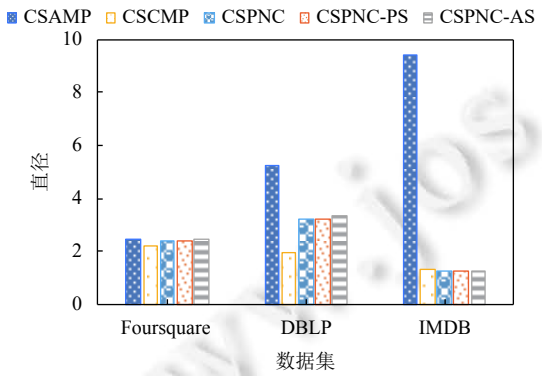

图 3 平均直径

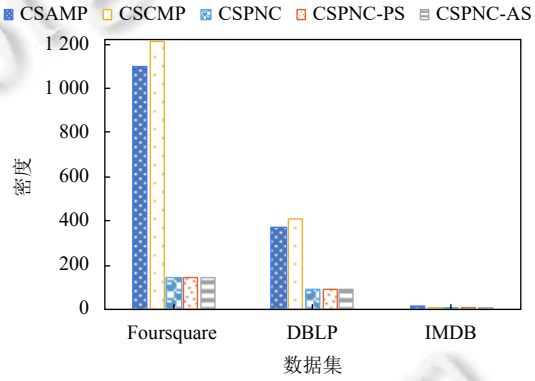

图 4 平均密度

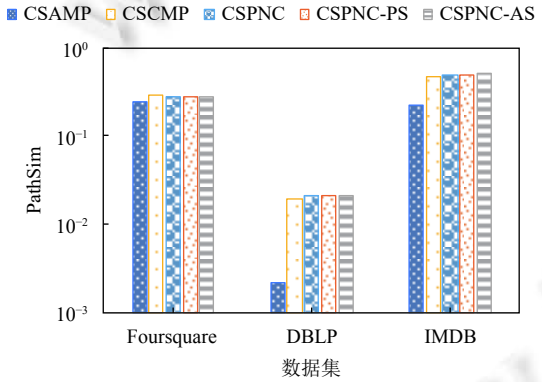

图 5 平均 PathSim 值

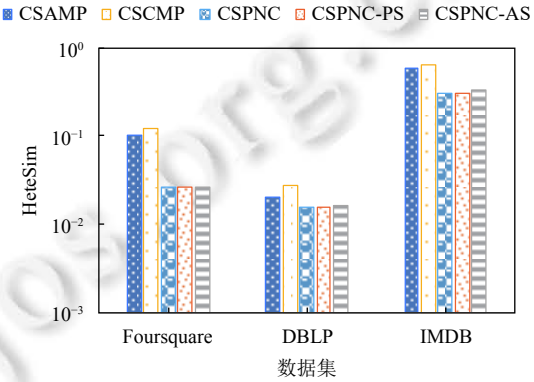

图 6 平均 HeteSim 值

F_1 分数: 最后采用 F_1 分数评估 5 个算法所得社区的精确度. 在 S-DBLP 数据集中, 根据作者发表的论文所属领域, 将作者和论文划分为 4 个社区. 对于每一篇论文, 分别用 CSAMP、CSCMP、CSPNC、CSPNC-PS 和 CSPNC-AS 进行社区搜索, 然后计算社区的 F_1 分数. 为了增加不同参数设置下社区的 F_1 分数的区分度, 参数设置如表 3 所示. 社区搜索的条件为: 必要查询节点为每一篇论文, 搜索元路径为 *PTPA*, 禁止节点为论文的作者, 禁止元路径为 *APA*, 计算 F_1 分数时将必要查询节点所属社区作为真实社区. 从表 3 的结果可以看出, 在 k_q=50 的条件

© 中国科学院软件研究所 http://www.jos.org.cn

下, 随着 k_f 的增大, CSAMP 和 CSCMP 两个算法所得社区的 F_1 分数不变, 因为这两个算法并不受 k_f 的影响, 而 CSPNC、CSPNC-PS、CSPNC-AS 所得社区的 F_1 分数变化不明显, 且 CSPNC-AS 对应的 F_1 分数与 CSPNC、CSPNC-PS 相差不大, 说明 CSPNC-AS 采用的近似策略保证了结果的准确性; 在 k_f=50 的条件下, 随着 k_q 的增大, CSAMP 所得社区的 F_1 分数有略微减小, 而其他 4 种算法所得社区的 F_1 分数有所增大. 在每一组参数设置下, CSAMP 对应的 F_1 分数始终大于其他 4 种算法, 这是因为 CSAMP 搜索得到的社区的节点比其他 4 种算法多得多, 而 CSCMP 对应的 F_1 分数始终都大于 CSPNC、CSPNC-PS 和 CSPNC-AS 这 3 种算法, 这是因为这 3 种算法的搜索条件更为严格, 得到的社区的节点数更小.

