规模受限的影响力社区搜索

杜 明,宋嘉祎,周军锋
(东华大学计算机科学与技术学院,上海 201620)

摘 要: 社区搜索用于返回包含给定查询结点且符合查询条件的密集连通子图. 目前,大部分已有社区搜索方法主要关注社区的结构,没有考虑到特定应用中资源受限的情况,且忽略了社区的属性特征,无法满足用户对社区搜索的个性化要求. 针对该问题,本文提出了规模受限的影响力社区搜索(Size-Constrained Influential Community search,SCIC),设计了基于深度优先搜索的基础算法,在此基础上进一步提出了基于结点预处理、剪枝规则和贪心策略的优化算法,用于减少冗余计算,加速枚举过程. 在10个不同规模的数据集上进行实验,实验结果表明基础算法在搜索获得的社区规模和影响力上均优于已有算法,同时,本文提出的优化算法能够显著提升搜索效率,将响应时间缩减至基础算法的1%.

关键词: 数据图;社区搜索;k-核;加权图;规模受限社区;影响力社区搜索
基金项目: 上海市自然基金项目(No.20ZR1402700);国家自然基金项目(No.61472339,No.61873337)
中图分类号: O157.5;TP301 **文献标识码:** A **文章编号:** 0372-2112(2023)05-1207-08
电子学报 URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20220538

Size-Constrained Influential Community Search

DU Ming,SONG Jia-yi,ZHOU Jun-feng
(*School of Computer Science and Technology*, *Donghua University*, *Shanghai* 201620, *China*)

Abstract: Community search is used to find a densely connected subgraph containing the given query vertex. Currently, most existing community search approaches do not consider the resource constraints and the attributes of the community, and they mainly focus on the structure of the community and cannot meet the personalized demands for users´ community search. For this problem, we propose size-constrained influential community search (SCIC). We design a baseline enumeration algorithm based on depth-first search. Further, we show three optimization techniques to reduce the redundant computation and accelerate the enumeration procedure, including node preprocessing, pruning rules and greedy strategies. The experimental results on 10 real datasets with different sizes demonstrate that comparing with the existing algorithms, our baseline enumeration algorithm has better performance on community size and influence score. At the same time, the experimental results also show that our optimized algorithm can significantly improve the search efficiency and shorten the response time to 1% comparing with the baseline enumeration algorithm.

Key words: data graph; community search; k-core; weighted graph; size-constrained community; influential community search
Foundation Item(s): National Natural Science Foundation of Shanghai (No.20ZR1402700); National Natural Science Foundation of China (No.61472339, No.61873337)

1 引言

社区是由紧密连接的结点构成的子图. 在社交网络中,社区由积极互动的用户组成[1,2];在万维网中,社区由主题相似的页面构成[3]. 社区搜索的目的是寻找一个包含查询结点且结点紧密连接的子图. 社区搜索关注局部网络结构,返回个性化的社区;例如:构建个性化研究小组[4],蛋白质功能检测[5]等.

目前,有两类社区搜索问题得到了广泛的关注. 第一类是无结点数量限制的社区搜索问题,如:异构图上的社区搜索[6]、结点位置限制的社区搜索[7]等. 这类问题主要关注社区的紧密度,没有考虑到社区搜索中存在资源限制情况. 第二类是结点数量受限的社区搜索问题,如:结点数量上界受限的社区搜索[8]、限制上下界的社区搜索[9]等. 相较于无结点数量限制的社区搜索,这类问

收稿日期:2022-08-29;修回日期:2022-09-29;责任编辑:覃怀银

题考虑到了资源限制，但是与第一类社区搜索问题相似，仅考虑社区的结构特征，忽略了社区的属性特征[10]. 例如，在构建服务集群问题中，除了需要考虑服务器数量和服务结点关联度两个因素之外，还需要考虑性能瓶颈因素才能构建最优服务集群. 以上三个因素分别对应规模、结构以及属性约束，而现有社区搜索仅考虑前两个因素，无法提供面向多约束问题的有效解决方案.

本文针对现有社区搜索技术存在的问题，提出规模受限的影响力社区搜索(Size-Constrained Influential Community search，SCIC)问题. 该问题不仅考虑了社区规模，还能够兼顾社区的结构特征和属性特征约束. 在此基础上，本文设计了一种基于枚举的解决方案，并提出三种优化策略，包括结点筛选、剪枝策略以及贪心优化，从而可以高效求解满足结构、规模约束且影响力最大的有效社区.

2 背景知识和相关工作

2.1 背景知识

给定无向图 $G=(V_G, E_G, \Omega)$，其中 V_G 为结点集，E_G 为边集；$\Omega: V \to R^+$ 表示图中结点到影响力的映射，$\Omega(u)$ 表示结点 u 的影响力. 结点 u 在图 G 中的邻居结点集合记作 $N_G(u)=\{v|(u,v)\in E_G\}$；结点 u 的度记作 $d_G(u)=|N_G(u)|$.

定义 1 给定图 G 的子图 $S=(V_S, E_S, \Omega)$，S 为 G 的诱导子图，则对 $\forall u,v\in V_S, (u,v)\in E_S$，当且仅当 $(u,v)\in E_G$.

