面向距离查询的属性加权图聚集算法

马慧芳[1,2,3],邴 睿[1],赵卫中[4],常 亮[2]
(1. 西北师范大学计算机科学与工程学院,甘肃兰州 730070; 2. 桂林电子科技大学广西可信软件重点实验室,广西桂林 541004;
3. 广西师范大学广西多源信息挖掘与安全重点实验室,广西桂林 541004; 4. 华中师范大学计算机学院,湖北武汉 430079)

摘 要: 图聚集技术是在保留原始图的结构和属性信息的同时,将一个大规模图聚集成简洁的小规模图的技术. 随着图的规模不断增加使得图数据变得难以查询和存储,而基于距离的查询,例如最短路径查询,非常依赖图的规模大小. 本文提出了面向距离查询的属性加权图聚集算法,在保证节点之间结构和属性相似的同时,保护了节点之间的距离,并有效地减小了图规模. 实验证明本文方法的有效性与在查询任务上的高效性.

关键词: 图聚集; 图查询; 距离保护; 结构相似度; 属性熵
中图分类号: TP301.6 **文献标识码:** A **文章编号:** 0372-2112 (2021)01-0132-08
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.12263/DZXB.20190129

Distance-Query-Oriented Attribute Weighted Graph Aggregation Algorithm

MA Hui-fang[1,2,3], BING Rui[1], ZHAO Wei-zhong[4], CHANG Liang[2]
(1. *College of Computer Science and Engineering, Northwest Normal University, Lanzhou, Gansu* 730070, *China*;
2. *Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin, Guangxi* 541004, *China*;
3. *Guangxi Key Laboratory of Multi-source Information Mining & Security, Guangxi Normal University, Guilin, Guangxi* 541004, *China*;
4. *School of Computer, Central China Normal University, Wuhan, Hubei* 430079, *China*)

Abstract: Graph aggregation is a technology that aggregates a large-scale graph into a compact and small-scale graph while retaining the structure and attribute information of the original graph. With the increasing size of graph, graph data becomes difficult to query and store. Distance-based queries, such as shortest path queries, depend heavily on the size of graph. In this paper, a distance query-oriented attribute weighted graph aggregation algorithm is proposed, which not only guarantees the similarity of structure and attributes between nodes, but also preserves the distance between nodes, and effectively reduces the size of the graph. The experiments prove that this method is effective and efficient in the query tasks.

Key words: graph aggregation; graph query; distance preserving; structure similarity; attribute entropy

1 引言

现实生活中,存在大量用图结构表示的数据,图中节点表示实体,边表示实体与实体之间的关系,如社交网络中用户之间的关系,交通网络中道路之间的关系,WEB 图中网页之间的关系等. 由于图数据的规模不断增大,无法直接通过肉眼视觉来处理和分析这些图数据,为了节省存储空间和便于对图数据进行分析,需要将大规模图进行压缩. 因此,图聚集技术成为了研究热点. 与图聚类技术[1]不同的是,图聚集是以相似度为标准进行图压缩,而图聚类则是根据图中节点密度,对节点进行聚类.

图聚集方法可以分为两类,以压缩为目的的图聚集方法和面向查询的图聚集方法.

(1)以压缩为目的的图聚集方法可归纳为以下四类[2]:①基于属性的聚集算法,与 OLAP 技术结合,形成 graph OLAP 方法[3];②基于结构的聚集算法,分为基于概率的图聚集方法[4]与基于最大化压缩的图聚集方法[5];③基于结构与属性的聚集算法,分为基于熵模型的图聚集方法[6,7],SNAP/k-SNAP 聚集算法[8,9];④基于加权图的聚集

收稿日期: 2019-01-22;修回日期:2020-03-12;责任编辑:王天慧
基金项目:国家自然科学基金(No. 61762078, No. 61363058, No. 61762079);广西多源信息挖掘与安全重点实验室开放基金(No. MIMS18-08);广西可信软件重点实验室研究课题(No. kx202003)

算法[10,11].以压缩为目的的图聚集方法在执行查询任务时,需要将聚集图解压,且只适用于单一的查询任务.

(2)面向查询的图聚集方法,目标就是要适用于多种查询任务,如邻域查询[12]、可达性查询[13]、距离查询[14]等. Shrink 方法提出了节点间距离保护的图聚集方法[15],用于距离查询. 但 Shrink 中节点都没有携带属性信息,同时也没有考虑节点邻域相似的情况.

提出了面向距离查询的属性加权图聚集算法(Distance-Query-Oriented Attribute Weighted Graph Aggregation Algorithm),可用于查询和存储加权图和无权图,且使用此算法进行聚集对节点间的距离影响最小.

2　基础知识

2.1　属性加权图与聚集图

定义 1　(属性加权图)　给定一个无向属性加权图 G 为四元组 $G=(V,E,A,W)$,其中 $V=\{v_1,v_2,\cdots,v_n\}$ 是图中节点的集合,$E\subseteq\{(v_i,v_j)\mid v_i,v_j\in V\}$ 是节点之间边集,(v_i,v_j) 表示图中节点 v_i,v_j 之间的边;有穷集合 $A=\{a_1,a_2,\cdots,a_m\}$ 是节点属性的集合;$W=\{w(v_1,v_1),\cdots,w(v_i,v_j),\cdots,w(v_n,v_n)\}$ 表示边权重集合,如果边 $(v_i,v_j)\notin E$,则 $w(v_i,v_j)=0$.

