

社交网络介数中心度快速更新算法

杨建祥 王朝坤 白易元 郝竞超 汪 浩

(清华大学软件学院 北京 100084)

(清华信息科学与技术国家实验室(筹) 北京 100084)

(信息系统安全教育国家重点实验室(清华大学) 北京 100084)

(rightyang@126.com)

A Fast Algorithm for Updating Betweenness Centrality in Social Networks

Yang Jianxiang, Wang Chaokun, Bai Yiyuan, Hao Jingchao, and Wang Hao

(School of Software, Tsinghua University, Beijing 100084)

(Tsinghua National Laboratory for Information Science and Technology, Beijing 100084)

(Key Laboratory for Information System Security (Tsinghua University), Ministry of Education, Beijing 100084)

Abstract As the rapid development of social network, the evaluation of members' importance has received more and more concentration. The Betweenness Centrality is a kind of widely used metrics of members' importance evaluation. But it is hardly to accept the time consuming of re-calculating the betweenness centrality of whole graph while facing with the rapid changed graph. In order to solve this problem, we changed the Brandes' algorithm for restoring the shortest path distance table between nodes. And we also reduced the node pair during pruning procedure. The algorithm of this paper showed a very good performance on the sparse unweighted directed graph or the graph with many distinct regions through the experiment on different kinds of real data set.

Key words betweenness centrality; fast updating; social network; member importance; unweighted directed graph

摘 要 近年来,社交网络的高速发展,社交成员的重要性度量受到了越来越多的关注.其中,介数中心度(betweenness centrality)被广泛地应用于节点重要性的度量中.但是在图结构频繁变化的场合,若每次都需要重新计算整个图的介数中心度,效率将很低.针对这一问题,通过在第1次计算介数中心度时改进 Brandes 算法,并在之后的更新操作时缩小需要计算的最短路径点对范围,提高寻找最短路径效率的方法,大大提高了有向无权图中加边操作时的介数中心度更新效率.该算法在不同真实数据上的实验,尤其是在稀疏图或具有较多独立区域的有向图中表现出很好的更新加速性能.

关键词 介数中心度;快速更新;社交网络;成员重要性;有向无权图

中图法分类号 TP311.13

社交网络中成员节点的重要性度量具有重要的理论价值和现实意义.在社交网络中,存在一些具有重要作用的成员节点.他们可能具有更大的影响

力,或者更强的消息传播能力.社交网络中大量活动将围绕这些重要成员节点展开,或与其具有密切关系^[1].

收稿日期:2012-07-06

基金项目:国家自然科学基金项目(61170064,60803016);“核高基”国家科技重大专项基金项目(2010ZX010402-002-002-01);清华信息科学与技术国家实验室(筹)学科交叉基金项目

研究者们已提出社交网络中诸多节点重要性度量指标,如 Closeness, Degree, Random walk betweenness, Stress, Betweenness^[2]和基于网络流的 CF^[3]等. 作为社交网络节点重要性度量的典型代表,近些年,介数中心度(betweenness centrality)在网络拓扑结构重要节点度量、性关系网络研究、重要恐怖分子发现等方面^[4]得到了广泛应用. 某点的介数中心度越大,代表着该点对图中其他点对最短路径的控制能力越强,是很重要的描述重要性度量方法.

有向图上的中间度介数是由 Anthonisse 等人 1971 年在文献[5]中首先提出, Freeman 等人 1977 年在文献[6]中将这一概念引入具有不连通节点的网络中,现在使用最广泛的是 Freeman 等人所提出的定义.

由于介数中心度的定义本身是基于图中所有点对间最短路径的计算,所以其初始算法具有 $O(n^3)$ 时间复杂度,这使得介数中心度的运算无法在较大的数据集上进行^[7]. 当前最快的精确计算介数中心度的算法是 Brandes 在文献[8]中提出的算法^[4,9],这一算法需要 $O(n+m)$ 空间,在无权有向图中需要 $O(nm)$ 时间复杂度,在有权有向图中需要 $O(nm + n^2 \log n)$ 时间复杂度,大大提升了介数中心度的运算效率.