平均相对误差 *MRE*: 另外, 为了评估基于近似策略的社区搜索算法 CSPNC-AS 的近似效果, 本文首先在 S-DBLP 数据集中采用精确算法 CSPNC-PS 和近似算法 CSPNC-AS 进行社区搜索, 然后根据两个算法的搜索结果计算平均相对误差. 其中, 社区搜索的条件为: 必要查询节点为每一篇论文, 搜索元路径为 *PTPA*, 禁止节点为论文的作者, 禁止元路径为 *APA*, 实验参数设置与 F_1 分数的实验相同. 在每一组参数设置下, CSPNC-PS 和 CSPNC-AS 对 S-DBLP 数据集中的每一篇论文进行社区搜索, 然后根据公式 $MRE=\left(\sum_{i=1}^{n}\left|C_i^2-C_i^1\right|/\left|C_i^1\right|\right)/n$ 计算对应的平均相对误差. 其中 C_i^1、C_i^2 分别表示 CSPNC-PS、CSPNC-AS 基于第 i 篇论文的社区搜索结果, n 表示论文的数量. 不同参数设置下的平均相对误差如表 4 所示, 由表中的结果可知, CSPNC-AS 的平均相对误差较小, 特别是在参数较大的情况下, 实验结果表明 CSPNC-AS 所使用的近似策略是有效的.

表 3 5 种算法在不同参数设置下所得社区的 F_1 分数

k_q	k_f	CSAMP	CSCMP	CSPNC	CSPNC-PS	CSPNC-AS
50	10	0.5152	0.3943	0.3941	0.3941	0.3695
50	50	0.5152	0.3943	0.3940	0.3940	0.3939
50	100	0.5152	0.3943	0.3940	0.3940	0.3940
50	150	0.5152	0.3943	0.3940	0.3940	0.3940
50	200	0.5152	0.3943	0.3940	0.3940	0.3940
10	50	0.5158	0.3943	0.3940	0.3940	0.3939
50	50	0.5152	0.3943	0.3940	0.3940	0.3939
100	50	0.5147	0.4039	0.4035	0.4035	0.4035
150	50	0.5145	0.4038	0.4035	0.4035	0.4034
200	50	0.5140	0.4230	0.4227	0.4227	0.4227

表 4 S-DBLP 数据集上 CSPNC-AS 算法的平均相对误差 *MRE*

k_q	k_f	*MRE* (%)
50	10	13.6548
50	50	0.0558
50	100	0
50	150	0
50	200	0
10	50	0.0558
50	50	0.0558
100	50	0.0564
150	50	0.0564
200	50	0.0577

5.3 算法的效率评估

5.3.1 CSAMP 和 CSCMP 算法的效率

首先在 DBLP 数据集上评估了 CSAMP 和 CSCMP 算法在不同正整数 k 以及不同比例的数据集下的运行时间, 其中正整数 k 是算法用于判断社区中节点的 $\mathcal{P}$-邻居个数是否满足社区结构的条件. FastBCore 是基准对比算法, 由于 FastBCore 只支持对称元路径的社区搜索, 所以搜索元路径为非对称元路径的补全元路径. 在图 7 所示的结果中, 3 个算法的运行时间都随着正整数 k 的增大而增多, 这是因为随着 k 的增大, 算法需要耗费更多的时间去寻找 k 个 $\mathcal{P}$-邻居. CSCMP 算法的运行时间始终低于其余两个算法, 这是因为元路径的约束对象使得 CSCMP 只需处理更少的节点. CSAMP 算法由于需要补全元路径, 所以耗费的时间比 FastBCore 更多. 在图 8 所示的结果中, 3 个算法的运行时间随着数据集比例的增加而增多, 这是因为需要处理更多的节点. CSCMP 耗费的运行时间最少, 原因与图 7 的分析相同. CSAMP 算法耗费的时间比 FastBCore 更多, 原因也与图 7 的分析相同.

5.3.2 CSPNC、CSPNC-PS 和 CSPNC-AS 算法的效率

为了评估 CSPNC、CSPNC-PS 和 CSPNC-AS 算法的运行效率, 分别在 Foursquare、DBLP 和 IMDB 数据集上进行了相应的实验. 图 9–图 11 所示的结果分别表示 Foursquare、DBLP 和 IMDB 数据集在 k_q 和 k_f 取值变化的

© 中国科学院软件研究所 http://www.jos.org.cn

情况下 3 个算法的运行时间. 从结果中可以看出, CSPNC-AS 的运行时间始终是最少的, CSPNC-PS 次之, CSPNC 最差. 这是因为 CSPNC-PS 和 CSPNC-AS 都在 CSPNC 基础上进行了优化, 并且 CSPNC-AS 在 CSPNC-PS 基础上还加入了获取近似禁止社区的方法, 所以 CSPNC-AS 的运行效率是最优的. 在结果中, 同样可以发现 k_q 和 k_f 的取值对 3 个算法运行时间的影响并不明显, 原因是数据集中节点的度本身就比较大.

