doi:10.3969/j.issn.1005-3026.2015.05.001

利用多级社区中心标签实现大规模图上距离查询

张翼飞,王国仁,张恩德,赵长宽
(东北大学 信息科学与工程学院,辽宁 沈阳 110819)

摘 要:距离查询是图数据挖掘应用中的最基本的操作之一,但是目前的现存查询算法均无法高效处理大规模图数据.针对这个问题,提出建立多级社区中心的标签机制,即首先在原图中将结点按社区划分为多个集合,然后再将各集合中的中心结点建成带权查询子图,经过多次递归操作,最终为各结点建立一个基于社区中心的树状结构标签集,该标签集可以实现利用较短的创建时间和较小的存储代价大幅度提高距离查询的效率.从实验结果可以看出,该方法综合效率明显优于现存的高效算法.

关 键 词:多级社区中心;标签;大规模图数据;距离查询;带权查询
中图分类号:TP 311 **文献标志码**:A **文章编号**:1005-3026(2015)05-0609-05

Utilizing Multilevel Community Center Labels for Distance Querying in Large Graphs

ZHANG Yi-fei, *WANG Guo-ren*, *ZHANG En-de*, *ZHAO Chang-kuan*
(School of Information Science & Engineering, Northeastern University, Shenyang 110819, China. Corresponding author: ZHANG Yi-fei, E-mail: zhangyifei@sau.edu.cn)

Abstract: Distance querying is one of the most fundamental operations in many graph data mining applications. However, most of the previous methods cannot handle large graphs, especially those with more than a hundred thousand vertices. To solve this problem, a multilevel community center labels index structure was proposed. Firstly, the vertices of the original graph were divided into different communities. Then a weighted query sub-graph was constructed by each community center. Finally, a tree-like label set was built for every vertex. The query efficiency could be improved greatly with small time and storage cost. The experimental result showed that the overall efficiency of this approach is significantly better than those of the-state-of-the-art algorithms.

Key words: multilevel community center; label; large graphs; distance query; weighted query

近年来,越来越多的应用把它们的数据建模为图的形式,如XML数据库、社会网络、生物信息网络、交通网络等.随着这些应用的不断深入,使得这些数据的规模越来越大,如何高效地管理和挖掘这些数据,已经成为目前的热点研究问题之一.

距离查询[1],即回答图中任意两个结点间的距离,是图中最基本的操作之一.如在社会网络中可以判断两个用户间关系的亲近程度,在交通网络中则可以回答任意地点间的最短距离,其他很多应用中也大量涉及到距离计算.

由于实时在线查询和计算传递闭包两种极端方法或者在查询时间或者在索引存储无法满足可扩展性的要求,目前常用的方法是对原图创建压缩的索引.而相对于可达查询[1-4],即关注任意两点之间是否存在路径,距离查询因为需要在所有可达路径中的计算最短距离,其难度更大.目前有关距离查询的研究焦点一般集中于两类图数据:一类是交通路网;另一类是复杂网络,如社会网络、生物信息网络等.对于前者,因为结构相对简单,目前出现了很多成功的研究成果[5].而对于后者,目前仍然是很具挑战性的课题.虽然现在有

收稿日期:2014-04-03
基金项目:国家自然科学基金青年基金资助项目(61303016);辽宁省教育厅一般项目(L2012045).
作者简介:张翼飞(1976-),男,吉林白山人,东北大学博士研究生,沈阳航空航天大学副教授;王国仁(1966-),男,湖北崇阳人,东北大学教授,博士生导师.

一些处理复杂网络的方法,但是都面临着性能的瓶颈,即只能处理小规模图(结点数量在几千到几万之间),而大多数算法在处理大规模图时,性能会急剧下降,甚至无法运行.