但是在很多应用场合,图结构会发生经常性变化,如微博会经常发生好友关注变化,网络连接会发生连接路径改变等,若每次图结构的改变都需要重新计算一次介数中心度的资源消耗是不可接受的. 在文献[9]中第 1 次提出了针对无向图的介数中心度快速更新算法,其利用最小连结环结构(minimum union cycle)减少每次需要计算最短距离的点对,从而提升更新效率.

但是在微博等社交网络应用场合更多地是采用有向图来表达成员节点之间的关系,而非无向图,文献[9]中的算法并不能适用于这些场合. 本文提出的快速更新算法基于有向无权图. 且由于社交网络中更多的是加边操作(如关注好友),很少发生删边操作(如删除好友),所以本文所描述的介数中心度更新算法是针对加边操作考虑的.

本文主要贡献如下:

- 1) 本文首次提出了针对有向无权图的介数中心度快速更新算法.
- 2) 对算法的加速原理和可能出现算法退化的情形进行理论分析.
- 3) 通过对同一数据不同数据规模 and 不同数据的更新测试,验证了之前的理论分析.

1 预备知识

1.1 基本概念

如图 1 所示,社交网络中对象的关系抽象为有向图 $G=(V, E)$, 节点 V 一般用来表示社交网络中的成员,边 E 表示成员节点之间的关系,如微博关注等. 在本文中将只考虑无权图算法,即认为每条边的权值为 1.

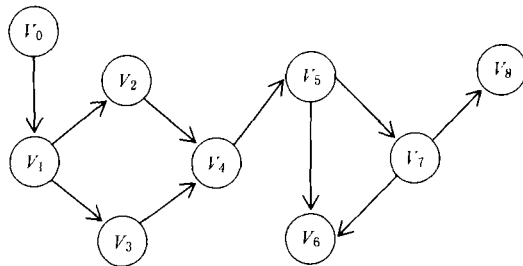


图 1 有向无权图 $G=(V, E)$

本文中算法还使用了反转图 $G_r=(V', E')$, 其点集合与图 G 是相同的,但是边方向相反.

对节点 $\forall s, t \in V, d_G(s, t)$ 表示 s 和 t 之间的最短路径的长度,若 $s=t, d_G(s, t)=0$.

定义 1. 介数中心度 (betweenness centrality).

对图 $G=(V, E), \forall s, t \in V$, 且 $s \neq v \neq t$, 节点 v 的介数中心度为

$$C_B(v) = \sum_{s, t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}, v \in V,$$

其中, σ_{st} 表示在有向图 G 中 s 点到 t 点的最短路径数量, $\sigma_{st}(v)$ 表示在 s 点到 t 点的最短路径通过点 v 的数量. 当 $v \in \{s, t\}$ 时, $\sigma_{st}(v)=0$, 若 s 点到 t 点不连通, 则 $\forall v \in V, \sigma_{st}(v)/\sigma_{st}=0$.

定义 2. 点对依赖.

在 Brandes 在文献[7]中首次定义了点对间依赖(pair-wise dependency)为

$$\delta_{st}(v) = \frac{\sigma_{st}(v)}{\sigma_{st}}.$$

将同一出发点的点对间依赖加起来,引出单点依赖(one-sided dependency)的定义:

$$\delta_s(v) = \sum_{t \in V} \delta_{st}(v).$$

这样就把介数中心度的公式由单点依赖表示为

$$C_B(v) = \sum_{s, v \in V} \delta_s(v), v \in V.$$

定义 3. 更新源节点组和更新目的节点组.