定义 2　(聚集图)　无向属性加权图 $S=(V_s,E_s,A_s,W_s)$ 是图 G 的聚集图,其中 $V_s=\{v_{s1},v_{s2},\cdots,v_{sk}\}$ 是节点集合 V 的划分,满足以下条件:

$$v_{si}\subset V;\quad \bigcup_{i=1}^{k}v_{si}=V;\quad v_{si}\cap v_{sj}=\varnothing,\quad i\neq j$$

节点 $v_{si}\in V_s$ 称为超级节点(简称超点);$E_s=\{(v_{si},v_{sj})\mid \exists v_i\in v_{si},v_j\in v_{sj},(v_i,v_j)\in E\}$ 为超边集合,其中 (v_{si},v_{sj}) 称为超点 v_{si},v_{sj} 之间的超边;$A_s=\{a_1,a_2,\cdots,a_m\}$ 表示超点的属性集合,维度与原始图中属性维度相同;$W_s=\{w_s(v_{s1},v_{s1}),\cdots,w_s(v_{si},v_{sj}),\cdots,w_s(v_{sk},v_{sk})\}$ 表示超边权重值集合,其中 $w_s(v_{si},v_{sj})$ 表示超点 v_{si},v_{sj} 之间超边的权重值.

2.2　结构相似度

为了衡量节点对在结构上是否相似,接下来给出结构相似度的定义[16].

定义 3　(结构相似度)　节点对 v_x,v_y 之间的结构相似度记为 $\sigma(v_x,v_y)$,定义为节点 v_x 的结构闭邻域 $N[v_x]$ 和节点 v_y 的结构闭邻域 $N[v_y]$ 中共同节点的数量与 v_x,v_y 度数的几何平均数的比值,即:

$$\sigma(v_x,v_y)=\frac{|N[v_x]\cap N[v_y]|}{\sqrt{d[v_x]\cdot d[v_y]}}\qquad(1)$$

其中,节点 v_x,v_y 的度数分别为 $N[v_x]$ 和 $N[v_y]$ 的基数,即 $d[v_x]=|N[v_x]|$,$d[v_y]=|N[v_y]|$,即节点结构闭邻域的度数为节点的边数加 1. 对于给定两个节点,结构闭邻域中共同邻居节点越多,结构相似度越大. 结构相似度 σ 的取值范围为[0,1]. 设结构相似度阈值为 θ,本文算法将对结构相似度超过阈值 θ 的节点对进行合并.

以图 1 中的节点 v_1,v_4 为例,节点 v_1 的结构闭邻域为 $N[v_1]=\{v_1,v_2,v_3,v_4\}$,节点 v_4 的结构闭邻域为 $N[v_4]=\{v_1,v_2,v_3,v_4,v_5\}$,设结构相似度阈值为 $\theta=0.8$,计算得到 $\sigma(v_1,v_4)\approx0.89$,超过了阈值 θ,所以认为节点对 v_1,v_4 在结构上是相似的.

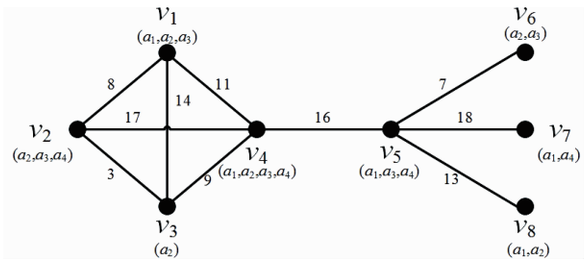


图1　属性加权图

使用引理 1 中的剪枝策略[16],判断出结构不相似的节点对,减少了相似度计算量.

引理 1　(剪枝策略)　令结构相似度阈值为 θ,对于节点 v_x,v_y,如果 $d[v_x]<\theta^2\cdot d[v_y]$ 或者 $d[v_y]<\theta^2\cdot d[v_x]$,那么 $\sigma(v_x,v_y)<\theta$.

2.3　属性相似度

为了衡量节点之间的属性相似程度,使用属性熵来衡量节点之间的属性相似程度[7]. 定义如下:

定义 4　(属性熵)　节点 v_x,v_y 对之间的属性熵定义为合并 v_x,v_y 之后形成的超点内部的原始节点之间的属性一致性熵值,记为:

$$\text{Entropy}(v_x,v_y)=\sum_{j=1}^{m}\text{H}(p(a_j^{v_i}=1))\qquad(2)$$

其中 $a_j^{v_i}$ 表示形成的超点中的原始节点 v_i 在第 j 维属性上的二元取值,如果节点 v_i 具有属性 j,那么属性 $a_j^{v_i}$ 的取值为 1,否则为 0. m 表示属性的总维度. 节点间的属性一致性熵值越小,属性越相似.

以图 1 为例,节点之间的属性关系如表 1 与表 2 所示,对 v_1,v_4 形成的超点内部属性一致性熵值更小,节点对 v_1,v_4 的属性更相似.

表 1　节点属性关系

	节点	a_1	a_2	a_3	a_4
v_{s1}	v_1	1	1	1	0
	v_4	1	1	1	1
v_{s2}	v_3	0	1	0	0
	v_5	1	0	1	1

表 2　属性一致性熵值

	Entropy	$p(a_1=1)$	$p(a_2=1)$	$p(a_3=1)$	$p(a_4=1)$
v_{s1}	0.30	1	1	1	0.5
v_{s2}	1.20	0.5	0.5	0.5	0.5

2.4 合并误差与大数定理

本节将介绍合并误差与如何利用大数定理保证在规模越大的图上,合并误差越小.

定义5 (最短路径) 节点v_x,v_y之间的最短路径定义为$d(v_x,v_y)$,表示节点对v_x,v_y之间的路径中边权重最小的路径. $d'(v_x,v_y)$表示包含节点对v_x,v_y的超点之间的最短路径.

定义6 (压缩误差) 节点对v_x,v_y之间由压缩产生的误差为$Err(v_x,v_y)=|d(v_x,v_y)-d'(v_x,v_y)|,x\neq y$,表示节点对$v_x,v_y$在原始图中之间的最短路径与在聚集图中对应超点之间的最短路径的差值.