标签法[1,3-4,6-8]是在建立图数据索引时常用的一种方法,即为图中每个结点赋予一个标签,在进行可达查询或距离查询时,直接通过两个结点的标签集的 join - 查询即可快速获得结果. 目前大部分的算法都是直接在原图上创建索引[6,9],当图的规模增大到一定程度时,这些索引结构一般都无法具备良好的可扩展性. Jin 等[7]参考交通路网算法提出 Highway 标签结构,可以处理中等规模甚至大规模稀疏图,但不是所有的图都可以高效地提取出 Highway 结构.

文献[2,10]中提出利用创建通用查询子图的方法来提高其他算法的执行效率,即在原图中仅选取部分结点作为查询骨架,在进行可达查询或距离查询时,每个结点只需快速找到离自己最近的骨架结点,然后利用子图实现与目标结点信息的查询. 这两个方法都具有一定的可扩展性,但是它们创建的查询子图本身对现有算法来说仍然可能过大而无法提高其效率.

总的来说,可扩展性已经成为影响距离查询效率的一个最重要障碍,目前急需解决的问题是当图的规模达到十几万甚至上百万结点的时候,如何建立一个高效的索引结构以实现高效查询. 基于此,本文提出了利用社区划分的方法在原图中建立查询子图,称为查询主干,该主干保留了原图的所有拓扑信息,而且该算法可以递归进行,建立起一个多级标签索引结构,利用该结构既提高了查询速度,也大幅度降低了创建时间和索引大小. 实验结果表明,本方法不仅适用于小规模的图数据,对于大型和超大型图数据也具有非常好的可扩展性.

1 相关概念介绍

本文研究建模为图的形式的网络数据,设 $G=(V,E)$ 为有顶点集 V 和边集 E 的图,用符号 n 表示图的顶点数量 $|V|$,m 代表边的数量 $|E|$. 任意两点间的最短距离用 $d(u,v)$ 表示,如果两点之间没有路径则 $d(u,v)=\infty$. 用 $N_\varepsilon(v)$ 表示任意顶点 $v\in V$ 的 ε 邻居,即 $N_\varepsilon(v)=\{d(u,v)\leqslant\varepsilon \mid u\in V,(u,v)\in E\}$.

根据迪杰斯特拉最短路径算法思想,任意两点间的最短距离满足三角形不等式,对于图中任意三个连通结点 u,v 和 s:

$$d(u,v)\leqslant d(u,s)+d(s,v)\ , \tag{1}$$

$$d(u,v)\geqslant |d(u,s)-d(s,v)|\ . \tag{2}$$

问题定义:给定一个图 G,如何创建一个索引结构以高效回答任意两点间的距离. 为了方便描述及与其他算法对比,本文主要讨论如何处理无向无权图,目前大多数的应用可以建模为这种形式,而且该方法也可以很容易地扩展到有权图或有向图的应用中.

直观地说,一个图的查询主干应该具备以下几个特征:1) 它的规模必须远远小于原图;2) 它必须保留原图所有的拓扑结构和距离信息;3) 原图中的任一结点都能在少数几步到达主干中的某一结点.

为了达到这个目标,本文引入了社区划分的概念. 根据 Milgram 的小世界理论[11-12],社区聚集是很多网状数据的共有现象,如图 1 所示的一个社交网络,所有用户根据兴趣、职业等特征划分为若干个集合,在集合内部各结点联系紧密,而在集合之间联系则很少. 本文利用了这个现象,首先将原图划分为若干个社区,其中每个社区包括一个与社区内所有结点连接的社区中心,而其他结点为该中心的 ε 邻居.

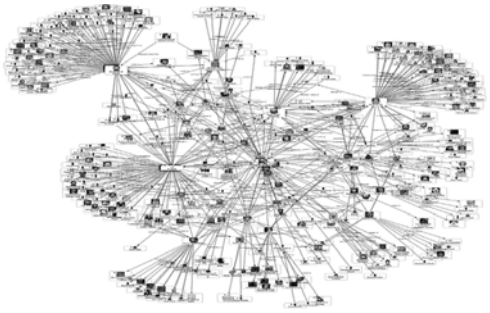

图 1 社交网络示意图
Fig. 1 A real social network

综上,查询主干可定义如下:

定义 1 给定一个图 $G=(V,E)$ 和一个阈值 ε,则它的查询主干为 $G^*=(V^*,E^*)$,其中 $V^*\in V$,E^* 通常包括 E 中不存在的边,E^* 中的权值为原图中两点最短距离.