在图 G 中加入边 $s \rightarrow t$, 所有需要更新最短路径的源节点集合称为更新源节点组 S , 所有需要更新

最短路径的目的节点集合称为更新目的节点组 T . 本文将采用 S 和 T 分别表示这两个概念.

1.2 介数更新操作的 5 个定理

假设在图 $G=(V,E)$ 中加入边 $w \rightarrow v$, 观察图特点, 引出如下 5 个定理:

定理 1. 更新源节点组 S 中点必可达 w 节点.

证明. 若 S 中节点不可达 w 节点, 则意味着 s 到 t 的最短路径不会经过 $w \rightarrow v$ 边, 其最短路径也不会因为增加 $w \rightarrow v$ 而改变, 这与更新源节点组定义矛盾. 所以 S 中点必可达 w 节点.

即证 S 中点必可达 w 节点.

定理 2. T 中点必可由 v 节点到达.

同理定理 1 的证明过程.

定理 3. 若 $d_G(s,t) \geq d_G(s,w) + 1 + d_G(v,t)$, $s \in S, t \in T$, 则加入 $w \rightarrow v$ 边后 s 到 t 的最短路径长度为 $d_G(s,w) + 1 + d_G(v,t)$.

证明. 如图 2 所示, 设 s 到 t 之间最短路径为经过点 m 这条路径, 用 $\{s \cdots m \cdots t\}$ 表示, 这条路径长度为 $d_G(s,t)$. 若加边 $w \rightarrow v$, 且此时 $d_G(s,t) > d_G(s,w) + 1 + d_G(v,t)$.

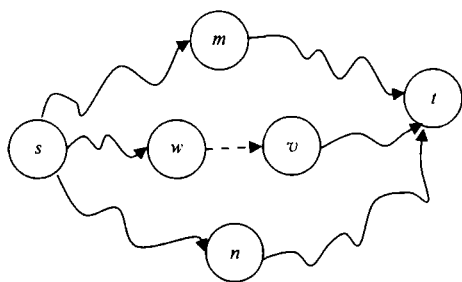


图 2 s 到 t 的最短路径示意图

若加入边 $w \rightarrow v$ 此时最短路径不为 $\{s \cdots w, v \cdots t\}$ 这条路径, 而是其他不经过 $w \rightarrow v$ 边的路径, 如 $\{s \cdots n \cdots t\}$ 这条路径, 则这与 $\{s \cdots m \cdots t\}$ 为最短路径矛盾, 所以此时最短路径定为 $\{s \cdots w, v \cdots t\}$ 这条路径. 而 s 到 w 最短路径长度为 $d_G(s,w)$, v 到 t 最短路径长度为 $d_G(v,t)$, 即此时最短路径长度为 $d_G(s,w) + 1 + d_G(v,t)$.

同理, 若 $d_G(s,t) = d_G(s,w) + 1 + d_G(v,t)$ 时的加边后最短 s 到 t 之间的最短距离仍然为 $d_G(s,w) + 1 + d_G(v,t)$. 即证定理 3.

定理 4. $s \in S, t \in T$, 若要 s 到 t 的最短路径改变, 当且仅当 $d_G(s,t) \geq d_G(s,w) + 1 + d_G(v,t)$ 时成立.

证明. 当 $d_G(s,t) < d_G(s,w) + 1 + d_G(v,t)$, 若最短路径改变, 改变后的最短路径不可能经过 $w \rightarrow v$

边, 这与定理 3 矛盾. 若要 s 到 t 的最短路径改变, 定当满足 $d_G(s,t) \geq d_G(s,w) + 1 + d_G(v,t)$.

若 $d_G(s,t) \geq d_G(s,w) + 1 + d_G(v,t)$, 由定理 3 知当前的最短路径为 $d_G(s,w) + 1 + d_G(v,t)$, 且经过 $w \rightarrow v$ 边, 则定会改变.

即证上述定理的充要性.