定义 2 给定图 G 的诱导子图 $S=(V_S, E_S, \Omega)$，S 的影响力 $\inf(S)$ 为 S 中所有结点影响力的最小值.

定义 3 给定图 G 的诱导子图 $S=(V_S, E_S, \Omega)$，S 的最小度为 S 中结点的度的最小值，用 $\mathrm{minD}(S)$ 表示.

定义 4 给定图 $G=(V_G, E_G, \Omega)$，查询结点 $q\in V_G$，内聚度阈值 k，以及结点数量范围 $[l, h]$，若诱导子图 S 满足以下条件：(1) S 连通且包含查询结点 q；(2) $l\leqslant|V_S|\leqslant h$；(3) $\mathrm{minD}(S)\geqslant k$；则称 S 是一个可行社区.

定义 5 给定图 $G=(V_G, E_G, \Omega)$，查询结点 $q\in V_G$，内聚度阈值 k，以及结点数量范围 $[l, h]$，影响力最大的可行社区称为结果社区.

值得注意的是，在可行社区集合中可能存在多个影响力相同的社区，因此结果社区不具有唯一性.

问题定义(结点数量限制的影响力社区搜索)：给定图 $G=(V_G, E_G, \Omega)$，查询结点 $q\in V_G$，内聚度阈值 k，以及结点数量范围 $[l, h]$，结点数量范围限制的影响力社区搜索 SCIC 从图 G 中找到查询点 q 的一个结果社区 H.

例 1 考虑图 1，设查询结点为 v_3，$[l, h]=[4,6]$，$k=3$. 结点集合 $\{v_1, v_2, v_3, v_4, v_5\}$ 是一个可行社区，其影响力为 20；结点集合 $\{v_3, v_5, v_6, v_7\}$ 是另一个可行社区，其影响力为 70，同时该社区也是影响力最大的可行社区，因此它是结果社区.

表 1 是本文内容涉及到的重要符号及其意义.

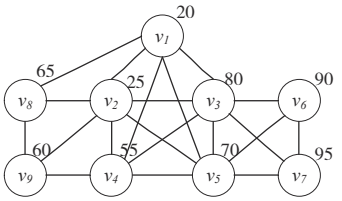

图 1 结点带权的无向图 G 示例

表 1 本文所用符号及其意义

符号	意义
$\inf(G)$	图 G 的影响力
$\Omega(u)$	结点 u 的影响力
$d_G(u)$	结点 u 在图 G 中的度
$N_G(u)$	结点 u 在图 G 中的邻居结点集
$\mathrm{dis}_G(u,v)$	结点 u 和结点 v 在图 G 中的最小距离

2.2 相关工作

2.2.1 结点数量范围限制的社区搜索

目前，结点数量限制的社区搜索主要有两类算法：只限制结点数量上界的 ES(Evolution Strategy)算法[8]以及同时限制结点数量上下界的 SC-BRB(Self-Centering Buckling-Restrained Brace)算法[9].

ES 算法[8]分为向社区内添加结点的 Expand 算法以及从图 G 中删除结点的 Shrink 算法. ES 算法使用 k-truss[11]作为子图紧密度模型，根据结点数量上界选择不同策略搜索符合条件的社区，并提出了一种结点紧密度计算方法.

SC-BRB 算法[9]能够解决同时限制社区数量上下界的社区搜索. 该算法使用 k-core[12,13]模型作为子图紧密度的衡量标准. 在枚举社区的过程中，SC-BRB 算法使用剪枝算法和贪心算法来提升查询效率.

ES 算法与 SC-BRB 算法都没有考虑图中结点的权值，因此无法作为 SCIC 问题的解决方案.

2.2.2 影响力社区搜索

在影响力社区搜索的研究中，Li[14]提出了 DFS(Depth First Search)搜索算法以及离线的 ICP(Index Condition Pushdown)索引查询方法. DFS 搜索使用了贪心思想从图中迭代删除影响力最小的结点，直到返回影响力最大社区. ICP 索引使用 k-core 模型对图进行分解[13]，建立从 $k=1$ 到 k 最大的索引森林. 查询时根据给定内聚度阈值 k 首先找到对应索引树；而后根据给定的查询结点，找到索引树中的节点及其子孙，由节点中的结点集合构成的诱导子图即为最终影响力最大社区. 这些方法在 SCIC 问题中不具有可行性.

除了上述两类社区搜索外，还有在异构图上进行社区搜索[6]、基于属性相似度的社区搜索[10,15]、基于空间的社区搜索[7]等问题. 这些社区搜索问题在不同类

型的图上,根据对应限制条件,搜索符合条件的子图.这些社区搜索问题没有考虑到结点数量限制以及影响力最大[16]的要求,因此不再详述.

3　基于状态的枚举算法

针对SCIC问题,需要能够兼顾结点数量和社区影响力的算法.本文提出一种基于枚举的方法——BasicEnum算法,其基本思想如下:使用深度优先搜索算法穷举社区的枚举状态,获得所有包含查询结点的子图,从子图集合中找到所有可行社区,并选择影响力最大的可行社区作为结果社区输出.