查询主干中的结点应该具有如下性质:1) 原图中任意一个非主干结点都可以至多在 2ε 步内到达查询主干中的一个结点;2) 查询主干中的两个相邻结点的最大步长至多为 $2\varepsilon+1$ 步.

图 2b 为原图 2a 的一个查询主干,从该例可以看出,查询主干的结点数量远远小于原图中结点的数量,该子图可以继续进行查询主干的创建操作,直至结点个数为 1 为止.

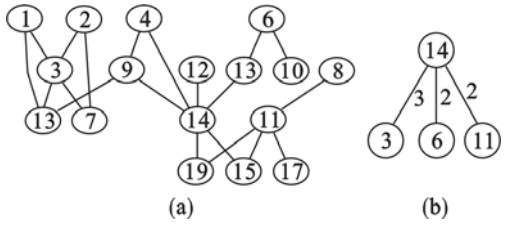

图 2　原图及其查询主干

Fig. 2　The original graph anf query trunk

(a)—原图；(b)—查询主干.

很显然，本问题的核心是如何在图中找到尽可能少的社区中心以覆盖原图中的所有结点，即创建尽可能小的索引，其定义如下：

定义 2　最小查询主干(MQT，minimum query trunk)：给定图 $G=(V,E)$，则必存在一个查询主干 $G^*=(V^*,E^*)$，在所有的可能集合中 $|V^*|$ 最小.

但是这个问题无法在多项式时间内解决，因为社区发现本身就是一个 NP－hard 问题，因此本文提出了一个简单但是非常有效的启发式算法，可以快速在一个图中建立查询主干并创建索引.

2　算法描述

2.1　社区中心发现

如果原图中的任意结点都可以在 2ε 步内到达主干 T 中的某一结点，则称 T 对原图完全覆盖. 最理想情况下，先将原图划分为多个社区，然后再从各社区中确定社区中心建立查询子图，但这个方法本身也是 NP－hard 问题，所以本文提出一个基于度最大结点优先覆盖算法，其基本思想和原因如下：1) 从图 1 中可以发现，处于社区中心位置的结点往往是与其他结点关系最为紧密的，具体表现就是结点的度往往远大于其他普通结点；2) 如果先确定社区再确定中心，问题会非常复杂，而如果先确定中心再覆盖附近的邻居结点，则效率会明显提高.

利用这两点发现，本文提出了如算法 1 和算法 2 所示的基于度最大优先的启发式算法. 这里还需给出一个有关有权图查询主干发现的定理.

定理 1　给定一个边上权值为非负的无向图 $G=(V,E)$，基于它的最小生成树生成的查询主干可以实现对图 G 的完全覆盖，证毕.

证明：根据最小生成树的定义，树中任意两个顶点的边都是原图中无环情况下权值最小的，因此计算 G 中任意两个顶点的距离所需步长一定小于或等于在树中两个顶点的步长，即通过最小生成树计算得到的查询主干可以实现对原图 G 的完全覆盖，证毕.

