第46卷第3期 中国科学技术大学学报 Vol. 46, No. 3
2016年3月 JOURNAL OF UNIVERSITY OF SCIENCE AND TECHNOLOGY OF CHINA Mar. 2016

文章编号:0253-2778(2016)3-0180-08

一种有效的加权图聚集算法

胡宝丽,游进国,周翠莲,王 洋,崔红波

(昆明理工大学信息工程与自动化学院,昆明 650500)

摘要:图聚集(图概括)技术是解决大规模网络的有效技术之一.现实生活中,这些图不仅规模大,而且边可能带有权重,当前图聚集算法很少或未考虑边的权重或边存在的概率等信息,导致聚集图与原图的误差大.为了提高加权图的图聚集的质量和效率,对加权图的图聚集算法进行了研究.为此引入超图邻接矩阵分组的权重值一致性来衡量边权重的一致性,定义压缩率衡量图聚集算法的空间效率,使用误差率衡量聚集图与原图的误差;通过控制图的误差率来控制图的压缩质量,并与现有图聚集算法进行了对比.实验论证了本文图聚集算法的有效性.

关键词:图数据; 加权图; 图聚集; 图概括; 压缩率

中图分类号:TP311 **文献标识码**:A doi:10.3969/j.issn.0253-2778.2016.03.002

引用格式:HU Baoli, YOU Jinguo, ZHOU Cuilian, et al. An efficient weighted graph aggregation algorithm[J]. Journal of University of Science and Technology of China, 2016,46(3):180-187.
胡宝丽,游进国,周翠莲,等. 一种有效的加权图聚集算法[J]. 中国科学技术大学学报,2016,46(3):180-187.

An efficient weighted graph aggregation algorithm

HU Baoli, YOU Jinguo, ZHOU Cuilian, WANG Yang, CUI Hongbo

(*Kunming University of Science and Technology Faculty of Information Engineering and Automatic* 650500)

Abstract: Graph aggregation (graph summarization) technique is one of the effective ways to mine and analyze huge graphs. However, in reality, these graphs are not only huge but also carry weighted edges. The current algorithms do not or seldom take the weight into consideration, leading to a great difference between the aggregation graph and the original one. In order to solve this problem and improve the quality and efficiency of graph aggregation, The weighted graph aggregation algorithm was studied, the consistency of grouping area values of the adjacent matrix of the aggregation graphs was introduced to measure the consistency of weights of edges, compression ratio was defined to measure the spatial efficiency of the graph aggregation algorithm, and error rate was used to evaluate the difference between the aggregation graph and the original graph. The compression quality is ensured by controlling error rates and a comparison is made between the proposed algorithm and the existing graph aggregation algorithms. The experiment results show the effectiveness of the graph aggregation algorithm.

Key words: graph data; weighted graph; graph aggregation; graph summarization; compression ratio

收稿日期:2015-08-27;**修回日期**:2015-09-29

基金项目:国家自然科学基金(61462050)、云南省自然科学基金(2013FZ020,KKSY201303095)资助.

作者简介:胡宝丽,女,1989年生,硕士生.研究方向:数据挖掘. E-mail: 18213920251@163.com.

通讯作者:游进国,博士/副教授. E-mail: jgyou@126.com

0　引言

在现实生活的大多数应用中，经常用图数据来描述实体之间的关系，如社交网络用户之间的好友关系、城市与城市之间形成的交通网络关系、蛋白质之间两两交互关系等. 随着这些领域的发展和数据的增加，目前对于图的研究相当广泛，为了节省图存储空间、提高图聚集算法的效率，图聚集技术成为了研究热点[1-3]. 所谓的图聚集技术是将原图中的顶点和边的集合抽象到一个更小范围的、高度概括原图结构或属性信息的层次，从而获得一个简洁的超图. 在超图中，顶点称之为超级顶点(简称超点)，它代表原图中的一组属性相似或是结构相似的顶点；其边称之为超级边(简称超边)，它代表其关联的两个超点之间存在的边.

图聚集与图聚类的相似点在于它们均对顶点进行分组. 区别在于图聚类侧重寻找局部稠密的结构；而图聚集从全局出发，侧重寻找属性或者结构相似的顶点，它不仅概括稠密的结构，还概括稀疏的结构. 图聚集和图聚类之间详细的比较见表 1.