算法1给出了BasicEnum算法的主要过程.第1~5行为算法的主流程,第6~9行展示了预处理流程,第10~18行展示了BasicEnum算法的枚举过程:使用深度优先搜索穷举子图的构成状态.最后输出Result$C$作为SCIC问题的最优解.算法将复杂的子图穷举转化为递归搜索枚举状态<C, R>.枚举状态将V_G划分为候选集C以及结果集R:候选集C表示后续枚举过程中可能加入社区的结点集合,R表示目前已加入社区的结点集合,满足$V_G \supseteq C \cup R$且$C \cap R=\varnothing$,$R \subseteq V_X \subseteq C \cup R$.候选集$C$反映了可行社区未来可能的构成情况,结果集$R$反映了当前社区的构造情况.BasicEnum通过预处理方法获得初始候选集C和结果集R:初始结果集R为空;初始候选集$C=\{u|\ \mathrm{dis}_G(u, q) \leqslant h,\ u \in V_G\}$是$V_G$中与查询结点$q$的距离小于等于社区数量上界$h$的结点集合.

BasicEnum算法递归搜索枚举状态的过程可以构建为一棵搜索树,搜索树中的节点表示当前枚举状态<C, R>,其孩子节点表示两个枚举子空间:左子树为<$C\backslash\{v\}$, $R\cup\{v\}$>,右子树为<$C\backslash\{v\}$, R>.BasicEnum算法的时间复杂度即为搜索树的节点数量,易得时间复杂度是$O(n^h)$,n是初始候选集C的结点数量,h是题设给定的结点数量上界.

算法1　BasicEnum算法

输入:图G,查询结点q,结点数量上下界$[l, h]$,内聚度阈值k

输出:结果社区Result$C$

```
1. BasicEnum(G, q, l, h, k)
2.    G' ← Pretreat(G, q, h, k)
3.    IF |G'| ≥ l THEN
4.       C ← V_G', R ← ∅
5.       Enum(C\{q}, R∪{q})
6. Function Pretreat(G, q, h, k)
7.    WHILE exist dis_G(v, q) > h DO
8.       G = G\{v}
9.    RETURN G
10.Function Enum(C, R)
11.   IF |R|∈[l, h] & minD(S_R) ≥ k & inf(S_R) > inf(ResultC) THEN
12.      ResultC = S_R
13.      RETURN
14.   IF |R| < h THEN
15.      v ← C
16.      Enum(C\{v}, R∪{v})
17.      Enum(C\{v}, R)
```

例2　给定如图1所示的无向图,内聚度阈值$k=2$,结点数量范围$[4,6]$,查询结点$q=v_3$.将深度优先的递归搜索过程转换为如图2所示的搜索树.初始阶段,R中仅包含查询结点q,inf(R)为80.在随后的迭代过程中判断v_1是否加入R:左子树表示<$C\backslash\{v_1\}$, $R\cup\{v_1\}$>,右子树表示<$C\backslash\{v_1\}$, R>.不断向下递归的判断结点,最后获得$\{v_0, v_1, v_2, v_3, v_5\}$为影响力最大社区.

BasicEnum算法的优点是逻辑简单,且可获得精确解.缺点是效率低,主要原因如下:

(1)访问结点数量多:算法的响应时间与初始候选集C中的结点数量密切相关.需要更加严格的筛选策略以减少其数量.

(2)搜索树宽度大:算法需要搜索枚举状态的左右子树,导致搜索树的宽度不断增加、响应时间过长.

(3)搜索树深度大:算法需要对每个结点进行递归搜索,导致搜索树深度增加,响应时间过长.

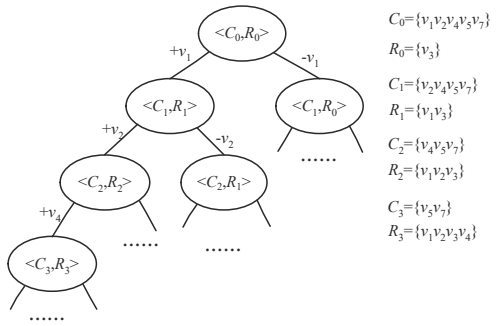

图2　BasicEnum枚举搜索树

4　算法优化

针对BasicEnum算法存在的问题,本节提出贪心剪枝算法(Greedy Reducing Enum,GRE)进行优化,在预处理阶段筛选候选结点,在求解过程中使用剪枝规则(prune rule)和贪心策略(greedy strategy)进行优化.

4.1　候选结点筛选策略

候选结点筛选策略通过删除初始候选集内的冗余结点减少参与递归的结点数量.

定义6　给定图$G=(V_G, E_G, \Omega)$,G的直径D_G为G中任意结点间最小距离的最大值.