定理 5. 如图 2 所示, 若路径 $\{s$ 到 w 最短路径 $\} + \{w$ 到 $v\} + \{v$ 到 t 最短路径 $\}$ 是 s 到 t 的最短路径, 则 s 到 t 且经过 $w \rightarrow v$ 边的最短路径数量为 $\sigma_{st} = \sigma_{sw} \times \sigma_{vt}$.

证明. 由组合乘法定律易知, s 到 t 点的最短路径数量为 $\sigma_{st} = \sigma_{sw} \times \sigma_{vt}$.

2 介数中心度更新

2.1 Brandes 最短路径介数算法改进

本文提出的算法需要对 Brandes 算法进行改进, 增加点对间最短距离表 $d_G(s,t)$ 的记录, 以便后续更新操作使用.

算法 1. Brandes 最短路径介数算法改进.

输入: 有向图 $G=(V,E)$;

输出: 介数中心度 $C_B(v)$, $\forall v \in V$ 和点对间最短距离表 $d_G(s,t)$, $\forall s, t \in V$.

① 初始化变量. 队列 Q , 栈 S 初始化为空.

② 对每个点 $s \in V$, 进行初始化变量过程.

③ 若队列 Q 不为空, $v = Q.front()$, 将点 v 压入栈 S . 将针对 v 进行路径发现或路径数量累积, 并记录点对间距离表 $d_G(s,t)$.

④ 从后置点向前序点运算单点依赖 $\delta_s(v)$, 计算介数中心度 $C_B(v)$.

如图 3 是经过该算法计算后的图原始介数中心度 $C_B(v)$.

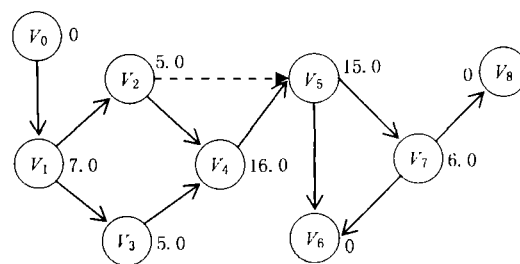


图 3 加边 $V_2 \rightarrow V_5$ 之前图 1 的介数中心度

2.2 更新源节点组和更新目的节点组搜索算法

由本文 1.3 节所述的定理 1, 2 知, 做介数更新

操作首要的是发现更新源节点组 S 和更新目的节点组来缩小最短距离点对的范围. 设将有向图 G 中加入边 $w \rightarrow v$, 寻找点集的伪代码如下:

算法 2. 更新源节点组和更新目的节点组搜索算法.

输入: 有向图 G 和 G_r , 新加入边 $w \rightarrow v$;

输出: 更新源节点组 S 和更新源节点组 T .

① 初始化队列 Q_s, Q_t, w 入队列 Q_s, v 入队列 Q_t, w 点加入到点集 S 中, v 点加入到点集 T .

② 利用有向图 G , 从 w 点开始遍历, 获得点集 S ; 利用有向图 G_r , 从 v 点开始遍历, 获得点集 T .

图 3 对应的更新源节点组为 $\{V_0, V_1, V_2\}$, 更新目的节点组为 $\{V_5, V_6, V_7, V_8\}$.

2.3 最短路径搜索算法

这一步需要通过之前已经保存的最短路径长度表 $d_G(s, t), (s, t \in V)$ 来使用 BFS 方式获得指定点对间的最短路径和最短路径数量. 这是一个递归过程.

伪代码如下:

算法 3. 最短路径搜索算法.

输入: 有向图 $G=(V, E)$ 和点对间最短路径长度表 $d_G(v_i, v_j), (0 \leq i, j \leq |V|)$, 源节点 s , 目的节点 t ;

输出: s 到 t 最短路径上的节点链表 $minPath(s, t)$, 最短路径数目 σ_{st} .

① 初始化 $\sigma_{st} = 0$.