表 1　图聚集与图聚类之间的比较

Tab. 1　Comparison between graph aggregation and graph clustering

	分组顶点连通性	是否概括稠密或稀疏结构	分组的视角	输出是否为图
图聚集	不要求	均可	整体	是
图聚类	高	稠密	局部	不要求

现有的图聚集技术可以分为[4]：①基于属性一致性的图聚集：一般与 OLAP 技术相结合，代表技术为 Graph OLAP/Cube[5]；但基于属性一致性的图聚集仅针对顶点的属性进行分组，无法回答基于原图结构的查询. ②基于结构一致性的图聚集：根据研究的重点不同可分为基于准确性的图聚集以及基于压缩率的图聚集. 其中，Grass[6] 是基于准确性的图聚集，*S*-Node 是基于压缩率的图聚集，*S*-Node 的改进方法是基于 MDL[7] 原则构建超边的图聚集；但基于结构一致性的图聚集仅针对顶点的结构进行分组，无法回答基于原图属性的查询. ③基于属性和结构的图聚集[8]：当前基于属性和结构的图聚集，存在两种不同的做法，大多数是先利用属性一致性进行分组，再利用结构一致性划分属性一致的分组；另一种则将属性信息转换为结构信息，使用同一个度量函数同时度量两者，进而直接获取最终聚集图. 其中代表性的图聚集算法有：基于信息论中熵模型[9-10] 的图聚集技术，基于 SNAP 方法的图聚集技术，基于 *k*-SNAP[11] 方法的图聚集技术.

在许多应用领域，图中的关系是带权重的，而且权重对于图的分析和运用是至关重要的[12]. 例如，社交关系网络中，边的权重可以表示两好友之间的联系频度，从而可以看出两个人关系的密切程度. 带权重的图聚集不同于普通的图聚集或者图划分，是通过节点相互之间的关系来找到强联系的节点分组. 文献[13]介绍了带权重图的聚集方法，文中采用了蛮力贪婪算法(brute-force greedy algorithm)、阈值算法(thresholded algorithm)、随机半贪婪算法(randomized semi-greedy algorithm)三种算法来衡量带权重图压缩的时间复杂度和空间复杂度. 其中，在寻找压缩点对的过程中，使用了 2-hop 方法和随机方法找点对. 文献[13]的结果表明：带权重的图是可以在误差率很小的情况下，在很大程度上被压缩；带权重的图可以进行有效地压缩[13]. 该论文在进行图聚集的时候只考虑了边的权重信息，没有考虑到边是否存在以及存在概率的相关问题.

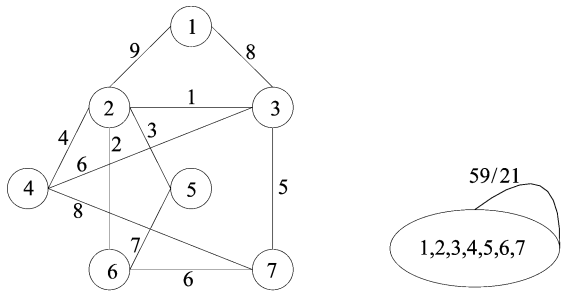
本文使用超图邻接矩阵分组的权重值一致性来衡量带权重图的边权重的一致性，并在图聚集过程中计算超边实际存在的概率，这样在图聚集的过程中可以考虑到边的权重和边存在概率两方面的信息. 首先，找出图中所有 2-hop 点对并计算每个点对超图邻接矩阵分组的权重值一致性的值；然后，利用超图邻接矩阵分组的权重值一致性对图聚集的每个步骤进行度量，取其值最小的点对进行合并；最后，利用压缩率、误差率评价图聚集算法的优劣.

1　模型

1.1　问题引入

给定一个无向图 G，当顶点之间的边不带权重时，图可以形式化表示为 $G=(V,E)$[14]；当顶点之间的边带权重时，图可以形式化表示为 $G=(V,E,W)$. 如图 1(a)所示，顶点 1～7 表示社交网络中 7 个不同的人，顶点之间的边表示人与人之间在社交网络中是好友关系，边上的权重表示好友关系的亲密程度，权重值越大说明两个人的关系越密切. 使用文献[13]中 $\lambda=1$ 的半贪婪算法(semi-greedy-hop2 algorithm)进行图聚集，在不控制压缩率的情况下，聚集结果如图 1(b)所示. 超边的权重

值取原图中边的权重的平均值. 用这种方法进行带权重图聚集时,只考虑到了边的权重信息,没有考虑到边是否存在以及存在概率的相关问题. 这种方法下的聚集图压缩程度高,但是还原后与原图的误差较大. 为了解决这个问题,本文提出了新的加权图的图聚集方法. 本文对加权图的聚集图的期望包括:①聚集图简洁,节约存储空间;②通过聚集图尽可能完整还原到原图(即聚集图最大限度地保留原图的结构信息和权重信息).



(a) 加权图 (b) 图 (a) 聚集图

图 1 加权图

Fig. 1 Weighted graph

1.2 加权图聚集

图 G 是一个无向、带权重的图,形式化表示为 $G=(V,E,W)$,其中 $V=(v_1,\cdots,v_n)$ 指的是图 G 中所有顶点的集合;$E\subseteq\{(v_i,v_j)\mid v_i,v_j\in V\}$ 包含两点之间有边的所有顶点对集合;$W=(w_{11},\cdots,w_{ij},\cdots,w_{m})$ 指的是图 G 中顶点对 (v_i,v_j) 之间边的权重值的集合. 图 G 的邻接矩阵 $\boldsymbol{W}=\bigcup_{i=1}^{|v|}\bigcup_{j=1}^{|v|}w_{ij}$,其中 w_{ij} 对应着顶点对 (v_i,v_j),w_{ij} 的值表示顶点对 (v_i,v_j) 之间边的权重. 如果两点之间存在边则 w_{ij} 的值为边上存在的权重值;否则 w_{ij} 为 0.