定义7 (相关压缩误差) 使用$d(v_x,v_y)$对压缩误差$Err(v_x,v_y)$进行归一化,定义为节点对v_x,v_y之间的相关压缩误差,记为:

$$RErr=\frac{|d(v_x,v_y)-d'(v_x,v_y)|}{d(v_x,v_y)},\quad x\neq y \tag{3}$$

定义8 (压缩率) 给定原始图G和聚集图S,压缩率CR定义为:

$$CR(S)=\frac{|V_S|}{|V|} \tag{4}$$

其中,$|V_s|$表示聚集图中的节点个数,$|V|$表示原始图中的节点个数. 压缩率CR的取值范围为$(0,1)$.

定义9 (合并误差) 设节点对v_x,v_y是待合并节点,pa是未知两点之间的最短路径,且pa通过节点v_x或节点v_y,合并节点对v_x,v_y在路径pa上产生的合并误差为:

$$M_{v_xv_y}(pa)=l(pa)-l'(pa) \tag{5}$$

其中$l(pa)$表示合并节点对v_x,v_y路径将会改变的部分,$l'(pa)$表示合并节点对v_x,v_y后路径改变了的部分.

如图2所示,设路径pa通过了节点v_i,v_x,v_y,v_j,那么$l(pa)=w(v_i,v_x)+w(v_x,v_y)+w(v_y,v_j)$,在聚集图中,$l'(pa)=w_s(v_i,v_{sz})+w_s(v_{sz},v_y)$,因此在路径$pa$上合并节点对$v_x,v_y$所产生的合并误差为$[w(v_i,v_x)+w(v_x,v_y)+w(v_y,v_j)]-[w_s(v_i,v_{sz})+w_s(v_{sz},v_y)]$. 值得注意的是,合并误差的值可以是正值可以是负值,取决于节点v_i,v_j和权重$w_s(v_i,v_{sz}),w_s(v_{sz},v_y)$.

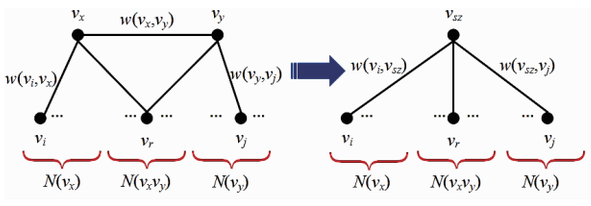


图2 节点合并示例

在路径pa上,每一次合并节点对产生的合并误差$M_{v_xv_y}(pa)$都可以看作为一个随机变量. 每次合并的目标是保持每个节点对的合并误差的均值为零.

在概率论中,相互独立且服从同一分布的若干随机变量,这些随机变量的和的期望等于它们期望之和. 因此,可以得到:

$$\mathrm{E}(M(pa))=\mathrm{E}(M_{v_{x1}v_{y1}}(pa))+\cdots+\mathrm{E}(M_{v_{xm}v_{ym}}(pa)) \tag{6}$$

pa是任意两个结点之间的最短路径,$M(pa)$是路径pa上合并节点对产生的总误差.

设节点对v_x,v_y为待合并节点对,v_x,v_y的邻居可分为三个集合:$N(v_x),N(v_y),N(v_xv_y)$,其中$N(v_x)$是v_x的邻居集合,$N(v_y)$是v_y的邻居集合,$N(v_xv_y)$是v_x,v_y的共同邻居集合. 并且$N=N(v_x)\cup N(v_y)\cup N(v_xv_y)$,$|N|=k$.

利用k个变量定义了k个线性方程组来寻找新的权重,其中k是v_x,v_y的邻居的总数. 在原始图上通过$v_i\in N$和v_x或v_y路径长度的和值应等于在压缩图上通过v_x和v_{si}的路径长度的和值,即:

$$\sum_{v_j\in N,j\neq i}l'(v_i,v_j)=\sum_{v_j\in N,j\neq i}l(v_i,v_j) \tag{7}$$

其中,v_i,v_j是节点对v_x,v_y的邻居节点. 让合并前与合并后的路径长度相等,就能使合并误差$M_{v_xv_y}(v_i)$成为零均值随机变量.

根据大数定理,数量足够大的独立且服从同一分布的零均值随机变量收敛到零. 即:

$$\lim_{m\to\infty}\frac{Mv_{x_1}v_{y_1}(pa)+\cdots+Mv_{x_m}v_{y_m}(pa)}{m}=\lim_{m\to\infty}\frac{M(pa)}{m}=0 \tag{8}$$

因此,在路径pa上产生的零均值的合并误差越多,路径pa上的总合并误差越接近于零. 在规模越大的图上,压缩效果越好.

3 面向距离查询的属性加权图聚集算法

本文提出了面向距离查询的属性加权图聚集算法,在保证了节点间结构和属性相似度的情况下,保护了结点间距离,且引入方程组使误差最小化,即使误差的平均值等于零.

3.1 质量分数

接下来给出节点对之间质量分数的定义.

定义10 (质量分数) 结合了节点对v_x,v_y之间的结构相似度与属性熵,v_x,v_y之间的质量分数定义为:

$$Q(v_x,v_y)=\frac{1}{2}\sigma(v_x,v_y)+\left(\frac{m}{\mathrm{Entropy}(v_x,v_y)+m}-\frac{1}{2}\right) \tag{9}$$

质量分数Q用来衡量节点对之间的属性与结构相似度且质量分数的取值范围设定在$[0,1]$.

3.2 超边权重计算

与Shrink方法不同,本文可对之间没有边的节点对进行合并,并计算合并后超边之间的权重值.

图3,图4与图5给出了节点v_i,v_j在三种不同的情况下,路径$l'(v_i,v_j)$与$l(v_i,v_j)$的计算过程. 在图3,图4与图5中,加粗的边表示通过的路径.