给定子图直径D_G,内聚度阈值k,根据文献[8,17],则子图S至少包含$n(k, D_G)$个结点,如式(1)所示:

$$n(k,D_G)=\begin{cases}k+D_G, k=1 \text{ or } D_G\in[1,2]\\ k+D_G+1+\left\lfloor\dfrac{D_G}{3}\right\rfloor(k-2), \text{other}\end{cases} \tag{1}$$

定理1 在给定结点数量上界h和内聚度阈值k时,有式(2)所示的子图直径公式:

$$D_{\max}=\begin{cases}h-1, k=1\\ \max\{2,\left\lceil\dfrac{3(h-k-1)}{k+1}\right\rceil\}, k\neq 1\end{cases} \tag{2}$$

证明 由式(1)求得的社区结点数量下界$n(k, D_G)$应小于等于给定结点数量上界h,有$n(k, D_G)\leqslant h$. 在题设中,k和h属于已知变量,根据式(1),需要根据k值进行分类讨论:在$k=1$时,易得$D_G\leqslant h-1$;而$k>1$时,则有$k+D_G+1+\left\lfloor\dfrac{D_G}{3}\right\rfloor(k-2)\leqslant h$,经过提取公因式和移项可得$D_G\leqslant\left\lceil\dfrac{3(h-k-1)}{k+1}\right\rceil$,若求得$D_G\leqslant 1$,与式(1)给出的限制条件矛盾,有$D_G>2$. 结合两种情况,最后得子图直径公式;因此,有式(2)成立. 证毕.

根据式(2),有如下候选结点约减规则,对$\forall u\in V_G$,若结点u符合:(1)查询结点q与u的距离大于$D_{\max}$,$\mathrm{dis}_G(q, u)>D_{\max}$;或(2)结点$u$的度小于内聚度阈值$k$,$d_G(u)<k$,那么$u$必不属于初始候选集.

例3 如图1所示,设查询结点为v_7,$[l, h]=[3, 4]$,$k=2$. 使用预处理算法可以获得$\{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9\}$构成的初始候选集$C$;使用候选结点筛选策略,可以得到子图最大直径为2,由此可得初始候选集C为$\{v_1, v_2, v_3, v_4, v_5, v_6\}$,缩小了初始候选集的大小.

4.2 剪枝规则

剪枝策略1(距离剪枝):从候选集C中选择结点v加入结果集R时,从C中删除与v的距离大于$D_{\max}$的结点,$V_{\mathrm{del}}=\{u|\ \mathrm{dis}_{C\cup R}(u, v)>D_{\max}\}$.

证明 使用反证法,设$\exists u\in C, v\in R, \mathrm{dis}(v, u)>D_{\max}$,使得$u$可以加入结果集$R$,且$S_R$为可行社区. 在递归搜索枚举状态的过程中,将$u$加入$R$,则有$n(k, \mathrm{dis}(v, u))>h\geqslant n(k, D_{\max})$,不符合结点数量限制,$S_R$不是可行社区,与给定要求矛盾. 因此在$C$中不存在与$R$的距离大于$D_{\max}$的结点. 证毕.

剪枝策略1根据子图直径公式,将候选集中距离过大的结点移除,避免在后续递归搜索中选择这些结点加入社区.

剪枝策略2(度剪枝):从候选集C中选择结点v加入结果集R时,若$d_{S_{C\cup R}}(v)=k$,则将$N_C(v)$全部加入结果集R.

证明 子图S_R为可行社区,则$\forall u\in R$,有$d_{S_R}(v)\geqslant k$. 若在$S_{C\cup R}$中,结点v的度恰好为k,将v加入R时,需要使$d_{S_R}(v)\geqslant k$,则将v的所有邻居都加入R. 证毕.

剪枝策略2针对度较小的结点,批量地将邻居加入结果集.

剪枝策略3(影响力剪枝):若存在影响力极大的可行社区Result$C$,从候选集C中删除权值小于等于Result$C$影响力的结点,即$\{v|\ \Omega(u)<\inf(\mathrm{Result}C)\}$.

证明 反证法:$\exists u\in C$,有$\Omega(u)<\inf(\mathrm{Result}C)$,使得$u$可以加入结果集$R$,且$S_R$为结果社区. 在递归搜索枚举状态的过程中,将$u$加入$R$,根据社区影响力的定义,有$\inf(S_R)=\Omega(u)<\inf(\mathrm{Result}C)$,因此$S_R$不可能为结果社区;矛盾. 证毕.

剪枝策略3根据已有的可行社区从候选集中移除权值较小的结点,避免在后续递归搜索中将这些结点加入社区.

在剪枝策略中,有三种算法:算法2为距离剪枝算法,算法3为度剪枝算法,算法4为影响力剪枝算法.

算法2迭代地删除不满足距离约束公式的结点,同时维护了子图的内聚度.

算法2 距离剪枝算法

输入:候选集C,结果集R,最大直径D
输出:删除结点集合del

```
1. distanceBranchReduce(C, R, D)
2.   del ←∅
3.   WHILE exist delete u DO
4.     FOR EACH u∈C, v∈R DO
5.       IF dis_{C∪R}(v,u)> D THEN
6.         push u into del
7.         delete u from C
8.       update degree;
9.       push v into del, delete d_{C∪R}(v)<k
10.  RETURN del
```

算法3批量地向结果中加入满足度约束的结点,减少了搜索的次数.

算法3 度剪枝算法

输入:候选集C,结果集R,内聚度阈值k
输出:批量加入结点集合add

```
1. degreeBranchReduce(C, R, k)
2.   add ←∅
3.   IF d_{C∪R}(v)=k THEN
4.     push N_{C∪R}(v) into add
5.   RETURN add
```

算法4在枚举过程中,删除权值较小的结点.