定义 1.1 给定图 G_s 是图 $G=(V,E,W)$ 的超图,形式化表示为 $G_s=(V_s,E_s,W_s)$,其中 $V_s=\{V_1,V_2,\cdots,V_k\}$,则图 G_s 满足以下条件:

① $V_i\in V_s,V_i\in V,V_i\neq\varnothing$;

② $\bigcup_{i=1}^{k}V_i=V,V_i\cap V_j=\varnothing$;

③$E_S=\{(V_i,V_j)\mid\exists u\in V_i,v\in V_j,(u,v)\in E\}$;

④$W_s=(P;W)$.

其中,P 表示超点之间超边存在的概率,根据超边所代表的是否是超点内部顶点之间的边,将超边存在的概率分为以下两种不同的计算方法:

(Ⅰ)超点之间存在超边的概率

超点间可能存在的边总数 $\Gamma_{ij}=|V_i||V_j|$;

实际存在的边的总数 $E_{ij}=\{|V_i,V_j||(V_i,V_j)\in E_s,W_s\neq\varnothing\}$;

超点之间存在边的概率 $P=E_{ij}/\Gamma_{ij}$.

(Ⅱ)超点所代表的内部点间的连接概率

超点内部顶点间可能存在的边总数 $\Gamma_{ii}=C^2_{|V_i|}=|V_i|(|V_i|-1)/2$;

实际存在的边的总数 $E_{ij}=\{|V_i,V_i||(V_i,V_i)\in E_s,W_s\neq\varnothing\}$;

超点内部存在边的概率 $P=E_{ii}/\Gamma_{ii}$.

其中 $|V|$ 表示超点中顶点的个数;超边的权重值等于聚集图在原图中相应的所有边权重的中位数.

表 2 图 1(a)的邻接矩阵 W

Tab. 2 The adjacent matrix W of Fig. 1(a)

	1	2	3	4	5	6	7
1	0	9	8	0	0	0	0
2	9	0	1	4	3	2	0
3	8	1	0	6	0	0	5
4	0	4	6	0	0	0	8
5	0	3	0	0	0	7	0
6	0	2	0	0	7	0	6
7	0	0	5	8	0	6	0

定义 1.2 给定 $\boldsymbol{M}$ 是图 $G=(V,E,W)$ 的超图邻接矩阵,$\boldsymbol{M}$ 由 C_{intra},C_{inter},C_{extra} 三部分分组权重值组成.

C_{intra} 表示要合并的点对之间的权重值区域,C_{inter} 表示要合并的点与邻接点之间的权重值区域,C_{extra} 表示要合并的点的邻接点之间的权重值区域.

例如:图 1(a)的邻接矩阵 $\boldsymbol{W}$ 如表 2 所示,假设在图 1(a)中要合并的点对是 5、6,则此时的超图邻接矩阵如表 3 所示.

表 3 图 1(a)的超图邻接矩阵

Tab. 3 The adjacent matrix of super graph of Fig. 1(a)

	5	6	2	7
5	0	7	3	0
6	7	C_{intra} 0	2	C_{inter} 6
2	3	2	0	0
7	0	C_{inter} 6	0	C_{extra} 0

定义 1.3　给定 $\boldsymbol{M}$ 是图 $G=(V,E,W)$ 的超图邻接矩阵，计算 $\rho=\rho_{\text{intra}}+2\rho_{\text{inter}}$ 其中，

$\rho_{\text{intra}}=\sqrt{\sum(w_{ij}-Me_{\text{intra}})^2}$，$i,j=0,1$；

$\rho_{\text{inter}}=\sqrt{\sum(w_{ij}-Me_{\text{inter}})^2}$，$1<i,j<2+$邻接点个数.

衡量不同超图邻接矩阵分组的权重值一致性，计算结果 ρ 越小，说明权重值的一致性越好，合并点对后的误差越小. C_{extra} 表示要合并点对的邻接点之间的权重值区域. 因为 C_{extra} 区域在点对合并过程中不受影响，不会产生变化，所以不用考虑这一部分权重的一致性. 不难看出，ρ_{intra} 和 ρ_{inter} 的计算式可以解释为超图邻接矩阵中权重值之间的欧氏距离，即可以用来衡量权重的一致性.

以图 1(a)为例，结构上先合并点对 5,6 与先合并点对 4,7 最终的聚集结果相同，聚集图如图 2 所示. 其中，边上的数字 11/21 表示两个超点之间存在边的概率.

