

文章编号: 1003-0077(2012)04-0122-07

基于幂律分布的网络用户快速排序算法

张 玥,张宏莉,张伟哲

(哈尔滨工业大学 计算机科学与技术学院,黑龙江 哈尔滨 150001)

摘 要: 随着网络论坛、博客、微博的发展,引出社会网络中的用户排序问题。将在线网络论坛中用户映射为节点,用户评论过程中形成的回复关系映射为有向关联图,其节点度符合幂律分布。且论坛中用户的主题发布行为和回复关系符合 Pagerank 算法的互增强和随机游走特性,因此选用 Pagerank 算法排序用户影响力。该文提出的研究问题:如何提高用户排序应用中数据的存储和运行效率。天涯网络论坛中 80% 以上用户入度为 0,据此,根据入度是否为 0 划分为两个集合,对入度为 0 集合按出度构造链接表,设计了基于集合划分的高效排序算法 SD-Rank。SD-Rank 时空复杂性为 $O(V')$, V' 为入度非 0 节点集。对天涯网络论坛真实用户数据的实验结果表明:SD-Rank 算法时空复杂性优于 Pagerank 算法。

关键词: 幂律;入度;集合划分;快速排序

中图分类号: TP391 **文献标识码:** A

Quick Ranking Algorithm for Network User Based on Power Law Distribution

ZHANG Yue, ZHANG Hongli, ZHANG Weizhe

(School of Computer Science and Technology, Harbin Institute of Technology, Harbin, Heilongjiang 150001, China)

Abstract: With the wide application of BBS, blog and micro blog etc, how to rank the user becomes a well-recognized research issue, especially in the social network area. The paper analyzes the users relational graph in network BBS, representing the correlation graph by users and replies between users, and reveals its power law distribution in in/out degree. Owing the fact that the user's behavior of posting and replying is in accordance with Pagerank's characteristics of random walking and mutual-enforcement, the user's influence is ranked with Pagerank algorithm. This paper further addresses the issue of time-space ratio in computation resulted by the exponent growth of user's number. Based on the observation that over 80 percent user's indegree is 0. Using list structure designs the efficient set-division-ranking arithmetic (SD-Rank) is designed by via list structure after dividing users into two sets: 0-indegree in set0 and non-0-indegree in set1. Through set partitions according to degree distribution, the time-space complexity of SD-Rank is decreased from $O(V+E)$ to $O(V')$, in which V' is the size of set1. Experiment on TIANYA BBS dataset shows that SD-Rank is more efficient than Pagerank.

Key words: power law; in degree; set division; quick rank

1 引言

随着 Web2.0 的兴起,计算机网络将用户从传统的信息接收者转变为信息制造者,网络论坛、博

客、微博作为新兴媒体对传统媒体产生了极大冲击,以前的社会影响主要由传统媒体决定和控制,但新兴媒体下用户发布的一条信息就可能引发蝴蝶效应,从天涯网络论坛中大量长期的对“药家鑫”事件的讨论到“郭美美”的炫富微博,无不显示出网络用

收稿日期: 2011-09-13 定稿日期: 2012-03-10
基金项目: 国家 863 自然科学基金(2010AA012504); 国家 973 重点基础研究发展规划项目基金(G2011CB302605); 国家自然科学基金(61173145)
作者简介: 张玥(1975—),女,博士研究生,讲师,主要研究方向为社会计算和数据挖掘;张宏莉(1973—),女,博士,教授,博士生导师,主要研究方向为网络安全与社会计算;张伟哲(1976—),男,博士,副教授,主要研究方向为网络安全与社会计算。

户在新兴媒体中“草根”阶层对社会发展的积极影响。如何评价新兴媒体中用户的影响力是近年来社会网络研究中的一个重要内容,文献[1]分析博客中用户影响力,文献[2-7]分析微博中用户影响力,用户影响力计算主要根据用户的活跃性和受众性以及发布的内容,对社会网络中的用户进行排序。文献[8-10]量化了社会网络中存在的影响力。文献[11-12]对主题进行区分量化了主题级用户影响力。针对大规模数据下影响力计算困难性,文献[11]采用分布式框架在 Map-Reduce 上量化影响力强度。