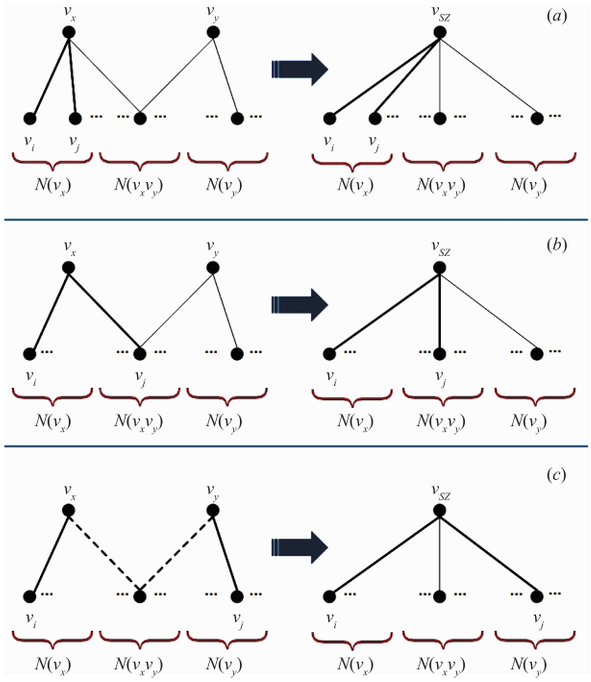


图3　路径计算过程

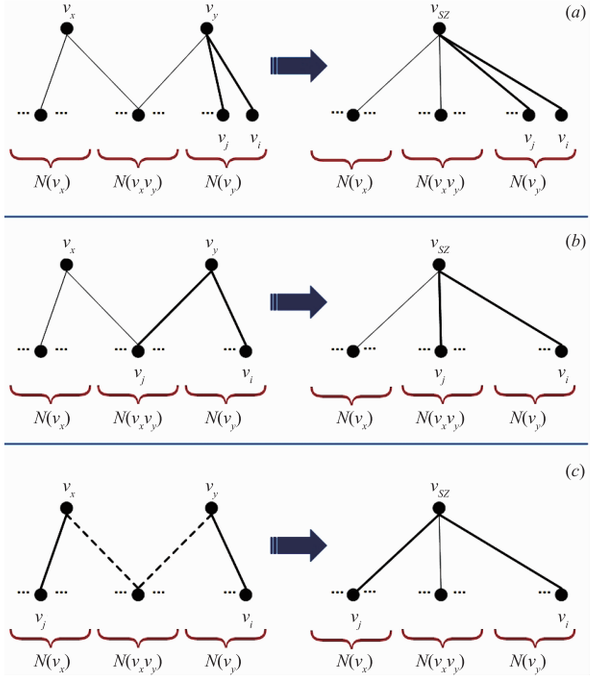


图4　路径计算过程

3.2.1　当$v_i \in N(v_x)$时

如图3所示,当$v_i \in N(v_x)$时,根据式(7),使原始图中合并v_x,v_y将会改变的路径长度$l(v_i,v_j)$的总和值等于合并后改变了的路径长度$l'(v_i,v_j)$的总和值,即:

$$(k-2)w_S(v_{SZ},v_i)+\sum_{v_j\in N}w_S(v_{SZ},v_j)=C_i \quad (10)$$

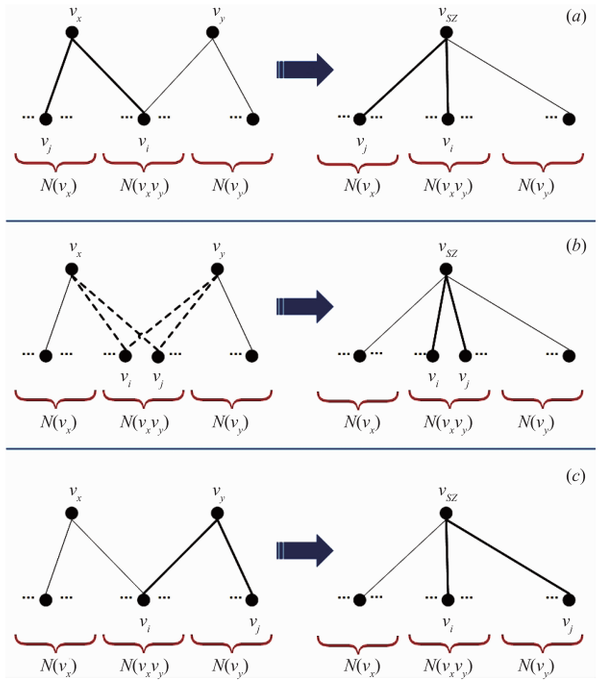


图5　路径计算过程

当$v_i \in N(v_x)$时,用C_i来表示原始图中合并点对v_x,v_y将会改变的路径长度$l(v_i,v_j)$的总和值,C_i可以看作为一个已知量.

3.2.2　当$v_i \in N(v_y)$时

如图4所示,当$v_i \in N(v_y)$时,合并后的路径长度$l'(v_i,v_j)$的方程定义与$v_i \in N(v_x)$时相似,两种情况是互为对称的关系,整理得到:

$$(k-2)w_S(v_{SZ},v_i)+\sum_{v_j\in N}w_S(v_{SZ},v_j)=C_i \quad (11)$$

3.2.3　当$v_i \in N(v_xv_y)$时

如图5所示,当$v_i \in N(v_xv_y)$时,根据式(7),整理得到:

$$(k-2)w_S(v_{SZ},v_i)+\sum_{v_j\in N}w_S(v_{SZ},v_j)=C_i \quad (12)$$

3.3　方程求解

定义方程组来计算新的权重,首先定义s为新形成的边的权重总和值,即:

$$s=\sum_{v_j\in N}w_S(v_{SZ},v_j) \quad (13)$$

使用s来代替式(10)、式(11)和式(12)中的新权重值总和,由于节点对v_xv_y共有k个邻居节点,于是就得到了如下所示的规模为k的线性方程组.

$$\begin{cases}(k-2)w_S(v_{SZ},v_1)+s=C_1\\ \vdots\\ (k-2)w_S(v_{SZ},v_i)+s=C_i\\ \vdots\\ (k-2)w_S(v_{SZ},v_k)+s=C_k\end{cases} \quad (14)$$