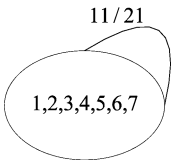


图 2　图 1(a)不考虑边权重的聚集图

Fig. 2　The aggregation graph of Fig. 1(a) when the weights of edges of Fig. 1(a) is not considered

考虑边权重以后，图聚集过程变得复杂. 因为权重值不同，合并点 5,6 与合并点 4,7 的结果可能会有所不同，通过以下计算过程来说明.

首先，合并点对 5,6 的计算过程如下：合并点对 5,6 的超图邻接矩阵如表 3 所示，表 3 中 C_{intra} 部分的权重值为 0,7,0,7,0 表示不存在边，所以不予以考虑。所以，权重值进行排序后的数列为：7,7，则其中位数为 $Me_{\text{intra}}=\dfrac{7+7}{2}=7$，故 $\rho_{\text{intra}}=\sqrt{(7-7)^2+(7-7)^2}=0$；

C_{inter} 部分的权重值分别为：3,2,6，进行排序后的数列为：2,3,6，则其中位数为 $Me_{\text{inter}}=3$，故 $\rho_{\text{inter}}=\sqrt{(3-2)^2+(3-3)^2+(3-6)^2}=3.162$；

则：$\rho_{56}=\rho_{\text{intra}}+2\rho_{\text{inter}}=0+2\times3.162=6.324$.

其次，合并点对 4,7，的超图邻接矩阵如表 4 所示. 计算过程同合并点对 5,6，则：$\rho_{47}=\rho_{\text{intra}}+2\rho_{\text{inter}}=0+2\times1.732=3.464$.

表 4　图 1(a)的超图邻接矩阵

Tab. 4　The adjacent matrix of super graph of Fig. 1(a)

	4	7	2	3	6
4	0	8	4	6	0
7	8	0	0	5	6
2	4	0	0	1	2
3	6	5	1	0	0
6	0	6	2	0	0

由以上计算可知，$\rho_{47}<\rho_{56}$，则说明点对 4,7 及其邻接点之间的权重值的一致性好，即权重值相近，合并点对后超图与原图的误差小；所以选择点对 4,7 进行合并，合并以后的超图如图 3 所示. 其中，边上括号中的数字 1/2 表示两个超点之间存在边的概率，4,6 分别表示超点之间超边的权重值.

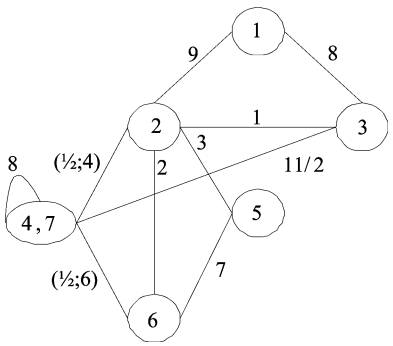


图 3　图 1(a)进行一步图聚集结果

Fig. 3　The aggregation result by one step of Fig. 1

1.3　图聚集质量评估

定义 1.4　给定图 $G_s=(V_s,E_s,W_s)$ 是图 $G=(V,E,W)$ 的超图，$|E|$ 表示图 G 边的数量，$|E_s|$ 表示超图 G_s 边的数量，则图 G 压缩率为 $\delta=\dfrac{|E|-|E_s|}{|E|}$.

以图 1(a)为例，$|E|=11$；$|E_s|=1$；则图 1(a)的压缩率 $\delta=\dfrac{11-1}{11}=\dfrac{10}{11}$. 图的压缩率 δ 越大，说明聚集图所占存储空间越小，图聚集的空间效率越好.

定义 1.5　设图 $G_s=(V_s,E_s,W_s)$ 是图 $G=(V,E,W)$ 的超图，$E_s=\{(V_i,V_j)\mid\exists u\in V_i,v\in V_j,(u,v)\in E\}$，其中，图 G 的超图邻接矩阵 $M=\bigcup_{i=1}^{|v|}\bigcup_{j=1}^{|v|}w_{ij}$，图 G_s 的超图邻接矩阵 W_s 对应的过渡矩阵 $M_S'=\bigcup_{i=1}^{|v|}\bigcup_{j=1}^{|v|}w_{ij}'$，则聚集图与原图的误差率

$$\varepsilon = \frac{1}{|E|^2}\sum_{i=1}^{|V|}\sum_{j=1}^{|V|}|w_{ij}^{'}-w_{ij}^{'}|^2.$$

误差率越小,说明图聚集的效果越好,即与原图的误差越小;所以通过控制误差率 ε 的大小,可以控制图聚集结果的质量. 以图 1(a)为例,对图 1(a)的进行图聚集:当 5,6 合并成一个超点,其进行图聚集前的超图邻接矩阵 $\boldsymbol{M}$ 如表 3 所示:合并点 5,6 以后的图聚集矩阵 $\boldsymbol{M}_s'$ 如表 5 所示.