例4 如图1所示,给定图G,查询结点$q=\{v_3\}$,结点数量范围$[4,5]$,内聚度阈值$k=3$. 设当前候选集$\{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9\}$,结果集为$\{v_3\}$,若将$\{v_6\}$加入$R$. 根据剪枝策略1,结点$v_8$, v_9与v_6的距离大于最大直径$D_{\max}=2$,因此从C中移除v_8, v_9. 根据剪枝策略2,为了使v_6

算法4　影响力剪枝算法

输入:候选集C,结果集R,可行社区最大影响力I

输出:删除结点集合del

```
1. influentialBranchReduce(C, R, I)
2.   del ←∅
3.   WHILE exist delete Verticle DO
4.     for each u∈C∪R
5.       IF inf(u) < I THEN
6.         push u into del
7.     update degree;
8.     push v into del, delete d_{C∪R}(v)<k
9.   RETURN del
```

满足最小度限制,将邻居结点v_5,v_7加入R. 根据剪枝策略3,若当前已有$\{v_3, v_4, v_5, v_7\}$构成的可行社区影响力为55,则将权值小于等于55的结点v_1,v_2,v_4从C中移除.

4.3　贪心策略

4.3.1　子树约减

为了使结果集R构成的诱导子图具有连通性,选择结点加入社区时,总是选择结果集R的邻居结点进行递归. 在候选集的基础上,设置可行集P,表示当前枚举状态下可以加入R的结点,有$P=\left\{v \middle| N_G(u_0)\cup N_G(u_1)\cup\cdots\cup N_G(u_i)/R, u_i\in R\right\}, P\subseteq C$. 因此,枚举状态可以表示为<$P$, R>.

为了使社区的影响力最大化,优先选择P中权值最大的结点加入R. 右子树约减策略:搜索枚举状态<P_0, R_0>时,若在后续枚举空间中,存在一条搜索树路径,使得所有加入结果集R的结点权值均大于等于$\inf(R_0)$,并且R构成的诱导子图是可行社区,那么枚举状态<P_0, R_0>不需要递归搜索右子树. 具体过程如算法5所示,主要过程与BasicEnum算法相似,但是结点选择时不再使用随机算法,而是选择权值最大的结点进行处理. 同时,在每次递归过程中,需要对P进行更新,使P中结点始终与R相邻. 此外,贪心右子树约减使用返回值向上层递归反馈搜索结果,约减冗余右子树搜索.

例5　给定如图1所示的图G,查询结点$q=\{v_3\}$,结点数量限制$\{4,5\}$,内聚度阈值$k=3$. 设当前结果集$R_0=\{v_3, v_5\}$,可行集$P_0=\{v_1, v_2, v_4, v_6, v_7\}$. 根据贪心右子树约减算法,可以得到如图3所示的搜索树. 枚举状态<P_0, R_0>的左子树中,可以找到$\{v_3, v_5, v_6, v_7\}$构成的影响力为70的可行社区. 而右子树中只有$\{v_3, v_4, v_5, v_6\}$构成的影响力为55的可行社区. 因此枚举状态<P_0, R_0>的右子树可以被省略.

4.3.2　结点选择策略

在递归搜索枚举状态时,结果集R无法构成可行社区时,则需要继续递归搜索枚举状态;而递归深度越大,搜索树的规模也越大,出现的冗余枚举状态也越多.

算法5　GreedyEnum算法

输入:可行集P,结果集R

输出:结果社区Result$C$

```
1. GreedyEnum(P, R)
2.   IF R is SCIC THEN
3.     Compare(ResultC, R)
4.     RETURN true
5.   IF |R| ≥ h THEN
6.     RETURN false
7.   // 选择权值最大结点
8.   v ← {u|Ω(u)≥Ω(u_i), ∀u_i ∈ V_P}
9.   // P内结点与R内结点始终相邻
10.  update P and R by v
11.  IF GreedyEnum(P, R) THEN
12.    RETURN true
13.  restore P and R
14.  RETURN GreedyEnum(P\{v}, R)
```

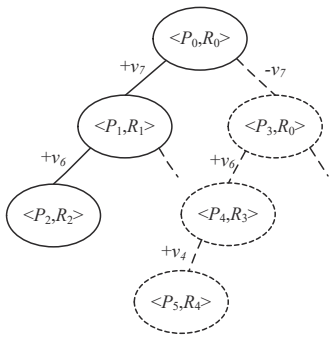

图3　贪心右子树约减实例

由于社区影响力仅和权值最小的结点有关,因此只要结点满足$\Omega(v)\geqslant\inf(R)$,就不会使社区影响力发生改变. 为此引入质量集$Q$和备选集$T$作为可行集$P$的补充,有$Q\cap T=\varnothing$,以及$Q\cup T=P$;其中,质量集$Q$是可行集$P$中权值大于等于$\inf(R)$的结点集合;备选集$T$是可行集$P$中权值小于$\inf(R)$的结点集合. 由此,枚举状态从<$P$, R>可以转化为<Q, T, R>.