② 从点 s 开始, 使用 BFS 方法遍历图, 若遍历到的边 $(w, v) \in E$ 且 $d_G(w, t) = d_G(v, t) + 1, v \neq t$ 则将 v 点加入到节点链表 $minPath(s, t)$ 中.

③ $\sigma_{st} = \prod_{\text{对任意不相同点 } v \text{ 存在于 } minPath} count(v).$

如图 4 为 V_0 到 V_5 之间的最短路径, 其中包含两条路径为 $\{V_0, V_1, V_2, V_4, V_5\}, \{V_0, V_1, V_3, V_4, V_5\}$, $minPath(V_0, V_5) = \langle V_1, V_2, V_3, V_4, V_4 \rangle$, $\sigma_{V_0 V_5} = 2$.

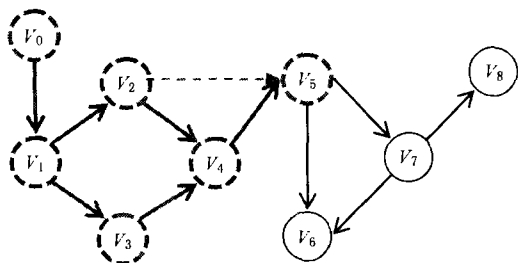


图 4 寻找 V_1 到 V_5 之间最短路径的迭代过程

2.4 加边介数中心度更新算法

这一节将是本文的核心算法, 会用到 2.2 节和 2.3 节所讲的 2 个算法.

算法 4. 加边更新介数中心度算法.

输入: 图 $G=(V, E)$, 反转图 $G_r=(V', E')$, 点对最短路径长度表 $d_G(s, t)$, 加入的边 $w \rightarrow v$;

输出: 介数中心度 $C_B(v), \forall v \in V$.

1) 使用算法 2 获得更新源节点组 S 和更新目的节点组 T .

2) $\forall s \in S, \forall t \in T$, 更新 s 到 t 之间的最短路径上节点的介数中心度.

① 若 $s=t$ 或 $d_G(s, t) < d_G(s, w) + 1 + d_G(v, t)$, 返回 3).

② 若 $d_G(s, t) = \infty$, $changedDis\langle s, t \rangle = d_G(s, w) + 1 + d_G(v, t)$ 中, 4) 将实际更新 $d_G(s, t)$.

③ 若 $\infty > d_G(s, t) \geq d_G(s, v) + 1 + d_G(w, t)$, 则 $changedDist\langle s, t \rangle = d_G(s, w) + 1 + d_G(v, t)$. 利用算法 2)③ 所得结果, 将当前 s 到 t 路径上点的介数中心度减去.

④ 若 $d_G(s, t) = d_G(s, v) + 1 + d_G(w, t)$, 由 1.3 节中定理 5 知, 此时点 s 到点 t 最短路径数量 $\sigma_{st}^* = \sigma_{st} + \sigma_w \times \sigma_{vt}$, 则保留原介数中心度.

⑤ 累积 $minPath(s, t)$ 中任意点 p 出现的次数 $\sigma_{st}^*(p)$. 该点 p 的介数中心度更新值为

$$C_B(p) = C_B(p) + \sigma_{st}^*(p) / \sigma_{st}^*.$$

4) 根据 $changedDist$ 更新 $d_G(s, t)$ 表.

5) 将 $v \rightarrow w$ 边插入图 G 和 G_r 中.