表 5 合并点 5、6 后的超图邻接矩阵

Tab. 5 The adjacent matrix of super graph after merging node 5 and 6

	5	6	2	7
5	0	7	5/2	0
6	7	0	5/2	6
2	5/2	5/2	0	0
7	0	6	0	0

此时,超图的误差率是:$\varepsilon_1 = \frac{1}{11^2}\times\left\{\left(|\frac{5}{2}-3|^2+|\frac{5}{2}-2|^2\right)\times 2\right\}=0.008$

本文研究的加权图的图聚集问题表示如下:给定加权无向图 $G=(V,E,W)$,聚集图的最大误差率 ε,求图 G 的聚集图 $G_s=(V_s,E_s,W_s)$,在满足 ε 的条件下,使聚集图 G_s 对原图 G 的压缩率最大,能最大程度地还原到原图.

2 算法

为了解决加权图的图聚集的问题,本文提出了新的图聚集算法思想. 首先,使用产生点对算法 CNP(construct candidate node-pairs)对图 G 产生所有 2-hop 点对;然后,使用自底向上算法 BUS_WG(bottom up summary for weighted graph)实现加权图的图聚集.

2.1 产生点对算法

本文使用产生点对算法 CNP 产生所有 2-hop 的点对的分组,CNP 算法过程如算法 2.1 所示. 算法中根据点对的超图邻接矩阵分组的权重值一致性的计算值 ρ 将点对存入小根堆 Heap 中.

Algorithm 2.1 CNP(G)

Input: The graph $G=(V,E,W)$

Output: A Heap of node pairs

```
1 /* Initialization Stage */
2 for (i=0,i≤n,j++)
3     get node v_i
4     for each node v_i∈V get degree d
5     for (j =0;j<d; j++)
6       get neighbor[j]
7       if v_i≠ neighbor[j]
8         for(k=j+1; j < d; k ++)
9           get neighbor[k]
10          if (neighbor[k] neighbor[j] or neighbor
    [k]≠v_i)
11              if neighbor[k] < neighbor[j]
12               get node pair (neighbor[k],neighbor[j])
13               calculate ρ of node pair (neighbor[k],
    neighbor[j])
14                Heap←{ρ, (neighbor[k], neighbor
    [j])}
15            else
16               get node pair (,neighbor[j])
17               calculate ρ of node pair (neighbor[k],
    neighbor[j])
18                Heap←{ρ, (eighbor[k], neighbor
    [j])}
19          end if
20        end for
21        end if
22    end for
23 end for
```

2.2 加权图的图聚集算法

自底向上算法每次选取两个分组进行聚集,因此分组的候选集仅有 C_n^2 个. 自底向上算法思想进行图聚集时,广度优先遍历 Heap,选取超图邻接矩阵分组的权重值一致性计算值最小的分组,计算聚集后每条边的权重值,验证当前的聚集图与原图的误差率,验证是否满足阈值. 若满足,继续上述步骤;否则,输出聚集图 G_s 以及图的邻接矩阵 $\boldsymbol{A}_s$. 自底向上的图聚集算法如算法 2.2 所示:

Algorithm 2.2 BUS_WG (G,ε_{max})

Input: The matrix $A_{n\times n}$ of $G=(V,E,W)$

themaximum error rate ε_{max}

Output: The matrix A_s of aggregation graph G_s

```
1 / * Initialization Stage * /
2 For A_{n×n}->a[n+1][],store the degree of each node;
3 Get the Heap Heap=CNP(G);
4 Heap H=Φ, Vector Value=Φ;
5 / * do while loop get the smallest value and the node
  pair * /
6 while ε<ε_max do
```

```
7    while Heap is not NULL do
8     Get the top value of Heap;
9     Calculate ρ, error rate ε
10     Calculate super edge weight w
11     the according value of H←Heap, H←ρ;
12 end while
13 Get root node from H
14 Update Heap;
15 BUS_WG (G,εmax)
16 end while
17 Output aggregate graph Gs;
```

以图 1(a)为例,假设图聚集过程中最大误差率 $\varepsilon=1$,也即每进行一步图聚集,计算其聚集结果图的误差率 ε 并判断 ε 是否小于 1. 如果 $\varepsilon<1$,继续进行图聚集;反之,结束图聚集操作.

由定义 1.5 中的例子可知,第一步,图聚集合并点是点 4 和点 7,$\varepsilon_1=0.008$,$\varepsilon_1<1$,进行下一步. 同理,计算剩余步骤图聚集的误差率. 第二步,图聚集合并的点是点 2 和 3,计算其误差率 $\varepsilon_2=0.041$,$\varepsilon_2<1$,继续第三步,合并点 5 和点 6,计算其误差率 $\varepsilon_3=0.050$,$\varepsilon_3<1$,继续第四步,合并超点 4,7 和超点 5,6,计算其误差率 $\varepsilon_4=0.103$,$\varepsilon_4<1$,继续第五步,合并点 2 和点 7,计算其误差率 $\varepsilon_5=0.347$,$\varepsilon_5<1$,继续第六步,合并超点 1,4,5,6,7 和超点 2,3,计算其误差率 $\varepsilon_6=1.207$,$\varepsilon_6>1$,图聚集过程结束. 所以,图聚集的最终结果如图 4 所示:此时,误差率 $\varepsilon=0.347$.