用户影响力计算结果与对影响力的定义和分析有很大关系,文献[4,7]中比较多种影响力计算方法,均认为影响力值由计算方法决定。文献[4]比较了 twitter 中利用粉丝数、依据粉线关系所形成的网络拓扑采用 Pagerank 算法[13]、回复数三种方法的排序比较,前两种方法都是反映了 twitter 用户在 twitter 空间中的被认识程度,该影响力是其综合表现和社会传播的结果,前两种方法结果相似,但没有明确反映出其内容影响力,后者与前两者结果不同,而依据回复数的排序结果仅部分反映了内容影响力而未考虑到传播的影响^[1]。网络用户产生影响过程也是信息扩散过程,从信息扩散角度研究影响力包括文献[14-15]等,文献[14]分析博客空间中的链接模式,文献[15]计算具有最在化影响力的少量用户。博客、微博和网络论坛信息传播方式不同,博客、微博是根据用户信息定制的定向推送式传播;网络论坛是公开的大众用户讨论场所,基于服务器的集中式讨论,网络论坛中信息全部用户可见。

早期应用于 Google 搜索引擎的网页排序算法 Pagerank^[1]利用网页间链接关系构造有向关联图,依据随机游走和关联图的排序算法,网页间的指向关系是网页分值的主要依据。Pagerank 算法设计思想也适用于网络论坛中:论坛中用户 A 回复了用户 B 是基于人主观判断后对 B 的一种认同表现,因此我们认为 B 对 A 产生了影响,而且论坛中主题内用户自然形成基于话题的社区。由此可根据主题内用户间回复关系,构造用户关联图,应用 Pagerank 算法排序网络用户。但 Pagerank 算法没有深入分析节点度分布,该算法运行效率有提高空间。本文的研究问题。主要针对大规模的社会网络数据,在对用户复杂排序应用中如何提高数据的存储和运行效率。本文在网络论坛中用户度分布符合幂律特性,对 Pagerank 算法依据度分布进行数据结构优化,按入度和出度进行集合划分,采用链表数据结

构,实现基于集合划分的快速排序算法 SD-Rank。在天涯论坛上的用户排序实验中,算法时空复杂性大大降低。

2 相关工作

1) Pagerank 算法

经典的网页排序算法包括 Pagerank 算法^[16]和 HITS 算法^[17]。Pagerank 算法根据页面间指向关系迭代计算页面的排序值,被大量指向的页面其排序值高,排序值高的网页所指向的页面排序值也高,具有互增强特性;Pagerank 算法还引入了随机游走机制,即每次以一定的概率随机选择节点以防止进入连通子图中。Pagerank 算法可表示为式(1):

$$X^{(i+1)} = sAX^{(i)} + ((1-s)/n)I \quad (1)$$

A 是网页间关联矩阵,X 是 Pagerank 迭代向量, $X^{(0)}$ 是初始随机游走向量,s 是阻尼系数,1-s 为随机游走参数。归一化后,网页排序值最后收敛于特征值为 1 对应的主特征向量,与 $X^{(0)}$ 无关,但 $X^{(0)}$ 影响算法迭代速度。

2) Pagerank 改进快速算法

提高 Pagerank 算法运算速度主要从算法和数据两个角度,文献[18]总结了 Pagerank 算法的本质:节点 rank 值主要取决于连接该节点的边,且在边上仅传递 rank 值。文献[18]加速计算方法:a)减少迭代次数法^[19]。在计算排序值的收敛向量时,Pagerank 的 Power 算法迭代次数取决于所选取的初始值 $X^{(0)}$ 。改进算法:迭代过程中修改迭代向量,令 $X^{(i+1)} = X^{(i)} + Z^{(i)}$ 使其快速逼近主特征向量,且迭代过程中利用启发式修正 Z 向量;b)划分数据。按节点间连通性划分为多个连通区域^[9],单遍迭代过程中,在节点间连接稠密区域可连续计算几次再进入下一连通区域继续;c)减少计算量。当节点间关联图随时间变化时,若关联图变化不大可采用前一次 rank 结果作为启发式来快速排序,当节点 rank 值稳定时不必进行下轮迭代,仅对未收敛于稳定概率分布的节点进行迭代以减少计算量。