表 1 为图 3 在介数中心度更新 $V_2 \rightarrow V_5$ 计算过程中介数中心度变化过程:

表 1 接下来点对间最短路径更新的介数变化过程

点对 (s, t)	更改过的介数
(V_0, V_7)	$V_1: 7.5 \quad V_2: 5.5 \quad V_3: 15$
(V_0, V_5)	$V_1: 8 \quad V_2: 6 \quad V_3: 4 \quad V_4: 14$
(V_0, V_6)	$V_1: 8.5 \quad V_2: 6.5 \quad V_3: 3.5 \quad V_4: 13$
(V_0, V_8)	$V_1: 9 \quad V_2: 7 \quad V_3: 3 \quad V_4: 12$
(V_2, V_7)	$V_4: 11$
(V_2, V_5)	$V_4: 10$
(V_2, V_6)	$V_3: 9$
(V_2, V_8)	$V_1: 8$
(V_1, V_7)	$V_2: 7.5 \quad V_3: 2.5 \quad V_4: 7$
(V_1, V_5)	$V_2: 8 \quad V_3: 2 \quad V_4: 6$
(V_1, V_6)	$V_2: 8.5 \quad V_3: 1.5 \quad V_4: 5$
(V_1, V_8)	$V_2: 9 \quad V_3: 1 \quad V_4: 4$

将 $V_2 \rightarrow V_5$ 边真正加入图 3 中,本次更新操作的介数中心度最终结果如图 5 所示:

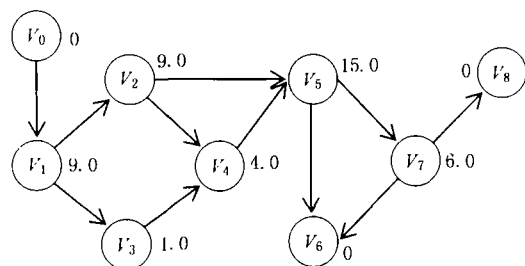


图 5 加边 $V_2 \rightarrow V_5$ 更新操作结束后图中介数中心度

3 算法理论分析

3.1 本文更新算法加速原理

本文提出的算法之所以能加快介数更新速度,理论上有两点原因:

1) 减少需要更新最短路径的点对

每次加边操作都需要全部更新介数中心度的算法之所以慢,是因为每次都会有大量的不必要的点对间最短路径需要被重新计算. 本文通过算法 2.2 找出更新源节点组和目的节点组可以减少不必要的更新点对操作.

这意味着本文提出的算法在较稀疏,且独立联通区域较多,并每个区域上点数量占整个图节点比例较小的图上性能最佳.

2) 保存点对最短路径长度表 $d_G(s, t), (s, t \in V)$

本文中主要作用有两个:

1) 按照 1.3 节提出的定理 4,判断所加边是否对点对间距离有影响. 这可以避免很多不必要的点对最短路径搜索操作.

2) 用来寻找点对间最短路径. 在更新介数中心度运算中,消耗时间最大的操作是寻找点对间最短路径. 本文通过广度优先搜索最短路径长度表提高了检索效率,但是缺点是因为需要保存长度表而消耗了大量内存空间.

3.2 算法的退化

设更新源节点组 S 和更新目的节点组 T 的节点个数分别为 $|S|$ 和 $|T|$. 则本文算法需要 $|S| \cdot |T|$ 次点对间最短路径搜索. 本文采用的是基于点对间最短路径长度表的 BFS 搜索算法,需要时间为 $O(n)$. 若在稠密图或者连通区域很大的情形下, $|S| \approx |E|, |T| \approx |E|$, 此时本文更新算法的时间复杂度

将为最坏情形,为 $O(n^3)$. 在这种情况下,本文算法会退化为原始的介数中心度计算过程,效率不如 Brandes 算法.

但是在较为稀疏或者独立区域较多且每个独立区域的节点比较少少的情况下,采用本文的算法可以大大提高效率.

4 实验

4.1 实验数据

本次实验采用了 5 种图结构可能会经常更新的真实数据进行测试和分析.

1) NLPir 微博关注关系数据

来自 <http://www.nlpir.org/> 公开的数据. 它是通过挖掘新浪微博、腾讯微博,获得用户关注关系信息.

2) Wiki-Vote 数据集

来自 <http://snap.stanford.edu/data> 公开的数据,是表示 Wikipedia 在 2008 年 1 月后的投票关系图.