使用优先队列维护质量集与备选集:质量集Q根据结点v与结果集R的关联程度对集合进行排序. 使用两类规则对集合内的结点比较大小:(1)$\mathrm{conn}(v, R)=|\{u|(u, v), u\in V_C\}|$越大,关联越紧密;(2)若$\mathrm{conn}(v, R)$相等,$\mathrm{co}(v, R)=|\{u|(u, v)\in E, u\in R, d_R(u)<k\}|$越大,关联越紧密. $\mathrm{conn}(v, R)$表示结点v和图G的紧密程度;值越大表示结点v与结果集R中更多的结点关系密切,更有可能加入R中. $\mathrm{co}(v, R)$表示结点对R的内聚度满足度;值越大表示结点v可以使结果集R中更多的结点满足内聚度限制,更有可能使R构成可行社区.

例6　给定如图1所示的图G,查询结点$q=\{v_3\}$,结点数量限制$[4,5]$,内聚度阈值$k=3$. 设当前结果集

$R_0=\{v_2, v_3, v_5\}$，质量集 $Q_0=\{v_6, v_7, v_9\}$，备选集 $T_0=\{v_1, v_2\}$ 根据质量集排序策略，v_6，v_7 相较于 v_9 和 R_0 的关系更加密切，因此优先选择 v_6，v_7 进行搜索．而在 T_0 中 v_2 的影响力大于 v_1，因此 v_2 优先级较高．

4.4 算法描述

综合上文提出的三种优化策略，可得到优化后的算法，如算法6所示本文提出了贪心剪枝算法（Greedy Reducing Enum，GRE）．

算法6 GreedyReducingEnum 算法

输入：质量集 Q，备选集 T，结果集 R
输出：结果社区 Result$C$

```
GreedyReducingEnum(Q, T, R)
  IF |R|∈[l,h] and minD(R) ≥ k and inf(R) >inf(ResultC) THEN
    ResultC = R
    RETURN true
  // 距离剪枝
  vSet = distanceBranchReduce(Q∪T, R, D)
  remove vSet from Q and T
  // 度剪枝
  vSet = degreeBranchReduce(Q∪T, R, k)
  add vSet into R
  // 影响力剪枝
  vSet = influentialBranchReduce(C, R, I)
  remove vSet from Q and T
  // 贪心策略
  IF Q =∅ THEN
    v ← T.top
  ELSE THEN
    v ← Q.top
  update Q and T by v
  IF GreedyReducingEnum(Q, T, R) then
    RETURN true
  restore Q, T and R
  RETURN GreedyReducingEnum(Q, T, R)
```

贪心剪枝算法仍然使用了基于深度优先搜索的递归状态枚举，搜索符合条件的社区．贪心剪枝算法根据当前枚举状态（Q，T，R）动态维护质量集 Q 和备选集 T：使用剪枝策略约减结点，根据 R 的影响力添加或删除 Q 与 T 中的结点．在递归枚举搜索的过程中，使用贪心右子树规约，减少递归次数，加速算法效率．

最坏情况下，优化算法和基础算法有相同的时间复杂度，但是实验结果可以看出，使用优化策略后，算法性能有显著的改善．

5 实验结果

5.1 实验环境

本实验所用计算机的配置如下：处理器为 Intel（R）Core（TM）i5-5200U CPU @ 2.20 GHz 2.19 GHz，内存（RAM）12 GB，操作系统为 Windows 10.

用于比较的算法包括 GreedyD[1]，GreedyF[1]，SC-BRB[9] 以及本文提出的 BasicEnum 基础枚举算法和 GRE 算法．算法采用 C++ 语言实现，通过 Visual Studio 2019 编译运行，解决方案为 Release，解决方案平台为 x64. GreedyF，GreedyD 以及 SC-BRB 代码由本文作者根据伪代码实现．

5.2 数据集

本文所用数据集为10个大型有向无环图数据集，表2所示为数据集统计信息．数据集 arxiv、pumbed、mtbrv、anthra、human、Email-Euall、10citeseerx、05citeseerx、wikitalk、cit-Patents 均为图数据集；作者对上述数据集进行调整：首先，将原始数据集中的有向边改为无向边；其次，对数据集中的每一个结点，根据正态分布规则，添加影响力值；在上述两个改造后，数据集就变为带权无向图．为了检验 GRE 算法的优化效果，在响应时间对比实验中，使结果社区 Result$C\neq\varnothing$. 因此，本文对每一个数据集都进行预处理，使查询结点集合满足最小度限制，且 k-core 的结点数量大于给定结点数量范围下界．此外，为了直观的展示查询效果，控制查询变量，查询集合中仅包含一个结点．

表2 数据集统计信息

数据集	$\|V\|$	$\|E\|$	平均度
arxiv	6 000	66 707	11.12
pumbed	9 000	40 028	4.45
mtbrv	9 602	10 245	1.07
anthra	12 499	13 104	1.05
human	38 811	39 576	1.02
Email-Euall	231 000	223 004	0.97
10citeseerx	770 539	1 501 126	1.95
05citeseerx	1 457 057	3 002 252	2.06
wikitalk	2 281 879	2 311 570	1.01
cit-Patents	3 774 768	16 518 947	4.38