3) 本文思想

Pagerank 算法本质思想为在关联网络中以一定概率随机选择节点,然后在该节点沿出边向外游走,节点 rank 值主要取决于连接该节点的边,且在边上仅传递 rank 值,该思想也适用于网络用户的影响力计算,但本文不同于文献[18-19],基于大量节点入度为 0 特点,对入度为 0 集合进行优化处理以

减少存储和运输复杂度。本文与上述 Pagerank 加速算法不同之处在于,基于网络论坛中 80% 以上用户入度为 0 的数据特征,提出基于入度是否为 0 进行集合划分来加速 pagerank 算法,而该数据特征和文献[9]基于集合划分思想一致。入度为 0 节点为论坛中仅发表评论,没有引发别人评论的用户,这类用户对论坛的影响仅增加了帖子数量而没有产生交互性影响,用户实质性影响表现为发表主题以及在主题中发表见解并引发正负面争论。

3 网络用户快速排序算法

3.1 矩阵与图的稀疏性

对于一个 $n \times n$ 矩阵,其存储空间为 n^2 。一个 n 阶方阵与一个 n 维列向量相乘,运算复杂度为 $O(n^2)$ (需要 $n * (n$ 个乘法 $+(n-1)$ 个加法))。当 n 数量级较大时存储运算开销都较大。当 $n \times n$ 矩阵中多数值为 0 时,为稀疏矩阵^[20]。

图 $G=(V,E)$,图中节点数 $|G|$,边数 $|E|$ 。若图 G 为非稀疏图,至少要求 $|E| \geq \frac{|G|(|G|-1)}{2}$ 。令 $|E|=k|G|$,即边数是节点数的 k 倍。有 $k|G| \geq \frac{|G|(|G|-1)}{2}$,即 $k \geq \frac{|G|-1}{2}$ 。故至少要求节点度近似是节点数的 $1/2$ 倍。对应到社会网络中,需要满足每个人和网络中一半以上人有关联关系,这是非常不现实的,实际中度平均值很小, $k < n$,通常 $k < 10$ 。

3.2 关联图的邻接表表示

稀疏图可用邻接表表示,如图 1 所示。用邻接表形式表示存储空间小且计算复杂度低^[21]。有向图 $G=(V,E)$ 可表示为一个包含 $|V|$ 个链表的数组 Adj 。 $\forall u,v \in V, (u,v) \in E$,则 v 在 u 的邻接表中。图 G 的邻接表所需存储空间为 $\Theta(V+E)$ 。

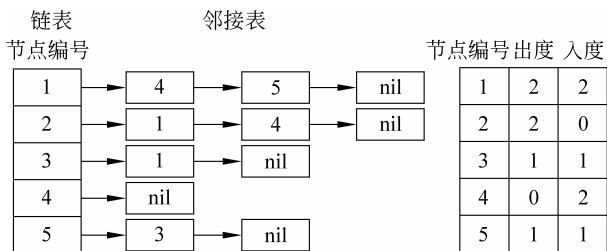


图 1 用户关联图所对应的邻接表表示

3.3 基于邻接表的 Pagerank 计算

根据 Pagerank 算法(式 1),图 $G=(V,E)$ 中节点 v 的 $rank$ 值由两部分而来。

1) 当边 $(u,v) \in E$ 时, u 的 $rank$ 值 $rank[u]$ 沿关联矩阵游走,乘以因子 s 后按其出度均分到 u 所指向的节点,故 v 的 $rank$ 值从指向其的节点 u 处传播而来。当多个点 u_1, u_2, \dots, u_k 都指向 v 时, $(u_1, v) \in E, (u_2, v) \in E, (u_3, v) \in E, \dots (u_k, v) \in E, Rank[v] = \sum rank[u_i] * s / out[u_i]$