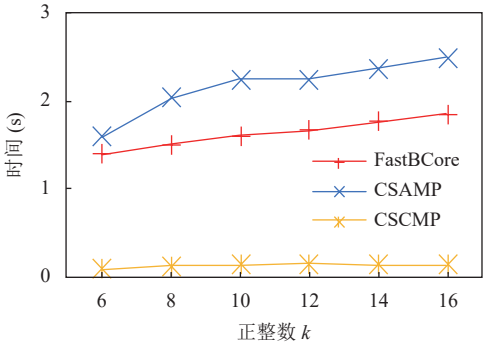

图 7　正整数 k 对算法运行时间的影响

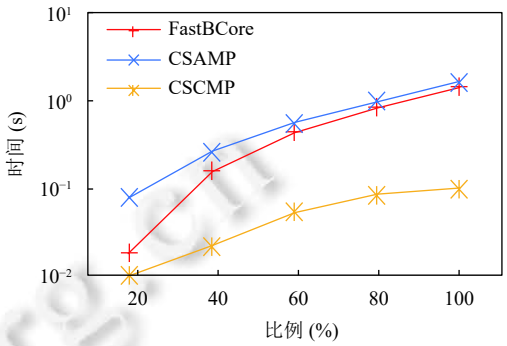

图 8　数据集大小对算法运行时间的影响

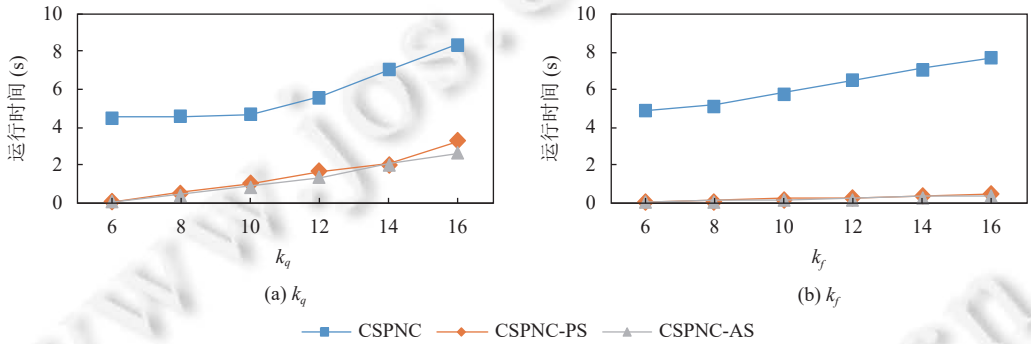

图 9　k_q 和 k_f 对算法运行时间的影响 (Foursquare 数据集)

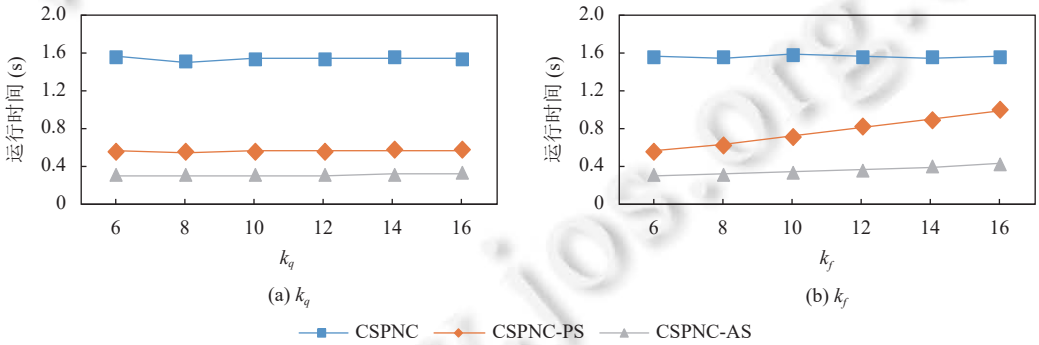

图 10　k_q 和 k_f 对算法运行时间的影响 (DBLP 数据集)

图 12–图 14 所示的结果分别是不同比例的数据集对算法运行时间的影响, 从结果中可以看出, 算法的运行时间随着数据集比例的增加而增多, 这是因为算法需要处理更多的节点, 其中 CSPNC-AS 的运行时间始终是最优的, CSPNC-PS 次之, CSPNC 最差.

5.4　应用案例

投稿论文的审稿人选择场景具有实际的应用价值, 为了说明本文所提出的异质信息网络的复杂条件社区搜索

© 中国科学院软件研究所 http://www.jos.org.cn

方法能够很好地解决该问题, 本节给出一个在真实数据集 S-DBLP 上进行论文的审稿人选择的应用案例.