5.3 性能比较以及分析

5.3.1 社区规模与影响力对比

对于给定 $k=3$，$[l, h]=[6, 9]$ 的 SCIC 问题，使用 GreedyD，GreedyF，SC-BRB 以及 GRE 算法进行结果对比．如表3所示，每一列数据中，左侧数据表示对应算法搜索社区的规模，右列括号内的数据表示算法搜索社区的影响力．GreedyF 以及 GreedyD 算法在实验中不能找到满足题设结点数量限制的社区．其主要原因在于算法仅针对上界受限，无法找到规模下界受限的社区，因此可能出现社区结点数量小于给定下界的情况．GreedyF 算法与 GreedyD 算法既不能保证结果社区满足规模限制，也无法保证社区影响力最大化．而 SC-BRB 算法以及本文的 GRE 算法找到的结果社区都严格符合给定的规模限制．SC-BRB 算法仅关注社区结构，找到

表3　社区节点数量范围与影响力(k=3,[l,h]=[6,9])

数据集	GreedyD	GreedyF	SC-BRB	GRE
arxiv	4 (8)	4 (8)	7 (8)	9 (26)
pumbed	73 (3)	112 (12)	9 (23)	9 (23)
mtbrv	15 (1)	8 (1)	9 (1)	8 (1)
anthra	18 (10)	14 (10)	9 (12)	9 (54)
human	20 (23)	32 (23)	6 (23)	9 (32)
Email-Euall	12 (2)	73 (9)	6 (2)	7 (50)
10citeseerx	24 (9)	13 (17)	6 (17)	6 (17)
05citeseerx	53 (5)	14 (23)	6 (23)	6 (55)
wikitalk	296 (5)	113 (5)	9 (5)	6 (5)
cit-Patents	19 (3)	193 (3)	8 (17)	8 (17)

内聚度最大的社区,忽略了结点权值对社区的影响,因此找到的社区影响力都较小. GRE算法兼顾了影响力和社区结构,总是可以找到影响力最大的结果社区.

当内聚度阈值k以及社区规模发生变化时,会对结果社区产生相应影响,当社区规模变为[12,15]时,结果社区的影响力以及规模如表4所示.

表4　社区节点数量范围与影响力(k=3,[l, h]=[12,15])

数据集	GreedyD	GreedyF	SC-BRB	GRE
arxiv	15(8)	18(8)	15(8)	12(22)
pumbed	73(12)	112(12)	14(23)	12(28)
mtbrv	15(1)	25(1)	13(1)	13(1)
anthra	18(10)	29(10)	15(17)	12(22)
human	20(23)	32(23)	12(23)	14(23)
Email-Euall	12(2)	73(9)	15(9)	12(15)
10citeseerx	24(9)	30(17)	12(27)	12(27)
05citeseerx	53(5)	28(23)	14(5)	13(31)
wikitalk	296(5)	113(5)	15(5)	15(5)
cit-Patents	193(3)	19(3)	15(17)	12(31)

随着社区规模的增长,GreedyD和GreedyF仍然无法满足规模限制的要求;此外,SC-BRB算法与GRE算法求得的结果社区差别变大,GRE算法在社区影响力上有更大的优势.

5.3.2　查询响应时间对比

图4展示了BasicEnum算法,剪枝规则prune rule,贪心策略greedy strategy以及GRE算法在相同搜索条件下的响应时间. 可以看出,剪枝规则和贪心策略都能够提升查询效率,而GRE算法显著减少了搜索响应时间的数量级,有效缩短了查询时间.

为了验证不同结点数量范围对查询响应时间的影响,本文使用了控制变量法. 结点数量范围选择了[3,6],[6,9],[9,12],[12,15],k=3时各算法的响应时间对比如图5、6所示. 可以发现,随着结点数量上界的增长,查询时间也不断增长. 但是本文提出的GRE算法通过缩小候选集合数量,大幅缩减查询响应时间.