2) 每个节点随机游走所分的 $rank$ 值,每个节点都乘以系数 $(1-s)$ 后均分到所有节点:

$$Rank[v] = (1-s) * \sum (u \in G) rank[u] / n$$

$Rank$ 计算过程见算法 1。

算法 1 基于链表方式的排序算法

```
Input (链表 B, 初始 rank 向量 A0, 出度向量 out, 阻尼系数 s)
A1: 迭代过程保存 rank 值向量, 初始全 0, n 节点数
A2: 上一次迭代 rank 向量, 初始全 0, e, rank 向量收敛误差, k 迭代次数
function pagerank-iterate-computation(L, A0)
1   while (||A0-A2||_2 < e) //迭代 k 次, 向量 2 范数收敛误差 < e 结束
2   {   rank=0;   A2=A1;
3       For (i=1, i++, i<=n)   rank=rank+
                               A0[i];
4       For (i=1, i++, i<=n)   A1[i]=rank *
                               (1-s)/n;
5       For (count=0; count++ < n)
                               //循环 n 次
6       {   While (当 B[count]链表未考察结束)
                               //边数
7           {   read j // (i,j)? E
8               A1[j]= A1[j]+A1[count]*s/
                               out[count];
9           }
10      }
11      A0=A1;
12      }   Output A0
```

算法 2 基于 SD-Rank 的集合部分排序算法

```
Compute-rank-b0  计算 b0 桶中 rank 值的分配
{ 1   max-out=1; //出度最大值
  2   For (A[v]=w) //对 B0 中节点 v 的邻接表
                               按出度放入相应的桶中
  3       {   i=out[v];
  4           Insert List v into B0[i];
                               //插入排序,排序时计算节点出现次数
```

```
5           If (i>max-out )   max-out=i;
6       }
7       For(i=1 to max-out)
8           {   while list B0[i] is not null
9               {   read 节点 j and weight[j];
10                  A1[j]= A1[j]+A0[k]'s'/
                      weight[j]/i;
                      //节点 k 入度=0   }   }
```

3.4 基于集合划分的 SD-rank 改进算法

定义 1：入度为 0 的点所构成的集合称为 *set0*；入度大于 0 的点所构成的集合称为 *set1*。

定义 2：由入度为 0 的点所引发的边的集合为 *edges0*。其他边的集合为 *edges1*。*edges0* 集合即由 *set0* 中节点指向 *set1* 中节点所构成的边。*edges1* 集合即由 *set1* 集合内部节点间关联所构成的边。

性质 1 点集合 *set0* 和 *set1* 不相交，且点的总集由 *set0* 和 *set1* 构成， $setV=set0 \cup set1$ 。

性质 2 边集合 *edges0* 和 *edges1* 不相交，且 $edgesE=edges0 \cup edges1$ 。

关联图的点集合和边集合划分后如图 2 所示。*set1* 集合中节点影响力值：1)由随机游走；2)由集合内部相互指向得到；3)由集合 0 节点指向得到。

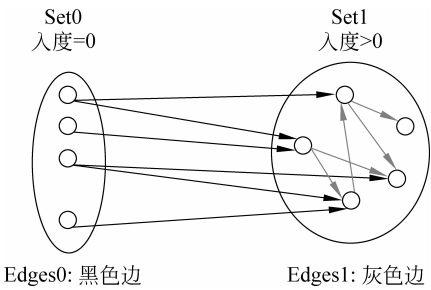


图 2 关联图基于入度的集合划分

3.4.1 集合划分后进行优化

Set0 中节点入度为 0，若其初始 *rank* 值相同，出度相同时其向外传播的 *rank* 值也相同，按出度值重新划分桶，那么链接表中数组 *B* 的元素数降低到 *set0* 的出度数。*Set0* 中出度相同的节点指向相同节点时，累计指向节点出现次数，邻接表采用加权表示，如图 3 所示。集合划分可采用对节点着色法表示，*set0* 集合用 w 表示；*set1* 点颜色为用 b 表示。对 *set0* 集合引发的 *edges0* 边集合，应用加权邻接表表示后的 *rank* 计算过程见算法 2。