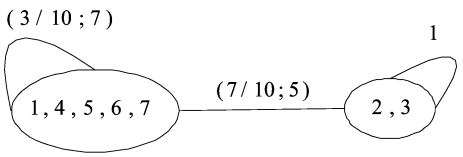


图 4　图 1(a)$\varepsilon=1$ 时的聚集图

Fig. 4　The aggregation graph of Fig. 1(a) when $\varepsilon=1$

当使用文献[13]中的 $\lambda=1$ 的半贪婪算法(semi-greedy-hop2 algorithm)对图 1(a)进行图聚集,图聚集过程为:第一步,点 2 和 3 合并成一个超点,合并后聚集图的误差率 $\varepsilon_1=0.355$,$\varepsilon_1<1$,继续第二步,合并点 4 和 7,合并后聚集图的误差率 $\varepsilon_2=0.756$,$\varepsilon_2<1$,继续第三步,合并点 5 和点 6,合并后聚集图的误差率 $\varepsilon_3=0.950$,$\varepsilon_3<1$,继续第四步,合并超点 4,7 和超点 5,6,合并后聚集图的误差率 $\varepsilon_4=1.587$,$\varepsilon_4>1$,图聚集过程结束. 其结果图如图 5 所示:误差率 $\varepsilon=0.950$.

由以上结果可知,当控制聚集图的最大误差率

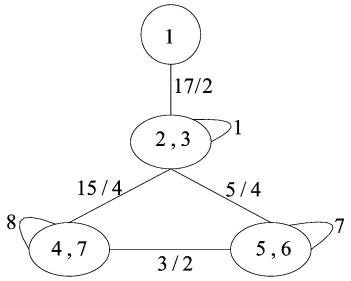


图 5　使用文献[13]中方法时图 1(a) $\varepsilon=3$ 时的聚集图

Fig. 5　The aggregation graph of Fig. 1(a) when $\varepsilon=3$ and use the method of Ref. [10]

时,本文方法压缩率 $\delta=\frac{11-2}{11}=\frac{9}{11}$ 大于文献[13]中的压缩率 $\delta=\frac{11-7}{11}=\frac{5}{11}$,并且在最大误差率 $\varepsilon_{max}=1$ 的情况下,本文方法的误差率较小,即聚集图与原图的误差率小,压缩率大,结果更优.

2.3　算法时间复杂度分析

CNP 算法的基本操作是算法 2.1 中的第 13 行和第 17 行,即找到所有具有公共顶点的顶点对,并计算每对点对的超图邻接矩阵分组的权重值一致性,该过程的时间复杂度为 $O(n^3)$.

自底向上算法中的基本操作是算法 2.2 中第 7 行 while 循环中的语句 9,10,即计算超图邻接矩阵分组权重值一致性的值、误差率、超边权重值. 当 Heap 为空时,基本操作的执行次数为 0;当 Heap 不为空时,基本操作执行的次数为 n 次. 语句 15 递归调用自底向上算法 BUS_WG,时间复杂度分析为 $T(n)=T(n-1)+O(1)$,从而推导其时间复杂度为 $O(n)$,因此,自底向上算法的时间复杂度为 $O(n^2)$,有效地将时间复杂度控制在多项式时间内.

3　实验结果及分析

3.1　实验环境

本文的算法是采用 C++实现的,实验平台是 C++开发工具 Visual Studio 2010 和图分析库 SNAP(Stanford network analysis platform). 所有实验都是在一台 PC 机上运行的,PC 机的配置如下:处理器 Intel 双核,2.4GHz,内存 4GB,Windows 7 操作系统.

3.2　实验数据

第一组数据集来自 DBLP 计算机科学文献的作者合作关系图(co-authorship graph). 图中顶点代表文献作者,边代表作者之间存在合著文献的关

系,边的权重值是大于 0 的,表示作者之间合作的次数,若不存在合作关系则边的权重值用 0 表示. 数据是以 dblp. xml 形式进行加载和使用的. 实验通过控制 DBLP 数据集中记录的条数来控制输入图的大小. 实验过程中使用的记录条数从 1 000 到 100 000 不等,实验相关数据如表 6 所示.

表 6 DBLP 实验数据

Tab. 6 The experiment data of DBLP

DBLP 记录条数	顶点数	边数	初始化 2-hop 点对数
10 000	18 396	36 361	115 184
30 000	46 392	81 695	271 126
50 000	75 689	141 004	516 985
70 000	99 691	191 297	784 426
100 000	136 804	283 370	1 319 647

第二组数据集是来自 arXiv 中的科学家之间的合作关系图(co-authorships between scientists). 图中的顶点代表科学家,边代表科学家之间存在的合作关系,边的权重值表示科学家之间的合作强度. 数据以 GML 格式进行加载和使用. 实验通过控制 arXiv 数据集中边的条数来控制输入图的大小. 实验中最大的边数是 60 000 条,此时,图中的顶点数为 11 899,初始化 2-hop 点对数是 757 016 对.