算法 1 给出了创建图 G 查询主干和为 G 中结点设置标签的过程. 1) 如果 G 是原图，则直接对 G 中所有结点按度进行排序，否则根据算法 1，首先生成 G 的最小生成树 G'(3～5 行)；2) 每次从当前未被访问集合中选择度最大的结点覆盖其周围所有 ε 邻居，直至所有结点都被访问为止(6～11 行)；3) 依次从查询主干 T 中的每个顶点出发，在 $2\varepsilon+1$ 的范围内进行宽度优先遍历(BFS)，对于该范围内的任一结点 v，如果同为查询主干结点，则在 E^* 中增加该边(12～16 行)；4) 如果 u 与 v 的步长为 $2\varepsilon+1$ 但是 v 不是主干中结点，这种情况称为主干未完全覆盖原图，这将造成一些路径信息无法正确计算，为了避免这种情况发生，将这类结点也加入主干中，以实现完全覆盖(17～19 行)；5) 对于其他非主干结点，则将该顶点 u 设为 v 的标签(20～21 行).

```
算法1：getTrunk(G,ε,L):计算查询主干T，并为G中非查询主干结点设置标签
input:G=(V,E),无向图；ε，阈值；
ouput:T=(V*,E*);L,结点标签；
1 V*←φ；
2 E*←φ；
3 if G是查询子图 then
4 ⌊ G′ ←G的最小生成树；
5 D←{将G或G′ 中所有结点按度排序}
6 for each u∈D do
7 |  if !visited[u] then
8 |  |  T←{u};
9 |  |  visited[u]=true;
10 |  |  for each v∈N_ε(u)do
11 |  ⌊  ⌊  visited[v]=true;
12 for each u∈V*do
13 |  V′ ←N_{2*ε+1}(u);
14 |  for each v∈V′ do
15 |  |  if v∈V*then
16 |  |  ⌊  E*←{(u,v,c)};//c为u与v之间的最短距离
17 |  |  else if u与v的步长为2*ε+1 then
18 |  |  |  V*←v;
19 |  |  ⌊  E*←{(u,v,c)};
20 |  |  else
21 |  ⌊  ⌊  L(v)←{(u,c)}.
```

算法 2 为递归生成查询子图，为 G 中所有结点设置标签的过程. 它依次将当前生成的查询主干作为输入，循环生成查询子图，直至 T 中只剩一个结点为止.

```
算法2：递归生成查询子图，为G中所有结点设置标签
input:G=(V,E),无向图；ε，阈值；
output:L,结点标签；
1 T=getTrunk(G,ε,L);//计算G的查询主干T=(V*,E*)
2 while|V*|>1 do
3 ⌊ T=getTrunk(T,ε,L).
```

2.2　查询处理

在介绍利用可达主干进行可达查询之前，首先给出一个性质：给定图 $G=(V,E)$，如果任意两

个结点 $u,v(u\in V,v\in V)$ 可达,则它们或者在 2ε 步内可达,或者通过某一级区域中心结点可达.

根据查询主干特点,该性质是显而易见的.

算法 3 给出了通过多级社区中心标签进行距离查询的基本框架. 该过程主要包括两个基本步骤:1) 如果 u 和 v 有共同的社区中心,则两者的距离可以通过局部搜索或者与相同社区中心之和计算;2) 如果 u 和 v 没有共同社区中心,则可以逐级向上查找直至产生交集或者查找到最高级中心为止.

```
算法3: 获得任意两个结点距离
  input:G=(V,E),无向图; ε, 阈值; u和v,
  图中任意两个结点;
  output:d,两个结点间距离;
1 d=∞;
2 S1←L(u);
3 S2←L(v);
4 S=S1∩S2;
5 while S≠φ do
6   for each u' ∈S1 do
7     S1' ←L(u');
8   for each v' ∈S2 do
9     S2' ←L(u');
10  S=S1' ∩S2';
11 if S≠φ then
12   d=localSearch(u,v);
13   if d=∞ then
14     for each t∈S do
15       if d>D1(t)+D2(t) then
16         d=D1(t)+D2(t);
17 return d.
```

2.3 复杂度分析

对于查询主干创建,主要时间代价为对图中结点的排序、临近结点的遍历及最小生成树的建立. 其中排序时间代价为 $O(\sum k\lg k)$,其中 k 的初值为 n,其他为各级查询主干结点个数,根据实验证明本算法收敛速度非常快,对于大型规模的图数据也仅需要运行少数几级即可创建所有结点的标签. 对于临近结点遍历的代价为 $O(\sum(k+e))$,其中 k 为各级主干结点个数,初值为 n;e 为各级主干边的个数,初值为 m. 存储代价为 $O(kn)$,其中 k 为各结点到达社区中心覆盖的个数,因为图数据的差异,无法确定 k 的规模,但是从实验结果看,绝大多数结点被覆盖的中心规模是可控的,对查询效果影响不大,尤其是稀疏图效果更明显.