3.4.2 运算复杂性分析

对 *B0* 链表的运算复杂度分析：*Compute-rank-b0* 的第二行对 *B0* 中节点进行处理，循环了 *count_*

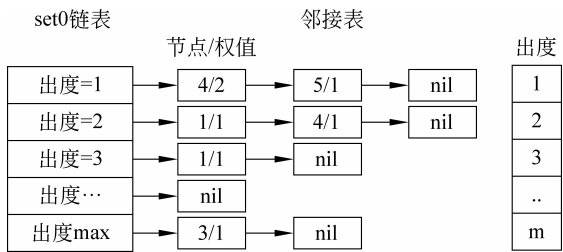


图 3 入度为 0 的节点加权邻接表表示

b0 次；第四行采用插入排序法链接到第 *i* 个桶，最少插入 *i* 次，最多插入 *count_b1* 次；第 7 行循环了 *max-out* 次；第八行同第四行。运算复杂度

$$T_n = \Theta(n) + \sum_{i=1}^{count_b1} O(n_i^2), \quad (n = \max-out),$$

令 $c=count_b1$

对 $T(n)$ 等式两边取数学期望，有

$$\begin{aligned} E[T(n)] &= E[\Theta(n) + \sum_{i=1}^c O(n_i^2)] \\ &= \Theta(n) + \sum_{i=1}^c O(E[n_i^2]) \end{aligned}$$

令 $X_{ij}=I\{A[j] \text{ 落入桶 } i \text{ 中}\}$ ， $i \in [1, \max-out]$ ， $j \in [1, c]$ 。

假定 X_{ij} 为 1 的概率为 $\frac{1}{c}$ ，则 $E[X_{ij}^2] = \frac{1}{c}$ ，

$$E[X_{ij}]E[X_{ik}] = \frac{1}{c^2}, E[n_i^2] = 2 - \frac{1}{c}。$$

$T(n)$ 的数学期望为 $\Theta(n) + nO\left(2 - \frac{1}{c}\right)$ ，因为 $n = \max-out$ ，*max-out* 最大为 *count_b1*，也就是 *max-out* 最大为 *c*，故 $T(n)$ 的数学期望为 $\Theta(c)$ 。

对 *B0* 桶进行改进后，运算复杂度下降到 $\Theta(count_b1)$ 。*count_b1* 为入度不为 0 用户数量。即：对入度为 0 节点按其出度值划分桶后其运行复杂度为 $\Theta(|set1|)$ 。

4 实验结果

4.1 数据描述

天涯网络论坛是中国门户网站论坛，用户活跃，内容广泛，部分主题讨论深入交互性好。实验数据集采集自天涯网络论坛天涯杂谈版块，采用工具为实验室开发的爬虫软件，于 2010 年 12 月 21 日按主题进行广度爬行。过滤掉回复数少于 50 的主题。抽取了 11 月 26 日到 12 月 7 日共 12 天的数据进行了测试。数据按天划分，提取出用户间回复关系，默

认情况下回复给主题创建者。统计结果如表 1 所示。

表 1 数据集统计

	11 月					12 月						
	26	27	28	29	30	01	02	03	04	05	06	07
用户数	12 277	6 209	5 572	5 572	5 184	6 422	6 836	6 189	7 870	5 725	7 621	7 621
边数	15 738	12 240	10 802	10 447	10 673	13 345	14 418	13 830	14 794	11 548	15 131	16 215

4.2 入度和出度统计结果

统计 11 月 26 日到 12 月 7 日的入度为 0 用户数及所占比例,如图 4 所示。图 4 左为日入度为 0 和出度为 1 用户数量,图 4 右为日入度为 0 和出度

为 1 用户占总数比例。平均日入度为 0 用户数 5 566。入度为 0 用户占总人数日平均比例为 86.16%。出度为 1 用户数日平均 4 430 人。出度为 1 用户占总人数日平均比例为 64.98%。