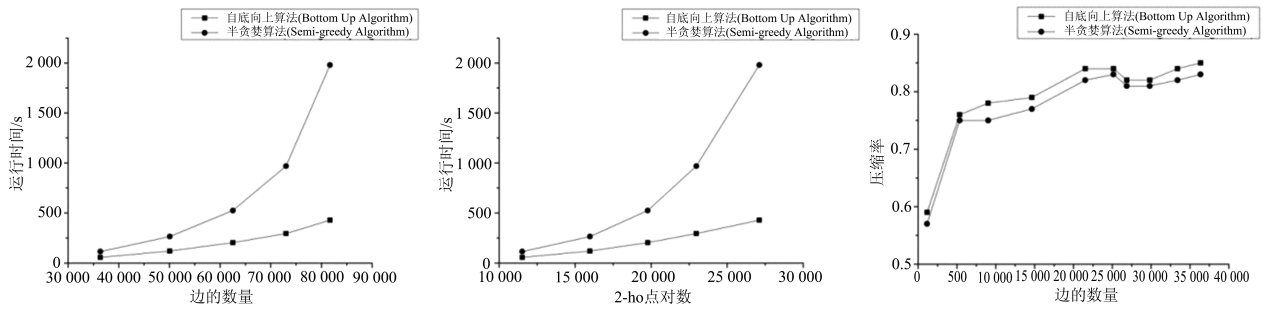
3.3 实验结果及分析

文献[13]中实现了带权重图聚集算法,本文选择 $\lambda = 1$ 的半贪婪算法进行对比,尽管它在时间效率上接近超线性,但是表现出来的效果较文献[13]中的其他算法要好[13].

本文将文献[13]中 $\lambda = 1$ 的半贪婪算法采用基于 SANP 的 C++实现,并且将 $\varepsilon = 0.01$ 情况下的运行时间、压缩率与自底向上算法进行对比.

由图 6(a)和图 6(b)可知,随着图中边的数量和初始时 2-hop 点对数的增加,运行时间越来越长. 随着图规模的逐渐增加,度数多的点的数量会增加,所以图聚集算法的运行时间不仅越来越长,而且时间的增长速度也越来越快. 由于图中边的数量与 2-hop 点对数量成正比,所以它们与算法运行时间的关系图也基本一致.

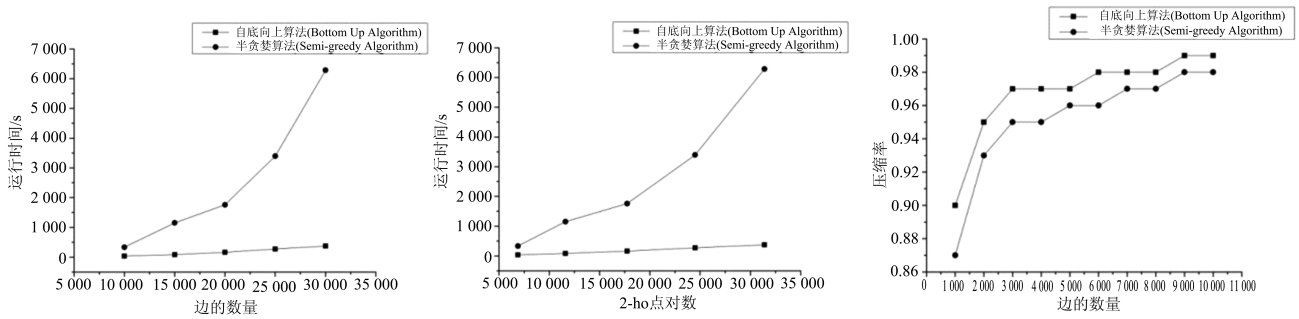
由图 6(a)和图 6(b)对比结果可知,当带权重图中边的数量和初始时 2-hop 点对数较少时,两者的运行时间相近;但是随着带权重图规模的增大,边的数量越来越多,半贪婪算法的时间增长得很快,相反自底向上算法的曲线则趋于平稳.



(a)原图中边的数量与运行时间的关系 (b) 原图中 2-hop 点对的数量与运行时间的关系 (c) 边的数量与压缩率关系

图 6 DBLP 数据集实验结果对比

Fig. 6 The experiment result of DBLP dataset



(a)原图中边的数量与运行时间的关系 (b) 原图中 2-hop 点对的数量与运行时间的关系 (c) 边的数量与压缩率关系

图 7 arXiv 数据集实验结果对比

Fig. 7 The experiment result of arXiv dataset

由图 6(c)可知，随着带权重图中边的数量的增多，图聚集算法对于原图的压缩率增加. 当边的数量大到一定程度，压缩率会大致稳定在一个范围. 本文的实验中，当边的数量超过 20 000 条时压缩率就稳定在 0.82 到 0.85 之间，说明自底向上算法对于图的压缩程度还是很大的. 同样由图 6(c)可知，与文献[13]中的半贪婪算法相比，压缩率要高.