对于查询效率,主要通过遍历查找结点标签以及计算两个结点标签集的交集实现距离计算. 根据结点标签的性质,这种遍历类似于树的遍历,所以最坏情况下的时间复杂度为 $O(k+s)$,k 和 s 分别为两个结点各自生成的树状标签集的大小.

3 实验结果及分析

为了测试本算法的效率,利用一系列实验进行了验证,实验平台为一台高性能 Windows 服务器,处理器为 Intel Xeon 2.10 GHz ,主存为 16 GB. 全部算法采用 C++ 语言实现. 在所有的实验中,统一限定 ε 值为 2.

在实验中,重点测试 3 个重要指标:索引创建时间、索引大小和查询时间. 对于查询时间,通过随机生成的 1×10^5 次查询的合计时间来进行评价. 为了测试算法的综合运行效率,利用随机生成的 7 组人工数据来对各算法进行综合对比,其结点数量从 1×10^3 至 2×10^7.

本算法的参照对象是由 Akiba 提出的 Pruned Landmark Labeling 算法[6](简称 PLL),该算法利用可剪枝的 BFS 来为原图的所有结点创建标签式索引,图 3 和图 4 给出了两个算法的索引创建时间和索引大小. 从实验结果可以看出,本算法执行时间和索引规模都远远小于 PLL 算法,而当图结点规模大于 2×10^6 时,PLL 算法已经无法正常运行,而本算法仍然可以正常运行.

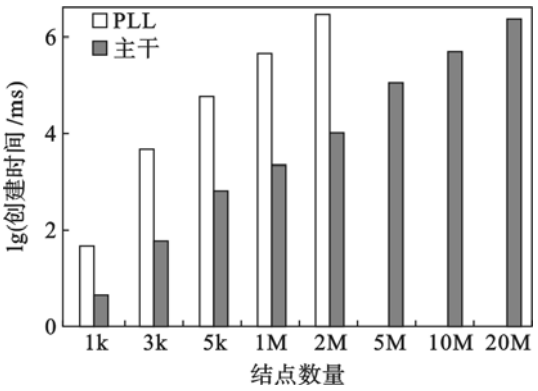

图 3 创建时间结果图

Fig. 3 The results of construction time

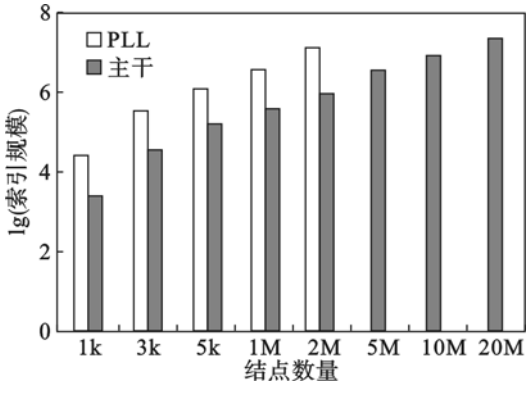

图 4 索引大小结果图

Fig. 4 The results of index size

图 5 给出算法的查询时间，当结点规模较小时，PLL 的查询时间趋近于 0，所以本文没有给出其查询时间实验结果. 但从图中可以看出，虽然本文的算法没有实现常数时间查询，但是与极端算法 BFS 相比，大幅度降低了查询时间.