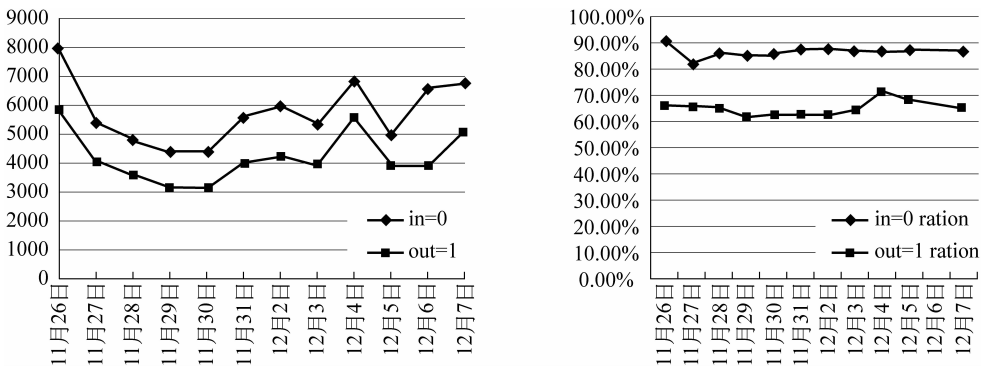


图 4 入度为 0 和出度为 1 日用户数统计及相对比例

选取 11 月 23 日作为样本点,统计入度和入度的频数、出度和出度的频数。图 5 为用户入度频数、出度频数对数坐标图,由图可看出入度、出度符合幂律分布。文献[22-23]分析了 web 入度符合幂律分布,web 入度为 k 的概率与 $1/k^a$ ($2 < a < 3$) 成正比^[24]。

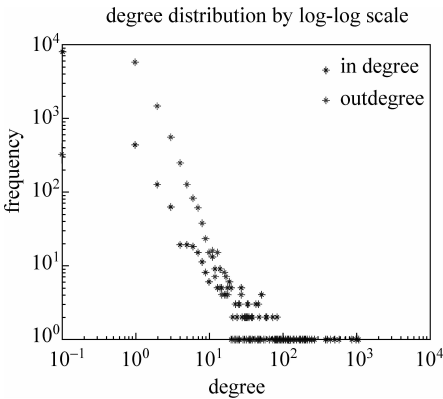


图 5 11 月 23 日度对数对数分布图

4.3 SD-Rank 算法和 Pagerank 算法运行结果比较

为考察 SD-Rank 算法和 Pagerank 算法排序结

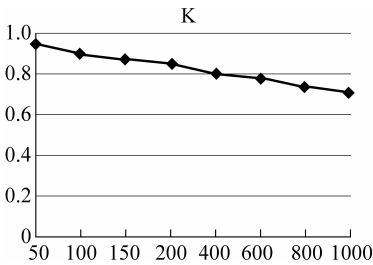


图 6 比较两种算法排序结果相似性

果的相似性,采用 Kendall's tau 排序相关性比较的改进公式^[4,25-26]比较排序结果差异,Kendall's tau 公式为:

$$K_{\tau}^{(0)}(R_1, R_2) = \sum_{r_1, r_2 \in R_1 \cup R_2} \bar{K}_{r_1, r_2}(R_1, R_2) \quad (2)$$

其中 R_1 和 R_2 是两个排序结果。 $\bar{K}_{r_1, r_2}(R_1, R_2)=1$ 要求 1)当 r_1 和 r_2 分别出现在 R_1 和 R_2 中; 2)一个排序结果中 r_1 比 r_2 排序值高且仅 r_2 出现在另一个结果中;3) r_1 和 r_2 均在两个结果中且排序相反;否则 $\bar{K}_{r_1, r_2}(R_1, R_2)=0$ 。 K 定义为式(3),其中 k 是待比较元素个数。