假设某期刊编辑部收到一篇论文“Space-efficient Relative Error Order Sketch over Data Streams” (简称为“Space-efficient”), 论文的主题有 Data 和 Streams 等, 作者为 Ying Zhang、Xuemin Lin、Jian Xu、Flip Korn 和 Wei Wang. 为了对该论文进行同行评审, 编辑部需要为该论文寻找审稿人, 并且审稿人需要满足两个条件: 第一, 审稿人从事的研究主题必须至少与论文“Space-efficient”的一个主题相关; 第二, 审稿人不能与论文“Space-efficient”的任意一个作者存在合著关系.

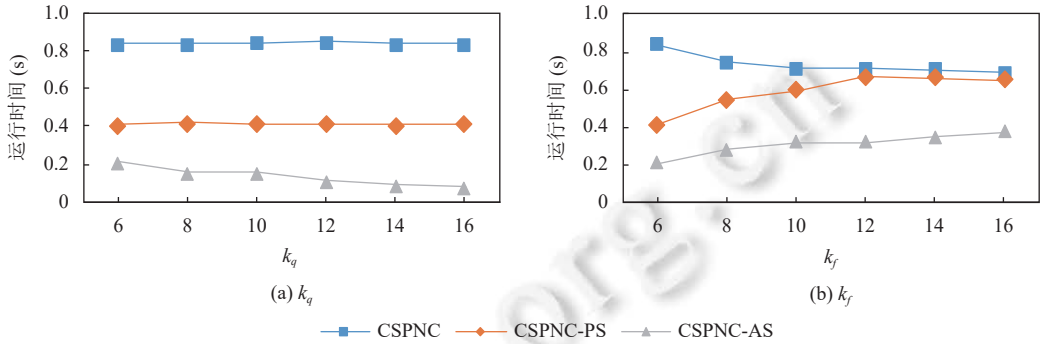

图 11 k_q 和 k_f 对算法运行时间的影响 (IMDB 数据集)

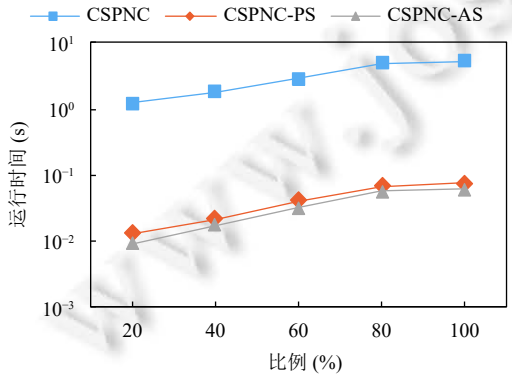

图 12 Foursquare 数据集大小对算法运行时间的影响

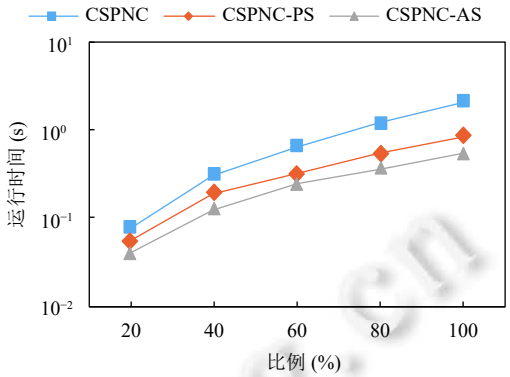

图 13 DBLP 数据集大小对算法运行时间的影响

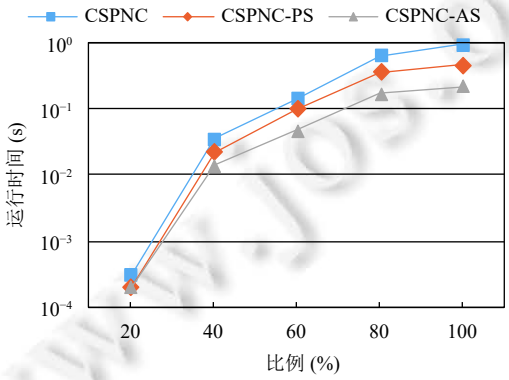

图 14 IMDB 数据集大小对算法运行时间的影响

基于上述的案例描述, 可以采用禁止节点约束的社区搜索方法在 S-DBLP 数据集上搜索作者作为审稿人候选名单供期刊编辑部选择. 该搜索过程的必要查询节点是“论文 Space-efficient”; 因为目标社区节点类型是作者类型

© 中国科学院软件研究所 http://www.jos.org.cn

并且存在主题类型的约束, 所以搜索元路径是 *PTPA*, 约束对象节点集是{Data, Streams}; 由于审稿人候选名单中的作者不能与论文“Space-efficient”的作者存在合著关系, 故禁止节点集是{Ying Zhang, Xuemin Lin, Jian Xu, Flip Korn, Wei Wang}, 禁止元路径是 *APA*; 此外, 为了确保所得社区具有高内聚性, 设置 k_q=10, 为了尽可能地排除与禁止节点有关联的节点, 设置 k_f=1; 因此形成一条搜索记录{论文 Space-efficient, *PTPA*, {Data, Streams}, 10, {Ying Zhang, Xuemin Lin, Jian Xu, Flip Korn, Wei Wang}, *APA*, 1}.