将这 k 项等式相加,整理得到:

$$(k-2)\left\{\begin{array}{l}\sum\limits_{v_i\in N(v_x)} w_S(v_{SZ},v_i)+\\ \sum\limits_{v_i\in N(v_x,v_y)} w_S(v_{SZ},v_i)+\\ \sum\limits_{v_i\in N(v_y)} w_S(v_{SZ},v_i)\end{array}\right\}+ks=C \tag{15}$$

式(15)可化简为:

$$(k-2)s+ks=C \tag{16}$$

整理得到:

$$s=\frac{C}{2k-2} \tag{17}$$

其中 C 是式(14)中常数 C_i 的和值. 确定了新形成的边的权重和值 s 后,就可以计算每一条新形成的超边的权重值.

$$w_S(v_{SZ},v_i)=\frac{C_i-s}{k-2} \tag{18}$$

3.4 特殊情况

当合并节点对的邻居节点的数量 $k\leq 2$ 的时候,合并节点对前后路径的计算方法与之前讨论的三种情况有所不同.

当 $k=2$ 时,待合并节点对 v_x,v_y 只有两个邻居节点 v_i,v_j. 若节点 v_i,v_j 是节点 v_x,v_y 的共同邻居节点时,可以得到:

$$C_i=\min\left\{\begin{array}{l}w(v_i,v_x)+w(v_x,v_j),\\ w(v_i,v_y)+w(v_y,v_j)\end{array}\right\}=s \tag{19}$$

其中,s 为新形成的边的权重总和值,C_i 是会改变的路径长度 $l(v_i,v_j)$ 的总和值. 此时,两条超边 $w_s(v_{sz},v_j)$,$w_s(v_i,v_{sz})$ 的权重值设定为 $C_i/2$,即 $w_s(v_{sz},v_j)=w_s(v_i,v_{sz})=C_i/2$;若节点 v_i,v_j 是节点 v_x 的邻居节点时,$C_i=w(v_i,v_x)+w(v_x,v_j)=s$,所以 $w_s(v_{sz},v_j)=w_s(v_i,v_{sz})=C_i/2$;若节点 v_i,v_j 节点 v_y 的邻居节点时,$C_i=w(v_i,v_y)+w(v_y,v_j)=s$,所以 $w_s(v_{sz},v_j)=w_s(v_i,v_{sz})=C_i/2$.

当 $k=1$ 时,待合并节点对 v_x,v_y 只有一个邻居节点 v_i,即只有一条边 (v_i,v_x) 或是 (v_i,v_y) 连向节点 v_x 或 v_y,合并后只会形成新的超边 $w_s(v_i,v_{sz})$,所以当 (v_i,v_x) 存在时,$w_s(v_i,v_{sz})=w(v_i,v_x)$,当 (v_i,v_y) 存在时,$w_s(v_i,v_{sz})=w(v_i,v_y)$. 解方程的时间复杂度为 $O(k)$,定义方程所需的时间复杂度为 $O(k^2)$,因为在式(14)中的定值需要 $O(k)$ 来确定. 在大规模图中,合并误差远远小于节点的数量[15],这时,总时间复杂度为 $O(|V|)$.

4 实验与性能分析

为了验证本文方法的有效性,将设计对比实验,从准确性、图规模、压缩率等几个方面进行比对,并给出实验结果分析.

4.1 实验数据分析

纽约市道路网络图(NY road network)数据集:该网络图为无向图,图中节点代表十字路口或是道路端点,边上的权重代表此道路的长度.

DBLP 作者合著关系图数据集:图中节点表示作者,边表示合著关系,边上的权重表示合作的次数. 节点属性是来自 5 个领域的 21 个会议. 若作者在某会议上发表过论文,则对应的属性值取 1,否则属性值取 0. 实验数据总结如表 3 所示.

表 3 实验数据集

数据集	节点	边
NY road network	264,346	733,846
DBLP	137,524	287,503

4.2 评价指标

本文选取压缩率、相关压缩误差、运行时间、聚集图规模、查询时间作为实验的评价指标.

4.3 实验结果与分析

本文选取 Shrink 方法[15] 与加权图聚集方法(CWG)[10] 作为对比算法,来评估本文算法的有效性. Shrink 与 CWG 的输入都是没有节点属性信息的加权图. 为了便于算法之间的比较,本文算法在与对比算法进行对比实验时,不再计算属性熵值和质量分数. 本文将在带有节点属性信息的 DBLP 数据集上执行完整的面向距离查询的属性加权图聚集算法. 在实验中使用“查询算法”来表示本文算法.

4.3.1 参数选择

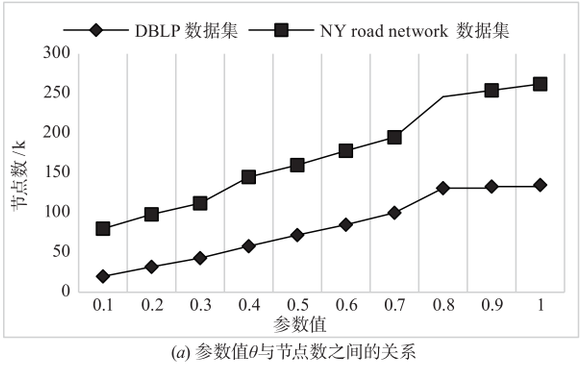
参数 θ 为节点对之间的结构相似度阈值,控制着图聚集过程中可合并的节点对的数量. 本文分别对比参数 θ 与聚集图规模和压缩时间之间的关系,以选取合适的参数 θ. 实验结果如图 6 所示.