$$K = 1 - \frac{K_{\tau}^{(0)}(R_1, R_2)}{k^2} \tag{3}$$

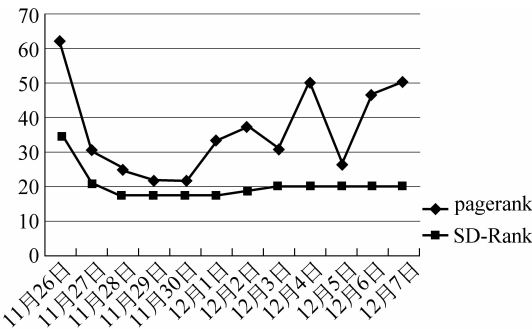
$0 \leq K \leq 1$, $K=0$ 表示 R_1 和 R_2 排序结果完全不同, $K=1$ 表示 R_1 和 R_2 排序结果完全相同。

表 2 11 月 23-24 日排序 top5 用户结果

11 月 23 日				11 月 24 日			
s=0.8		s=0.9		s=0.8		s=0.9	
name	value	name	value	name	value	name	value
纳尼兔 2010	0.107 5	纳尼兔 2010	0.139 1	柳随风 L	0.057 2	柳随风 L	0.075 8
秀才江湖	0.064 1	秀才江湖	0.101 5	纳尼兔 2010	0.035 1	纳尼兔 2010	0.047 6
zyl66523	0.021 8	zyl66523	0.032 1	秀才江湖	0.034 5	红色 De 脚印	0.036 2
红色 De 脚印	0.015 9	红色 De 脚印	0.024 4	红色 De 脚印	0.021 2	zyl66523	0.028 4
窗台上的灵猫	0.015 8	窗台上的灵猫	0.020 9	姐会过了求你	0.020 4	秀才江湖	0.027 4

4.4 SD-Rank 算法运行效率

比较运行 Pagerank 和 SD-Rank 算法的运行时



间。运行时间对比如图 7 左所示提高 4-30s。运行效率提高比例如图 7 右所示。SD-Rank 算法运行效率平均提高了 39%。

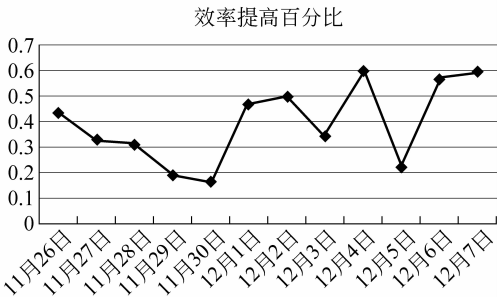


图 7 算法运行时间和 SD-Rank 提高效率

5 结论

本文以网络论坛为用户排序应用背景,提取用户关联图。基于用户度符合幂律分布,改进 Page-rank 算法,将用户划分为入度是否为 0 的两个集合。对入度为 0 的用户集合,再按出度构造链接表,采用加权链接数据结构。基于度分布进行集合划分,加权链接结构的 SD-Rank 算法,时空复杂度降低为 $O(V')$, V' 为入度不为 0 节点集合,大大降低了排序的存储空间和运行效率。

参考文献

[1] Agarwal N., Liu H., Tang L., et al. Identifying the influential bloggers in a community[C]//Proceedings

of the international conference on Web search and web datamining (ICWSM '08), New York, US, ACM, 2008:207-218.

[2] Paul N. Bennett, Krysta Svore, Susan T. Dumais. Classification-enhanced Ranking [C]//Proceedings of the 19th international conference on World Wide Web, Raleigh, NC, USA, 2010:111-120.

[3] T. L. Fond, J. Neville. Randomization Tests for Distinguishing Social Influence and Homophily Effects [C]//Proceedings of the 19th international conference on World Wide Web, Raleigh, NC, USA, 2010:601-610.

[4] H. Kwak, C. Lee, H. Park, et al. What is Twitter, a Social Network or a News Media[C]//Proceedings of the 19th international conference on World Wide Web, Raleigh, NC, USA, 2010:591-600.