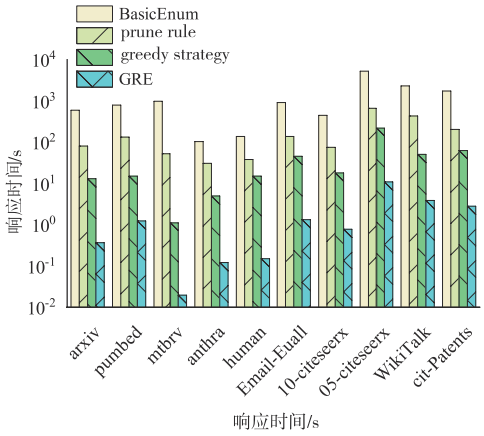

图4　[l, h] = [6,9], k = 3时在各数据图上的响应时间

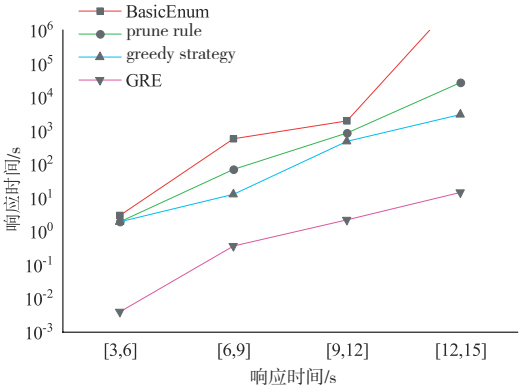

图5　arxiv数据图上不同社区规模的响应时间

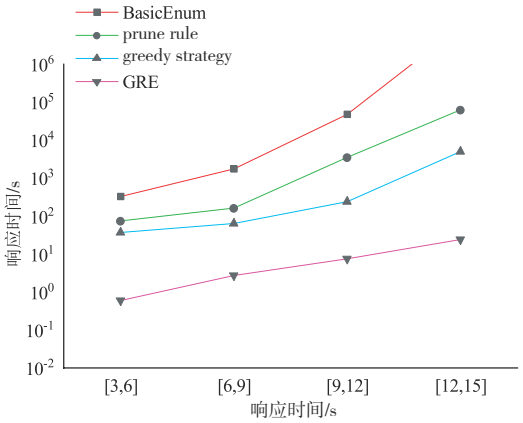

图6　cit-Patents数据图上不同社区规模的响应时间

综合来看,GRE算法找到的社区保证了其影响力最大化,同时确保了结点数量在题设要求的数量范围内,同时GRE算法相较于BasicEnum算法,在效率上获得了大幅提升.

6　总结

针对现有社区搜索方法仅关注社区结构,没有考虑到资源限制和社区属性特征的情况,本文提出了结点数量限制的影响力社区搜索问题,并设计了基于三

种优化策略的高效算法GRE. 该算法在满足用户个性化搜索需求的同时,可以快速返回满足条件的结果社区. 基于多个真实数据集的实验结果表明,在算法有效性方面,GRE优于现有的不考虑资源限制和社区属性特征的算法,能找到符合结点数量限制且影响力最大的社区. 在算法效率方面,GRE算法效率比基础的BasicEnum算法快100倍左右.

参考文献

[1] MAURO S, ARISTIDES G. The community-search problem and how to plan a successful cocktail party[C]//Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2010: 939-948.

[2] MA Y, YUAN Y, ZHU F, et al. Who should be invited to my party: A size-constrained *k*-core problem in social networks[J]. Journal of Computer Science and Technology, 2019, 34(1): 170-184.

[3] FANG Y, HUANG X, QIN L, et al. A survey of community search over big graphs[J]. The VLDB Journal, 2020, 29 (1): 353-392.

[4] WU Y, ZHAO J, SUN R, et al. Efficient personalized influential community search in large networks[J]. Data Science and Engineering, 2021, 6(3): 310-322.

[5] LUO W, ZHOU X, YANG J, et al. Efficient approaches to top-*r* influential community search[J]. IEEE Internet of Things Journal, 2021, 8(16): 12650-12657.

[6] FANG Y, YANG Y, ZHANG W, et al. Effective and efficient community search over large heterogeneous information networks[J]. Proceedings of the VLDB Endowment, 2020, 13(6): 854-867.

[7] AL-BAGHDADI A, LIAN X. Topic-based community search over spatial-social networks [J]. Proceedings of the VLDB Endowment, 2020, 13(12): 2104-2117.

[8] LIU B, ZHANG F, ZHANG W, et al. Efficient community search with size constraint[C]//2021 IEEE 37th International Conference on Data Engineering (ICDE). Piscataway: IEEE, 2021: 97-108.

[9] YAO K, CHANG L. Efficient size-bounded community search over large networks[J]. Proceedings of the VLDB Endowment, 2021, 14(8): 1441-1453.

[10] FANG Y, CHENG R, LUO S, et al. Effective community search for large attributed graphs[J]. Proceedings of the VLDB Endowment, 2016, 9(12): 1233-1244.

[11] CHEN C, ZHANG M, SUN R, et al. Locating pivotal connections: The K-truss minimization and maximization problems[J]. World Wide Web, 2022, 25(2): 899-926.

[12] LI C, ZHANG F, ZHANG Y, et al. Efficient progressive minimum *k*-core search[J]. Proceedings of the VLDB Endowment, 2019, 13(3): 362-375.

[13] BATAGELJ V, ZAVERSNIK M. An $O(m)$ algorithm for cores decomposition of networks[J/OL]. CoRR, 2003. http://arxiv.org/abs/cs.DS/0310049.

[14] LI R, QIN L, YU J, et al. Influential community search in large networks[J]. Proceedings of the VLDB Endowment, 2015, 8(5): 509-520.

[15] LIU Q, ZHU Y, ZHAO M, et al. VAC: Vertex-centric attributed community search[C]//2020 IEEE 36th International Conference on Data Engineering(ICDE). Dallas: IEEE, 2020: 937-948.

[16] XIE X, SONG M, LIU C, et al. Effective influential community search on attributed graph[J]. Neurocomputing, 2021, 444: 111-125.

[17] ERDŐS P, PACH J, POLLACK R, et al. Radius, diameter, and minimum degree[J]. Journal of Combinatorial Theory, Series B, 1989, 47(1): 73-79.

作者简介

杜 明 男,黑龙江鸡西人,博士,东华大学副教授,主要研究方向为图数据处理技术,自然语言处理等.
E-mail: duming@dhu.edu.cn

宋嘉祎 男,1997年9月出生于上海市. 东华大学计算机科学与技术学院硕士研究生,主要研究方向为社区挖掘技术.
E-mail: sshin_song@163.com

周军锋(通讯作者) 男,陕西西安人,博士,东华大学教授,博士生导师,主要研究方向为图数据处理技术、推荐系统关键技术等.
E-mail: zhoujf@dhu.edu.cn