在选取合适的参数 θ 之前,先将参数 θ 的初始值设置为 0,以 0.1 的步长递增,直到 θ 值为 1 时停止. 图 6(a)展示了参数 θ 与聚集图规模之间的关系. 参数 θ 的值越小,可以合并的节点对就越多. 当参数 θ 的值增长到一定程度后,没有可合并的节点对,聚集图规模趋于平稳. 图 6(b)展示了参数 θ 与压缩运行时间之间的关系,随着参数 θ 的值增大,压缩所需的运行时间越来越短,由于参数 θ 的值变大,可合并节点对减少,最后趋于平稳. 根据实验结果综合考量,选取参数 θ 值为 0.8.

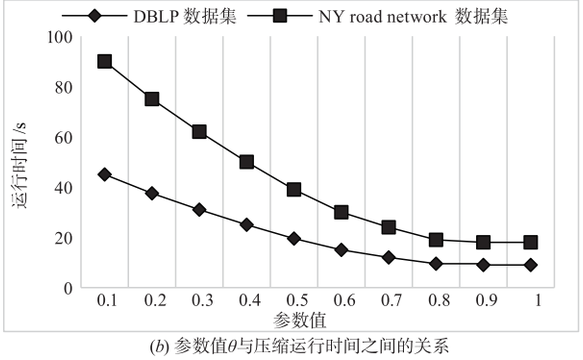
4.3.2 运行时间比较

选取 NY road network 数据集与 DBLP 数据集,来比较三种算法的压缩时间. 实验结果如图 7 所示.

随着节点数量的不断增加,算法运行时间越来越长. 查询算法稍慢于 Shrink 算法,因为查询算法需计算节点之间的结构相似度. 与此同时,Shrink 算法与查询算法的运行时间较为线性,而 CWG 算法的运行时间增长较快.

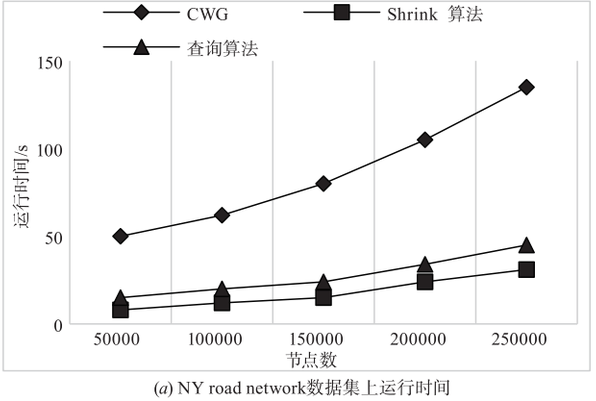


(*a*) 参数值θ与节点数之间的关系

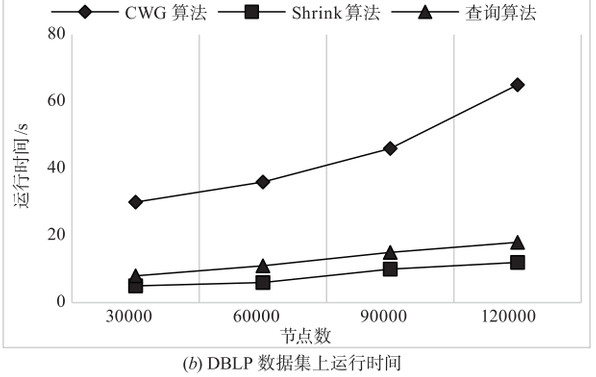


(*b*) 参数值θ与压缩运行时间之间的关系

图6　参数选择



(*a*) NY road network数据集上运行时间

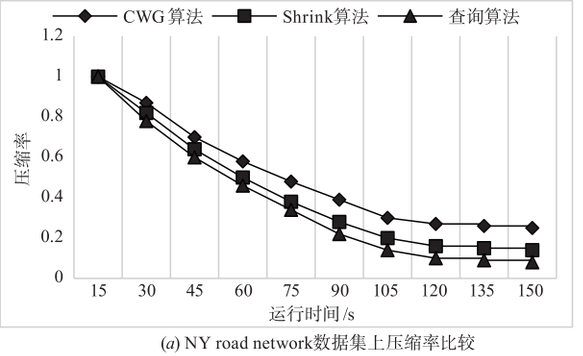


(*b*) DBLP 数据集上运行时间

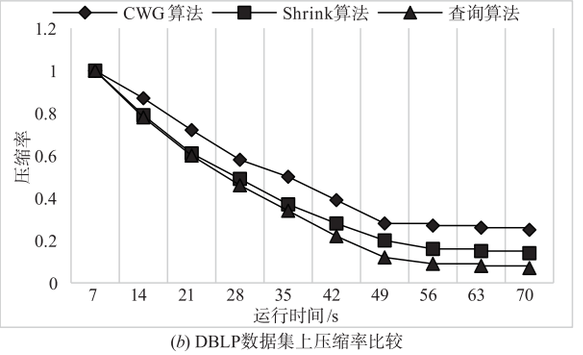
图7　运行时间比较

4.3.3　压缩率比较

在 NY road network 数据集和 DBLP 数据集来比较三种算法在运行时间和压缩率上的关系. 实验结果如图 8 所示. 查询算法的压缩率最低,是因为查询算法可以选取之间不存在边的节点对进行合并,所以可合并的节点对数量要大于 Shrink 算法和 CWG 算法. 当没有满足合并条件的节点对,压缩率就会趋于稳定.



(*a*) NY road network数据集上压缩率比较



(*b*) DBLP数据集上压缩率比较

图8　压缩率比较

4.3.4　相关压缩误差比较

本文通过相关压缩误差与节点数之间的关系来衡量三种算法的聚集质量. 实验结果如图 9 所示.

从图 9 中可以看到,Shrink 算法与查询算法的相关压缩误差远小于 CWG 算法的相关压缩误差,因为 CWG 算法没有保持在合并前后节点对之间的距离. 随着节点数的增加,相关压缩误差的始终在一个稳定的范围内.