此外, 本节还设置了一个基线方法作为对比. 基线方法的具体操作过程为: 首先利用 FastBCore 算法通过投稿论文“Space-efficient”和元路径 *PTP* 获得与投稿论文的主题存在相似关系的论文节点集 S_1, 再根据 S_1 中的论文找出所有的作者形成集合 S_2, 然后利用 FastBCore 算法通过投稿论文的作者{Ying Zhang, Xuemin Lin, Jian Xu, Flip Korn, Wei Wang}和元路径 *APA* 获得与投稿论文的作者存在合著关系的作者节点集 S_3, 最后从 S_2 中删除存在于 S_3 中的节点, S_2 中剩余的节点即为审稿人候选名单.

本文所提方法和基线方法的搜索结果如表 5 所示.

表 5 应用案例的搜索结果

约束对象	CSPNC	基线方法
T=“Data”	Jonathan Dinerstein, Lien-Chin Chen, Michael Griebel, Choh-Man Teng等339名候选人	Izchak Sharfman, Assaf Schuster, Daniel Keren, Shen-Shyang Ho, Harry Wechsler等2 471名候选人
T=“Stream”	William John Teahan, Shen-Shyang Ho, Man Hon Wong, Mi Zhou等14名候选人	
T={“Data”, “Stream”}	共344名候选人	

由表 5 可知, 基线方法获得的审稿人候选名单的人数远远大于本文所提出的方法得到的结果, 这是因为基线方法没有考虑主题约束, 将许多与投稿论文主题无关的节点都添加了进来, 导致基线方法所得的审稿人候选名单的人数较多, 可靠性较低. 通过对本文所提出的方法获得的审稿人候选名单进行追踪发现, 他们与论文“Space-efficient”的作者都不存在合著关系. 主题约束 *T*=“Stream”的 14 名候选人及其与主题相关的论文信息如表 6 所示, 从表中可知, 这 14 名候选人是 6 篇研究主题为“Stream”的论文的作者, 表明本文所提出的方法在解决投稿论文的审稿人选择问题上是可靠的.

表 6 14 名候选审稿人及其论文信息

论文	作者
Evaluation of Summarization Schemes for Learning in **Streams**	Alec Pawling, Nitesh Vijay Chawla, Amitabh Chaudhary
Efficient Online Subsequence Searching in Data **Streams** under Dynamic Time Warping Distance	Mi Zhou, Man Hon Wong
Detecting Changes in Unlabeled Data **Streams** Using Martingale	Shen-Shyang Ho, Harry Wechsler
Text Categorization for **Streams**	D. L. Thomas, William John Teahan
Finding Maximal Frequent Itemsets over Online Data **Streams** Adaptively	Daesu Lee, Wonsuk Lee
A Geometric Approach to Monitoring Threshold Functions over Distributed Data **Streams**	Izchak Sharfman, Assaf Schuster, Daniel Keren

6 结束语

异质信息网络的复杂条件社区搜索是基于简单条件社区搜索并结合实际的用户需求提出的新问题, 它除了包含现有的社区搜索问题之外, 还考虑了非对称元路径、受限元路径和禁止节点约束的情况. 本文首先提出了复杂条件的社区搜索问题, 然后针对不同的搜索条件给出了相应的解决办法, 设计了 CSAMP、CSCMP 和 CSPNC 算法, 并给出了针对 CSPNC 的优化策略, 得到了优化算法 CSPNC-PS 和 CSPNC-AS, 最后在 4 个真实数据集上对所提算法进行了效果和效率的评估, 并给出了一个真实的应用案例.

异质信息网络的复杂条件社区搜索还处在一个探索的阶段, 针对禁止节点约束的社区搜索方法, 直接从必要社

© 中国科学院软件研究所 http://www.jos.org.cn

区中删除禁止社区中节点的方式是否会导致其他问题是下一步需要研究的方向. 另外, 节点的属性信息在社区搜索中同样会影响搜索的结果, 在带属性信息的异质信息网络中进行复杂条件社区搜索同样是一个值得研究的问题.

References:

[1] Sozio M, Gionis A. The community-search problem and how to plan a successful cocktail party. In: Proc. of the 16th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. Washington: ACM, 2010. 939–948. [doi: 10.1145/1835804.1835923]

[2] Tang L, Liu H. Community detection and mining in social media. Synthesis Lectures on Data Mining and Knowledge Discovery, 2010, 2(1): 1–137. [doi: 10.2200/S00298ED1V01Y201009DMK003]

[3] Chen JC, Yuan B. Detecting functional modules in the yeast protein-protein interaction network. Bioinformatics, 2006, 22(18): 2283–2290. [doi: 10.1093/bioinformatics/btl370]

[4] Qiao SJ, Li TR, Han N, Gao YJ, Yuan CA, Wang XT, Tang CJ. Self-adaptive trajectory prediction model for moving objects in big data environment. Ruan Jian Xue Bao/Journal of Software, 2015, 26(11): 2869–2883 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4889.htm [doi: 10.13328/j.cnki.jos.004889]