3) P2P 数据集

来自 <http://snap.stanford.edu/data> 公开的数据,是对 Gnutella p2p 文件分享网络节点间关系的快照. 共有 9 个快照,本次试验采用其中 1 个快照.

4) EVA 数据集

来自 <http://denali.berkeley.edu/eva/dataset/> 公开的数据,是表示公司间隶属关系图的数据集.

5) Erdos 数据集

来自 <http://vlado.fmf.uni-lj.si/pub/networks/data> 公开的数据,是表示协作网络关系的图.

4.2 实验结果和分析

1) 在不同数据规模上的更新算法效率

本实验采用 NLPir 微博关注关系数据上的不同节点规模进行试验. 分别抽取在 1 000, 5 000, 20 000, 40 000 节点规模上研究本文算法与 Brandes^[8] 算法的比较. 对每组实验数据都进行了 1 000 次随机生成的插入边测试,将这 1 000 次更新分为 10 个区间,每个区间上的更新时间取平均值.

图 6 中图的纵坐标是平均每 100 次介数更新花费的时间,单位是毫秒(ms);横坐标是更新次数,从 0~1 000 次更新. 图 6(a)是 1 000 个节点规模时的更新时间,此时的本文更新算法跟 Brandes 算法的更新效率相差不多,这可能是因为节点规模小,更新加边更容易形成更大的连通区域,即造成 3.2 节所

提出的算法退化现象. 在图 9(f) 中第 200~300 次介数更新操作时, 本文提出的算法出现了明显的峰值, 可能是因为这个区段内插入了一条或几条连接两个大的连通区域的边, 但是总体来讲, 在 40 000 个节

点规模时的更新耗费时间还是稳定的. 在图 6(b)(c)(d) 中之所以稳定, 是由于这些图的点比较多, 插入 1 000 条边并不足以使这些图出现更新算法的退化, 所以本文算法易知比较稳定的优于 Brandes 算法.