[5] S. Wu, J. M. Hofman, W. A. Mason, et al. Who says What to Whom on Twitter[C]//Proceedings of

- the 20th international conference on World Wide Web, Madrid, India, 2011:705-714.
- [6] M. Cha, H. Haddadi, F. Benevenuto, et al. Measuring user influence on twitter: The million follower fallacy[C]//Proceedings of the 4th International AAAI Conference on Weblogs and Social Media, Washington DC, 2010.
 - [7] J. Weng, E. P. Lim, J. Jiang, et al. Twitterank: finding topic-sensitive influential twitterers[C]//Proceedings of the 4th ACM international conference on Web search and data mining(WSDM), 2010:261-270.
 - [8] Dan Cosley, D. Huttenlocher, Jon Kleinberg, et al. Sequential Influence Models in Social Networks[C]//Proceedings of the International Conference on Weblogs and Social Media(ICWSM), 2010.
 - [9] A. Anagnostopoulos, R. Kumar, M. Mahdian. Influence and correlation in social networks[C]//Proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining, 2008:7-15.
 - [10] P. Singla, M. Richardson. Yes, there is a correlation: from social networks to personal behavior on the web[C]//Proceedings of the 17th international conference on World Wide Web, 2008:655-664.
 - [11] Jie Tang, Jimeng Sun, Chi Wang, et al. Social Influence Analysis in Large-scale Networks[C]//Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, 2009:721-730.
 - [12] Aditya Pai, Scott Counts. Identifying Topical Authorities in Microblogs[C]//Proceedings of the fourth ACM international conference on Web search and data mining(WSDM), 2011:45-54.
 - [13] J. Leskovec, E. Horvitz. Planetary-scale views on a large instant-messaging network[C]//Proceedings of the 17th international conference on World Wide Web. Beijing, China, 2008:915-924.
 - [14] D. Kemp, J. Kleinberg, E. Tardos. Maximizing the spread of influence through a social network[C]//Proceedings of the 9th ACM SIGKDD international conference on knowledge discovery and data mining, 2003:137-146.
 - [15] S Brin. L Page, R Motwani, T Winograd. The pagerank citation ranking: Bringing order to the web[R]. Technical report, Stanford University, 1999.
 - [16] J Kleinberg. Authoritative sources in a hyperlinked environment[J]. Journal of the ACM, 1999, 46(5): 604-632
 - [17] Frank Mcsherry. A uniform Approach to Accelerated Pagerank Computation[C]//Proceedings of the 14th international conference on WWW, 2005:575-582.
 - [18] Sepandar D. Kamvar, T H. Haveliwala, C D. Manning, et al. Extrapolation Methods for Accelerating Pagerank[C]//Proceedings of the 12th international conference on WWW, 2003:261-270.
 - [19] Stoer Josef, R. Bulirsch. Introduction to Numerical Analysis[M]. Berlin, Dover Publications. 2002: 619
 - [20] Thomas A. Smith. The web of law[J]. San Diego Law Review, 2007, 44(309).
 - [21] A. Broder, R. Kumar, F. Maghoul, et al. Graph structure in the Web[C]//Proceedings of the 9th International World Wide Web Conference, 2000:309-320.
 - [22] David Easley, Jon Kleinberg. Networks, Crowds, and Markets: Reasoning about a Highly Connected World[M]. Cambridge University Press, 2010. P546
 - [23] R. Fagin, R. Kumar, D. Sivakumar. Comparing top k lists[C]//Proceedings of the 14th annual ACM-SIAM symposium on discrete algorithms, 2003:28-36.
 - [24] F. McCown, M. L. Nelson. Agreeing to disagree: search engines and their public interfaces[C]//Proceedings of the 7th ACM/IEEE-CS joint conference on digital libraries. ACM, 2007:309-318.
 - [25] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, et al. Introduction to Algorithms[M]. The MIT Press, 2001. P527
 - [26] N. Agrawal, H. Liu, L. Tang, et al. Identifying the Influential Bloggers in a Community[C]//Proceedings of the international conference on Web search and web data mining(WSDM'08), 2008:207-218.