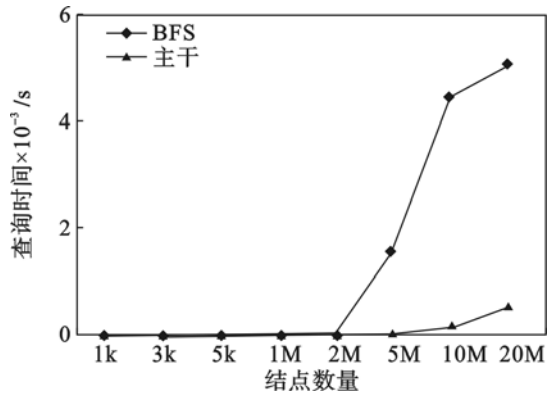

图 5　查询时间结果图

Fig. 5　The results of query time

有关 ε 取值问题，在实验中发现当 $\varepsilon = 2$ 时，原图结点的数量已经被压缩的非常可观，通常情况下，不建议设置 $\varepsilon > 4$. 因为这样查询主干的规模虽然很小，但是会大幅度降低查询效率，在某些极端情况下，已经与 BFS 的查询效率相当.

从实验结果可以看出，本文算法在索引创建时间、索引大小和查询效率等方面比现有算法都有较大提高，并可以实现对大规模图数据的高效处理.

4　结　　语

本文针对目前大规模图数据距离查询效率较低，现有方法可扩展性较差的问题，提出了多级社区中心标签的方法. 该方法可以实现较低的创建时间、较少的存储代价和较高的查询效率，尤其对于大规模及超大规模图数据的处理，本文给出了一个比较可行的解决思路. 另外，本算法的局部特性也非常适合分布式图数据的处理，今后将把研究重点放到这类数据的高效处理中.

参考文献：

[1] Aggarwal C C, Wang H X. Managing and mining graph data [M]. London: Springer-Verlag, 2010: 181 – 216.

[2] Jin R M, Ruan N, Dey S, et al. SCARAB: scaling reachability computation on large graphs [C]//ACM International Conference on Management of Data. Scottsdale, 2012: 169 – 180.

[3] Jin R M, Xiang Y, Ruan N, et al. Path-tree: an efficient reachability indexing scheme for large directed graphs [J]. *ACM Transactions on Database Systems*, 2011, 36(1): 1 – 44.

[4] Wang H X, He H, Yang J, et al. Dual labeling: answering graph reachability queries in constant time [C]//International Conference on Data Engineering. Munich, 2006: 961 – 979.

[5] Bast H, Funke S, Sanders P, et al. Fast routing in road networks with transit nodes [J]. *Science*, 2007, 316(5824): 316 – 566.

[6] Akiba T, Iwata Y, Yoshida Y. Fast exact shortest-path distance queries on large networks by pruned landmark labeling [C]//ACM International Conference on Management of Data. New York, 2013: 349 – 360.

[7] Jin R M, Ruan N, Xiang Y, et al. A highway-centric labeling approach for answering distance queries on large sparse graphs [C]// ACM International Conference on Management of Data. Scottsdale, 2012: 445 – 456.

[8] Cohen E, Halperin E, Kaplan H, et al. Reachability and distance queries via 2-hop labels [J]. *SIAM Journal on Computing*, 2003, 32(5): 1338 – 1355.

[9] Wei F. Tedi: efficient shortest path query answering on graphs [C]//ACM International Conference on Management of Data. Indianapolis, 2010: 99 – 110.

[10] Zhang Y F, Wang G R, Zhao C K, et al. Utilizing community centers to answer reachability queries for large graphs [C]// Web Information System and Application Conference. Yangzhou, 2013: 205 – 210.

[11] Milgram S. The small world problem [J]. *Psychology Today*, 1967, 67(1): 60 – 67.

[12] Zhang Y F, Wang G R, Zhao C K, et al. SPTI: efficient answering the shortest path query on large graphs [C]// IEEE International Congress on Big Data. Santa Clara, 2013: 195 – 202.