[5] Qiao SJ, Jin K, Han N, Tang CJ, Ge SDJ, Gutierrez LA. Trajectory prediction algorithm based on gaussian mixture model. Ruan Jian Xue Bao/Journal of Software, 2015, 26(5): 1048–1063 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4796.htm [doi: 10.13328/j.cnki.jos.004796]

[6] Qiao SJ, Wu LC, Han N, Huang FL, Mao R, Yuan CA, Gutierrez LA. Multiple-motion-pattern trajectory prediction of moving objects with context awareness: A survey. Ruan Jian Xue Bao/Journal of Software, 2023, 34(1): 312–333 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6395.htm [doi: 10.13328/j.cnki.jos.006395]

[7] Qiao SJ, Han N, Yue K, Yi YG, Huang FL, Yuan CA, Ding P, Gutierrez LA. Shared-bike demand prediction model based on station clustering. Ruan Jian Xue Bao/Journal of Software, 2022, 33(4): 1451–1476 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6461.htm [doi: 10.13328/j.cnki.jos.006461]

[8] Porter MA, Onnela JP, Mucha PJ. Communities in networks. Notices of the American Mathematical Society, 2009, 56(9): 1082–1100.

[9] Jiang CJ, Li Z. Network community detection. In: Jiang CJ, Li Z, eds. Mobile Information Service for Networks. Singapore: Springer, 2020. 71–101. [doi: 10.1007/978-981-15-4569-6_4]

[10] Zheng YY, Wang MS, Shi C, Wang R. Research on community detection algorithm based on meta path in heterogeneous information network. Journal of Chinese Information Processing, 2018, 32(9): 132–142 (in Chinese with English abstract). [doi: 10.3969/j.issn.1003-0077.2018.09.018]

[11] Huang X, Lakshmanan LVS, Xu JL. Community search over big graphs: Models, algorithms, and opportunities. In: Proc. of the 33rd IEEE Int'l Conf. on Data Engineering. San Diego: IEEE, 2017. 1451–1454. [doi: 10.1109/ICDE.2017.211]

[12] Fang YX, Huang X, Qin L, Zhang Y, Zhang WJ, Cheng R, Lin XM. A survey of community search over big graphs. The VLDB Journal, 2020, 29(1): 353–392. [doi: 10.1007/s00778-019-00556-x]

[13] Shan J, Shen DR, Kou Y, Nie TZ, Yu G. Approach for hot spread node selection based on overlapping community search. Ruan Jian Xue Bao/Journal of Software, 2017, 28(2): 326–340 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5117.htm [doi: 10.13328/j.cnki.jos.005117]

[14] Cui WY, Xiao YH, Wang HX, Wang W. Local search of communities in large graphs. In: Proc. of the 2014 ACM SIGMOD Int'l Conf. on Management of Data. Snowbird: ACM, 2014. 991–1002. [doi: 10.1145/2588555.2612179]

[15] Zhu JC, Wang CK. Approaches to community search under complex conditions. Ruan Jian Xue Bao/Journal of Software, 2019, 30(3): 552−572 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5699.htm [doi: 10.13328/j.cnki.jos.005699]

[16] Shi C, Li YT, Zhang JW, Sun YZ, Yu PS. A survey of heterogeneous information network analysis. IEEE Trans. on Knowledge and Data Engineering, 2017, 29(1): 17–37. [doi: 10.1109/TKDE.2016.2598561]

[17] Fang YX, Yang YX, Zhang WJ, Lin XM, Cao X. Effective and efficient community search over large heterogeneous information networks. Proc. of the VLDB Endowment, 2020, 13(6): 854–867. [doi: 10.14778/3380750.3380756]

[18] Yang YX, Fang YX, Lin XM, Zhang WJ. Effective and efficient truss computation over large heterogeneous information networks. In: Proc. of the 36th IEEE Int'l Conf. on Data Engineering. Dallas: IEEE, 2020. 901–912. [doi: 10.1109/ICDE48307.2020.00083]

[19] Jian X, Wang Y, Chen L. Effective and efficient relational community detection and search in large dynamic heterogeneous information networks. Proc. of the VLDB Endowment, 2020, 13(6): 1723–1736. [doi: 10.14778/3401960.3401969]

[20] Barbieri N, Bonchi F, Galimberti E, Gullo F. Efficient and effective community search. Data Mining and Knowledge Discovery, 2015, 29(5): 1406–1433. [doi: 10.1007/s10618-015-0422-1]

© 中国科学院软件研究所 http://www.jos.org.cn

[21] Huang X, Cheng H, Qin L, Tian WT, Yu JX. Querying *k*-Truss community in large and dynamic graphs. In: Proc. of the 2014 ACM SIGMOD Int'l Conf. on Management of Data. Snowbird: ACM, 2014. 1311–1322. [doi: 10.1145/2588555.2610495]