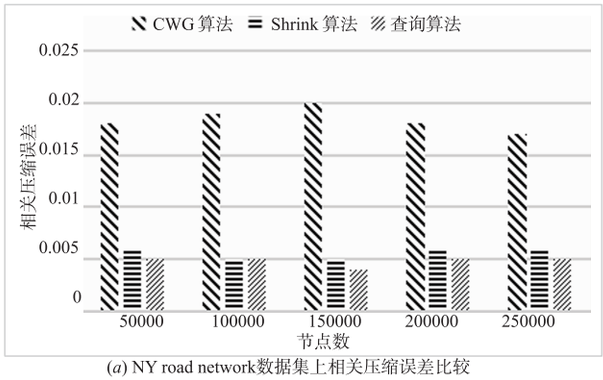
4.3.5　查询时间比较

本文设置实验来比较三种算法的图查询时间,实验结果如图 10 所示.

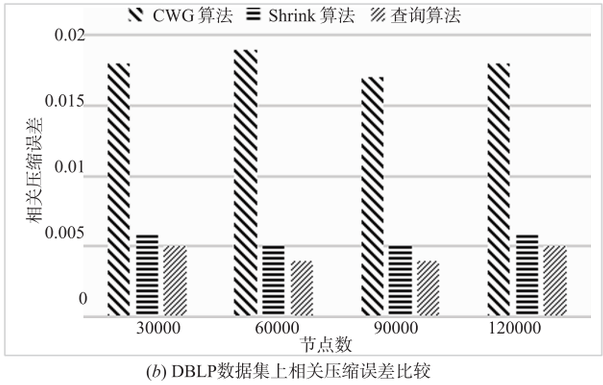
由图可知,CWG 算法执行查询任务耗时最多,因为 CWG 算法需要将聚集图解压成为原始图执行查询任务. 由于查询算法得到的聚集图规模小于 Shrink 算法,执行查询任务时所遍历的超点数较少,所以查询算法的查询时间略快于 Shrink 算法.

4.3.6　带有节点属性的算法比较

本节对比了完整的查询算法和无节点属性信息的查询算法之间的差异. 实验结果如图 11 所示.

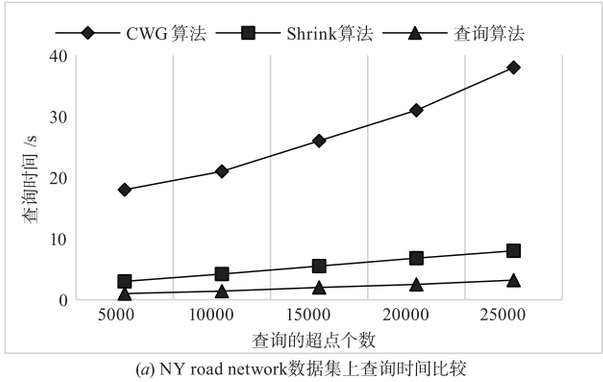


(*a*) NY road network数据集上相关压缩误差比较

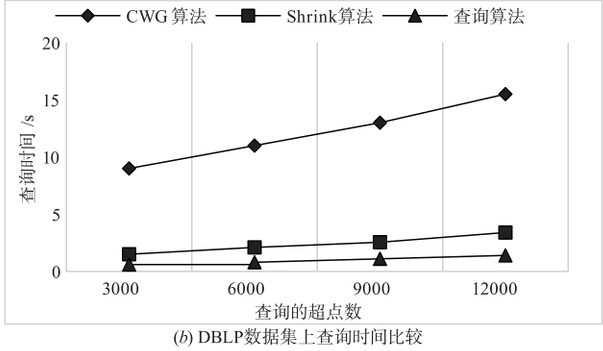


(*b*) DBLP数据集上相关压缩误差比较

图9 相关压缩误差比较



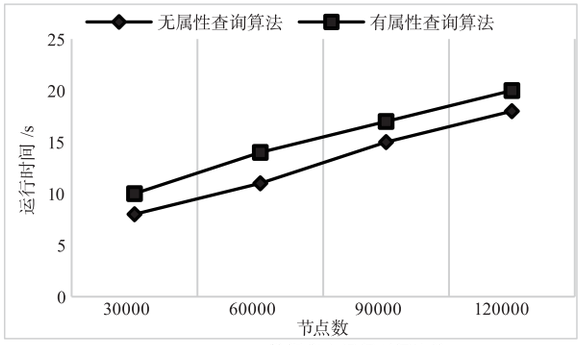
(*a*) NY road network数据集上查询时间比较



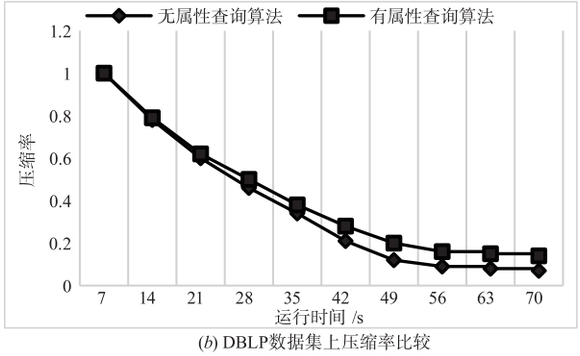
(*b*) DBLP数据集上查询时间比较

图10 查询时间比较

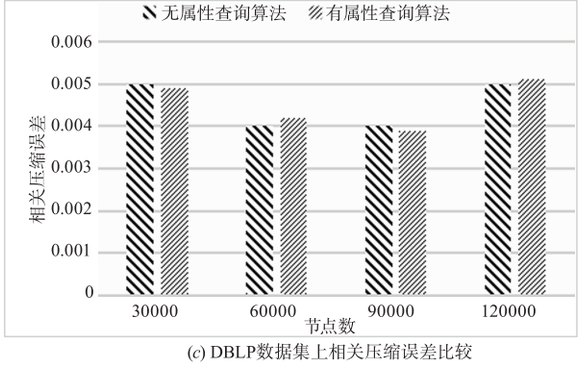
图11(*a*)展示了两种算法在运行时间上的对比结果,完整的查询算法略慢于无属性的查询算法,是因为