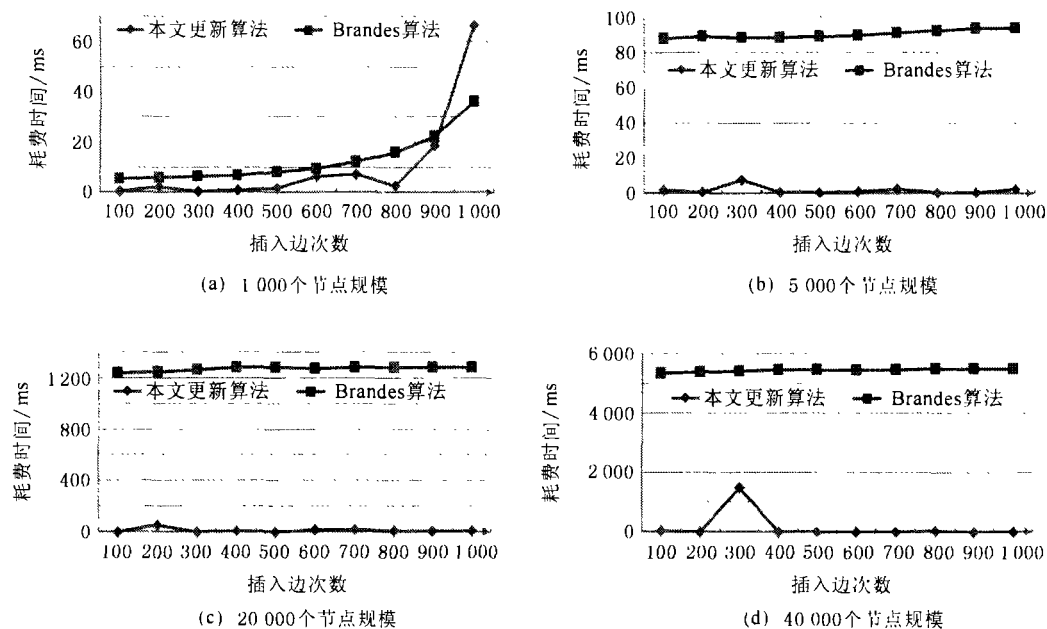


图 6 不同节点规模更新操作的效率

2) 在不同实际数据上的测试结果

表 2 列出了在 5 个数据集中, 进行 100 次更新操作的平均时间和加速比. 这几个数据集中, 加速效果最好的数据集 EVA 达到了 3 110.60 倍, 最差的 P2P 网络只有 1.48 倍. 而从 $|E|/|V|$ 可以看出, 随着图稠密程度增大, 本算法的加速效果相对变差, 出现算法的退化现象. 在稀疏且具有较多独立区域的图, 如 EVA, NLPir, Erdos 等图中, 加速效果非常明显.

表 2 在不同数据集上的加速比

数据集名称	$ V $	$ E $	$ E / V $	Brandes	本文算法	加速比
EVA	7 253	7 626	1.05	155.53	0.05	3 110.60
NLPir(10 000)	10 000	17 172	1.71	332.75	0.93	357.80
Erdos02	6 927	11 850	1.71	212.22	7.62	27.85
wikiVote	7 115	103 689	14.57	30 560.74	2 075.10	14.67
P2P	6 301	20 777	3.29	3 145.78	2 119.58	1.48

5 结束语

介数中心度是社交网络中重要的一种成员节点重要性度量指标. 本文将社交网络建模为有向无权图, 首次提出了针对有向无权图的介数中心度快速

更新算法. 该算法通过寻找更新源节点组和目的节点组算法减少了更新点对; 通过保存点对最短距离加快最短距离搜索过程. 本文算法在较稀疏或者独立区域较多的图中加速效果十分明显, 在理论分析部分对这种现象进行分析, 在实验中验证了之前的分析, 在真实数据中有很好的加速效果.

参 考 文 献

- [1] 彭卓, 王朝坤, 徐亚军, 等. 社会网络中的成员节点提升. 计算机研究与发展, 2010, 47(增刊 1): 330-336
- [2] Le Merrer E, Trédan G. Centralities: Capturing the fuzzy notion of importance in social graphs // Proc of SNS 2009. New York: ACM, 2009
- [3] Freeman L C, Borgatti S P, White D R. Centrality in valued graphs: A measure of betweenness based on network flow. Social Networks, 1991, 13(2): 141-154
- [4] Bader D, Kintali S, Madduri K, et al. Approximating betweenness centrality // Proc of WAW 2007. Berlin: Springer, 2007: 124-137
- [5] Anthonisse J. The rush in a directed graph. Amsterdam: Stichting Mathematisch Centrum, 1971
- [6] Freeman L C. A set of measures of centrality based on betweenness. Sociometry, 1977, 40: 35-41

- [7] Brandes U. On variants of shortest-path betweenness centrality and their generic computation. *Social Networks*, 2008, 30(2): 136-145
- [8] Brandes U. A faster algorithm for betweenness centrality. *The Journal of Mathematical Sociology*, 2001, 25(2): 163-177
- [9] Lee M, Lee J, Park J Y, et al. QUBE: A quick algorithm for updating betweenness centrality //Proc of WWW 2012. New York: ACM, 2012

杨建祥 男,1989年生,硕士研究生,主要研究方向为社交网络数据挖掘。

王朝坤 男,1976年生,博士,副教授,硕士生导师,先后在国内外发表论文 50 多篇,主要研究方向为非结构化数据管理、软件保护。

白易元 男,1988年生,硕士研究生,主要研究方向为社交网络数据挖掘。

郝竞超 男,1987年生,硕士研究生,主要研究方向为非结构化数据存储。

汪 浩 男,1989年生,硕士研究生,主要研究方向为社交网络数据挖掘。