[22] Akbas E, Zhao PX. Truss-based community search: A Truss-equivalence based indexing approach. Proc. of the VLDB Endowment, 2017, 10(11): 1298–1309. [doi: 10.14778/3137628.3137640]

[23] Wang CK, Zhu JC. Forbidden nodes aware community search. In: Proc. of the 33rd AAAI Conf. on Artificial Intelligence. Honolulu: AAAI, 2019. 758–765. [doi: 10.1609/aaai.v33i01.3301758]

[24] Fang YX, Wang ZR, Cheng R, Wang HZ, Hu JF. Effective and efficient community search over large directed graphs. IEEE Trans. on Knowledge and Data Engineering, 2019, 31(11): 2093–2107. [doi: 10.1109/TKDE.2018.2872982]

[25] Giatsidis C, Thilikos DM, Vazirgiannis M. D-cores: Measuring collaboration of directed graphs based on degeneracy. In: Proc. of the 11th IEEE Int'l Conf. on Data Mining. Vancouver: IEEE, 2011. 201–210. [doi: 10.1109/ICDM.2011.46]

[26] Fang YX, Cheng R, Luo SQ, Hu JF. Effective community search for large attributed graphs. Proc. of the VLDB Endowment, 2016, 9(12): 1233–1244. [doi: 10.14778/2994509.2994538]

[27] Fang YX, Cheng R, Chen YK, Luo SQ, Hu JF. Effective and efficient attributed community search. The VLDB Journal, 2017, 26(6): 803–828. [doi: 10.1007/s00778-017-0482-5]

[28] Huang X, Lakshmanan LVS. Attribute-driven community search. Proc. of the VLDB Endowment, 2017, 10(9): 949–960. [doi: 10.14778/3099622.3099626]

[29] Zhang ZW, Huang X, Xu JL, Choi B, Shang ZC. Keyword-centric community search. In: Proc. of the 35th IEEE Int'l Conf. on Data Engineering. Macao: IEEE, 2019. 422–433. [doi: 10.1109/ICDE.2019.00045]

[30] Liu Q, Zhu YF, Zhao MJ, Huang X, Xu JL, Gao YJ. VAC: Vertex-centric attributed community search. In: Proc. of the 36th IEEE Int'l Conf. on Data Engineering. Dallas: IEEE, 2020. 937–948. [doi: 10.1109/ICDE48307.2020.00086]

[31] Fang YX, Cheng R, Li XD, Luo SQ, Hu JF. Effective community search over large spatial graphs. Proc. of the VLDB Endowment, 2017, 10(6): 709–720. [doi: 10.14778/3055330.3055337]

[32] Fang YX, Wang Z, Cheng R, Li XD, Luo SQ, Hu JF, Chen XJ. On spatial-aware community search. IEEE Trans. on Knowledge and Data Engineering, 2019, 31(4): 783–798. [doi: 10.1109/TKDE.2018.2845414]

[33] Wang K, Cao X, Lin XM, Zhang WJ, Qin L. Efficient computing of radius-bounded *k*-cores. In: Proc. of the 34th IEEE Int'l Conf. on Data Engineering. Paris: IEEE, 2018. 233–244. [doi: 10.1109/ICDE.2018.00030]

[34] Zhu QJ, Hu HB, Xu C, Xu JL, Lee WC. Geo-social group queries with minimum acquaintance constraints. The VLDB Journal, 2017, 26(5): 709–727. [doi: 10.1007/s00778-017-0473-6]

[35] Luo JH, Cao X, Xie XK, Qu Q. Best co-located community search in attributed networks. In: Proc. of the 28th ACM Int'l Conf. on Information and Knowledge Management. Beijing: ACM, 2019. 2453–2456. [doi: 10.1145/3357384.3358107]

[36] Al-Baghdadi A, Lian X. Topic-based community search over spatial-social networks. Proc. of the VLDB Endowment, 2020, 13(12): 2104–2117. [doi: 10.14778/3407790.3407812]

[37] Li RH, Su J, Qin L, Yu JX, Dai QQ. Persistent community search in temporal networks. In: Proc. of the 34th IEEE Int'l Conf. on Data Engineering. Paris: IEEE, 2018. 797–808. [doi: 10.1109/ICDE.2018.00077]

[38] Xu LT, Li RH, Wang GR, Wang B. Research on *k*-Truss community search algorithm for temporal networks. Journal of Frontiers of Computer Science & Technology, 2020, 14(9): 1482–1489 (in Chinese with English abstract). [doi: 10.3778/j.issn.1673-9418.1909050]

[39] Wang ZZ, Yuan Y, Zhou XM, Qin HC. Effective and efficient community search in directed graphs across heterogeneous social networks. In: Proc. of the 31st Australasian Database Conf. on Databases Theory and Applications. Melbourne: Springer, 2020. 161–172. [doi: 10.1007/978-3-030-39469-1_13]

[40] Qiao LP, Zhang ZW, Yuan Y, Chen C, Wang GR. Keyword-centric community search over large heterogeneous information networks. In: Proc. of the 26th Int'l Conf. on Database Systems for Advanced Applications. Taipei: Springer, 2021. 158–173. [doi: 10.1007/978-3-030-73194-6_12]

[41] Yang J. The research on complex conditional community search in heterogeneous information network [MS. Thesis]. Kunming: Yunnan University, 2021 (in Chinese with English abstract).