在实验过程中，还测试了本文算法对更大规模的图的压缩效率和时间效率：使用 100 000 条 DBLP 的数据，其边数为 283 370，初始化 2-hop 点对数是 131 9647 对. 此时本文算法的压缩率为 0.93，运行时间为 7 985s.

对比图 6 和图 7，对比 DBLP 数据集和 arXiv 数据集实验结果发现：两种数据集实验结果的曲线趋势基本一致；但在 2-hop 点对相同的情况下，arXiv 数据集的压缩率大，运行时间长. 因为在 arXiv 数据集中度数大的顶点多，尽管边的数量相对来说较少，但由于图中聚集系数大，所以运行时间长，压缩率大.

4　结论

本文研究的是加权图的图聚集，用超图邻接矩阵分组的权重值一致性来衡量边权重的一致性. 这样在图聚集的过程中就既可以考虑到边的属性一致性信息，又可以考虑到边存在概率的问题. 本文首先找出图中所有 2-hop 的点对并计算它们之间的超图邻接矩阵分组的权重值一致性；然后利用超图邻接矩阵分组的权重值一致性对图聚集的每个步骤进行度量，找出超图邻接矩阵分组的权重值一致性值最小的点对进行合并；最后利用压缩率和误差率评价图聚集算法的优劣. 实验证明，对于图数据，本文提出的自底向上算法的时间效率相对于文献[13]中的 $\lambda=1$ 的半贪婪算法有效，在相同误差率的情况下压缩效率提高了 1%～3%；时间效率提高了一倍.

参考文献(References)

[1] LOUATI A, Aufaure M A, Lechevallier Y. Graph Aggregation: Application to Social Networks[A]. Advances in Theory and Applications of High Dimensional and Symbolic Data Analysis. 2013: 157-177.

[2] 冯国栋，肖仰华. 大图的分布式存储[J]. 中国计算机学会通讯，2012，8 (11): 12-15.

[3] 尹丹，高宏，邹兆年. 一种新的高效图聚集算法[J]. 计算机研究与发展，2011，48(10): 1831-1841.
YIN Dan, GAO Hong, ZOU Zhaonian. A novel efficient graph aggregation algorithm[J]. Journal of Computer Research and Development, 2011, 48(10): 1831-1841.

[4] 潘秋萍，游进国，张志朋，等. 图聚集技术的现状与挑战[J]. 软件学报，2015，26(1)：167-177.
PAN Qiuping, YOU Jinguo, ZHANG Zhipeng, et al. Progress and challenges of graph aggregation and summarization techniques[J]. Journal of Software, 2015, 26(1): 167-177.

[5] CHEN C, YAN X F, ZHU F D, et al. Graph OLAP: Towards online analytical processing on graphs[C]// Proceedings of the 8th International Conference on Data Mining. Pisa, Italy: IEEE Press, 2008: 103-112.

[6] LEFEVRE K, TERZI E. GraSS: Graph structure summarization[C]// SIDM International Conference on Data Mining. Columbus, USA: IEEE Press, 2010: 454-465.

[7] NAVLAKHA S, RASTOGI R, SHRIVASTAVA N. Graph summarization with bounded error [C]// Proceedings of the ACM-SIGMOD International Conference on Management of Data. Vancouver, Canada: IEEE Press, 2008: 419-432.

[8] ZHOU Y, CHENG H, YU J X. Graph clustering based on structural/attribute similarities [J]. Proceedings of the VLDB Endowment, 2009, 2(1): 718-729.

[9] LIU Z, YU J X. On summarizing graph homogeneously [C]// Proceedings of the 16th International Conference on Database Systems for Advanced Applications. Hong Kong, China: IEEE Press, 2011: 299-310.

[10] LIU Z, YU J X, CHENG H. Approximate homogeneous graph summarization [J]. Journal of Information Processing, 2011, 20(1): 77-88.

[11] ZHANG N, TIAN Y, PATEL J M. Discovery-driven graph summarization[C]// Proceedings of the 26th International Conference on Data Engineering. Long Beach, USA: IEEE Press, 2010: 41(3): 880-891.

[12] FAN T F, LIAU C J. Logical characterizations of regular equivalence in weighted social networks[J]. Artificial Intelligence, 2014, 214: 66-88.

[13] TOIVONEN H, ZHOU F, HARTIKAINEN A, et al. Compression of weighted graphs[C]// Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Diego, USA: ACM Press, 2011: 965-973.

[14] 潘秋萍. 基于条件熵的图聚集算法研究[D]. 硕士学位论文，昆明理工大学，2014.