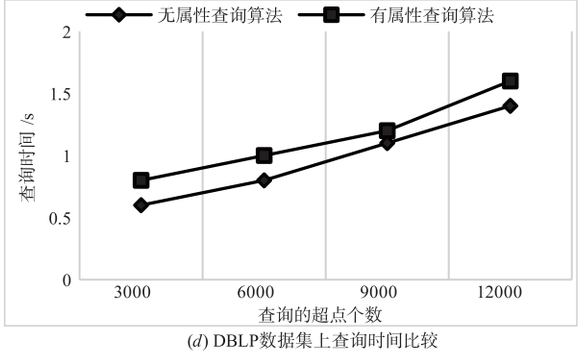
(*a*) DBLP数据集上运行时间比较



(*b*) DBLP数据集上压缩率比较



(*c*) DBLP数据集上相关压缩误差比较



(*d*) DBLP数据集上查询时间比较

图11 带有节点属性的算法比较

完整的查询算法需要计算节点之间的属性一致性熵,而图的属性个数一般远远小于节点的数量,所以熵值的计算对算法的运行时间影响很小. 图11(*b*)展示了两种算法在压缩率上的对比结果,由于计算属性一致性熵值,完整的查询算法在单位时间内的压缩速度略慢

于无属性的查询算法. 图 11(*c*)展示了算法在相关压缩误差上的对比结果,因为赋予超边新的权重的策略相同,所以相关压缩误差的性能表现基本相同. 图 11(*d*)展示了两种算法在查询时间上的实验结果,可以看到完整的查询算法略慢于无属性的查询算法,因为产生的聚集图规模会略大于无属性查询算法.

5　结束语

本文提出了面向距离查询的属性加权图聚集算法,可有效聚集加权图,并且在压缩后对节点间的距离影响最小,可在聚集图上执行距离查询任务. 同时保证了节点之间的结构与属性相似度,最大程度的保留了原始图中的信息. 该算法是一种通用的图聚集算法,适用于不同的图类型的查询任务,使得此方法能广泛应用于实际应用中.

参考文献

[1] 张建朋,等. 基于采样的大规模图聚类分析算法[J]. 电子学报,2019,47(8):1731 – 1737.
ZHANG Jian-Peng, et al. A sampling-based graph clustering algorithm for large-scale networks[J]. Acta Electronica Sinica,2019,47(8):1731 – 1737. (in Chinese)

[2] 潘秋萍,游进国,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, et al. Progress and challenges of graph aggregation and summarization techniques[J]. Journal of Software,2015,26(1):167 – 177. (in Chinese)

[3] Chen C, et al. Graph OLAP: towards online analytical processing on graphs [A]. Proceedings of the International Conference on Data Mining [C]. Pisa, Italy: IEEE, 2008. 103 – 112.

[4] Le Fevre K, Terzi E. GraSS: Graph structure summarization [A]. Proceedings of the International Conference on Data Mining [C]. Sydney, Australia: IEEE, 2010. 454 – 465.

[5] Navlakha S, Rastogi R, Shrivastava N. Graph summarization with bounded error [A]. Proceedings of the International Conference on Management of Data [C]. Vancouver, Canada: ACM, 2008. 419 – 432.

[6] 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. (in Chinese)

[7] Liu Z, Yu J X, Cheng H. Approximate homogeneous graph summarization [J]. Journal of Information Processing, 2011,20(1):77 – 88.

[8] Tian Y Y, Hankins R A, Patel J M. Efficient aggregation for graph summarization [A]. Proceedings of the International Conference on Management of Data [C]. Vancouver, BC, Canada: ACM, 2008. 419 – 432.

[9] Zhang N, Tian Y Y, Patel J M. Discovery-driven graph summarization[A]. Proceedings of the International Conference on Data Engineering [C]. Long Beach, USA: IEEE, 2010. 880 – 891.

[10] Toivonen H, Zhou F, Hartikainen A, et al. Compression of weighted graphs [A]. Proceedings of the International Conference on Knowledge Discovery and Data Mining [C]. San Diego, USA: ACM, 2011. 965 – 973.

[11] 胡宝丽,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.
Hu BaoLi, et al. An efficient weighted graph aggregation algorithm[J]. Journal of University of Science and Technology of China,2016,46(3):180 – 187. (in Chinese)

[12] Maserrat H, Pei J. Neighbor query friendly compression of social networks [A]. Proceedings of the International Conference on Knowledge Discovery and Data Mining [C]. Washington, DC, USA: ACM, 2010. 533 – 542.

[13] Van Schaik S J, De Moor O. A memory efficient reachability data structure through bit vector compression[A]. Proceedings of the International Conference on Management of Data[C]. Athens, Greece: ACM, 2011. 913 – 924.

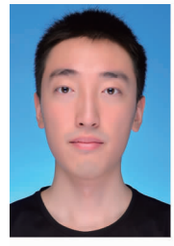
[14] Ruan N, Jin R, Huang Y, Distance preserving graph simplification [A]. Proceedings of the International Conference on Data Mining [C]. Vancouver, Canada: IEEE, 2011. 1200 – 1205.

[15] Sadri A, et al. Shrink: Distance preservinggraph compression[J]. Information Systems, 2017, 69: 180 – 193.

[16] Chang L, Li W, Qin L, et al. PSCAN: Fast and exact structural graph clustering[J]. IEEE Transactions on Knowledge and Data Engineering, 2017, 29(2): 387 – 401.

作者简介

马慧芳　女,1981 年 7 月出生,甘肃兰州人. 博士,硕士生导师,现为西北师范大学计算机科学与工程学院教授. 研究领域为数据挖掘与机器学习.
E-mail: mahuifang@ yeah. net



邢　睿　男,1994 年 10 月出生,甘肃兰州人. 现为西北师范大学计算机科学与工程学院硕士. 研究方向为机器学习.
E-mail: bingrui1030@ qq. com