[42] Sun YZ, Han JW. Mining Heterogeneous Information Networks: Principles and Methodologies. Beijing: China Machine Press, 2017 (in Chinese).

[43] Shi C, Li YT, Yu PS, Wu B. Constrained-meta-path-based ranking in heterogeneous information network. Knowledge and Information Systems, 2016, 49(2): 719–747. [doi: 10.1007/s10115-016-0916-1]

[44] Sun YZ, Han JW, Yan XF, Yu PS, Wu TY. PathSim: Meta path-based top-K similarity search in heterogeneous information networks. Proc. of the VLDB Endowment, 2011, 4(11): 992–1003. [doi: 10.14778/3402707.3402736]

© 中国科学院软件研究所 http://www.jos.org.cn

[45] Shi C, Kong XN, Huang Y, Yu PS, Wu B. HeteSim: A general framework for relevance measure in heterogeneous networks. IEEE Trans. on Knowledge and Data Engineering, 2014, 26(10): 2479–2492. [doi: 10.1109/TKDE.2013.2297920]

附中文参考文献:

[4] 乔少杰, 李天瑞, 韩楠, 高云君, 元昌安, 王晓腾, 唐常杰. 大数据环境下移动对象自适应轨迹预测模型. 软件学报, 2015, 26(11): 2869–2883. http://www.jos.org.cn/1000-9825/4889.htm [doi: 10.13328/j.cnki.jos.004889]

[5] 乔少杰, 金琨, 韩楠, 唐常杰, 格桑多吉, Gutierrez LA. 一种基于高斯混合模型的轨迹预测算法. 软件学报, 2015, 26(5): 1048–1063. http://www.jos.org.cn/1000-9825/4796.htm [doi: 10.13328/j.cnki.jos.004796]

[6] 乔少杰, 吴凌淳, 韩楠, 黄发良, 毛睿, 元昌安, Gutierrez LA. 情景感知驱动的移动对象多模式轨迹预测技术综述. 软件学报, 2023, 34(1): 312–333. http://www.jos.org.cn/1000-9825/6395.htm [doi: 10.13328/j.cnki.jos.006395]

[7] 乔少杰, 韩楠, 岳昆, 易玉根, 黄发良, 元昌安, 丁鹏, Gutierrez LA. 基于数据场聚类的共享单车需求预测模型. 软件学报, 2022, 33(4): 1451–1476. http://www.jos.org.cn/1000-9825/6461.htm [doi: 10.13328/j.cnki.jos.006461]

[10] 郑玉艳, 王明省, 石川, 王锐. 异质信息网络中基于元路径的社团发现算法研究. 中文信息学报, 2018, 32(9): 132–142. [doi: 10.3969/j.issn.1003-0077.2018.09.018]

[13] 单菁, 申德荣, 寇月, 聂铁铮, 于戈. 基于重叠社区搜索的传播热点选择方法. 软件学报, 2017, 28(2): 326–340. http://www.jos.org.cn/1000-9825/5117.htm [doi: 10.13328/j.cnki.jos.005117]

[15] 竺俊超, 王朝坤. 复杂条件下的社区搜索方法. 软件学报, 2019, 30(3): 552–572. http://www.jos.org.cn/1000-9825/5699.htm [doi: 10.13328/j.cnki.jos.005699]

[38] 徐兰天, 李荣华, 王国仁, 王彪. 面向时序图的*k*-Truss社区搜索算法研究. 计算机科学与探索, 2020, 14(9): 1482–1489. [doi: 10.3778/j.issn.1673-9418.1909050]

[41] 杨杰. 异质信息网络复杂条件社区搜索研究 [硕士学位论文]. 昆明: 云南大学, 2021.

[42] 孙艺洲, 韩家炜. 异构信息网络挖掘: 原理和方法. 北京: 机械工业出版社, 2017.

王家龙(1993－), 男, 博士生, 主要研究领域为数据挖掘, 社会网络分析.

王丽珍(1962－), 女, 博士, 教授, 博士生导师, CCF 杰出会员, 主要研究领域为数据挖掘, 数据仓库, 计算机算法.

杨杰(1996－), 男, 硕士生, 主要研究领域为数据挖掘, 社区搜索.

王睿康(1999－), 男, 本科生, 主要研究领域为社会网络分析.

周丽华(1968－), 女, 博士, 教授, 博士生导师, CCF 专业会员, 主要研究领域为数据挖掘, 社会网络分析, 人工智能.

© 中国科学院软件研究所 http://www.jos.org.cn