软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2012,23(12):3175−3186 [doi: 10.3724/SP.J.1001.2012.04206]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

MapReduce 环境下的并行复杂网络链路预测*

饶 君+, 吴 斌, 东昱晓

(北京邮电大学 北京市智能通信软件与多媒体重点实验室,北京 100876)

Parallel Link Prediction in Complex Network Using MapReduce

RAO Jun+, WU Bin, DONG Yu-Xiao

(Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia, Beijing University of Posts and Telecommunication, Beijing 100876, China)

+ Corresponding author: E-mail: raojun_06@bupt.edu.cn

Rao J, Wu B, Dong YX. Parallel link prediction in complex network using MapReduce. *Journal of Software*, 2012,23(12):3175−3186 (in Chinese). http://www.jos.org.cn/1000-9825/4206.htm

Abstract: To apply link prediction methods into large-scale complex network, this paper designs and implements a parallel link prediction algorithm based on MapReduce, which includes nine similarity Indices via local information. The parallel link prediction algorithm has a time complexity of $O(N)$ in sparse networks. First, the paper verifies the validity of the algorithm on public datasets, increase in the extraction factor, recall ascends, and precision descends. The experimental results on ten large-scale datasets of variety network types show that the parallel link prediction algorithm is more effective than traditional ones, and its running time decreases with more compute units. The upper and lower bounds of AUC (area under a receiver operating characteristic curve) are proposed. The experimental results show the median of the upper and lower bounds are close to the real value of AUC, which focuses on whether prediction score is zero rather than the actual score value. The network average clustering coefficient has the greatest impact on AUC among most topological features and AUC rises as the network average clustering coefficient increases.

Key words: link prediction; complex network; local information; parallel algorithm; MapReduce

摘 要: 为使链路预测应用于大型复杂网络,设计并实现了一种基于 MapReduce 计算模型的并行链路预测算法,包含了 9 种基于局部信息的相似性指标,在稀疏网络上的时间复杂度为 $O(N)$.首先,在公共数据集上验证了并行算法的有效性,随着抽取因子的增加,召回率升高而准确率下降.在不同类型的 10 个大规模复杂网络数据集上的实验结果表明,基于 MapReduce 计算模型的并行链路预测算法比传统算法具有更高的效率,算法的运行时间随着并行程度的增加而下降.提出并证明了 AUC(area under a receiver operating characteristic curve)评价指标的上下界,实验表明,上下界的中值和实际 AUC 值很接近,并且 AUC 评价指标侧重于预测分数值是否为 0 而不是分数值的大小.在网络拓扑性质中,平均聚集系数对 AUC 值的影响最大,并且 AUC 值随着网络平均聚集系数的增加而提高.

关键词: 链路预测;复杂网络;局部信息;并行算法;MapReduce

* 基金项目: 国家自然科学基金(90924029, 60905025, 61074128)
收稿时间: 2011-07-18; 定稿时间: 2012-02-28

中图法分类号: TP393 文献标识码: A

复杂网络中的链路预测作为图挖掘方向之一,近年来受到了广泛的关注.链路预测是指如何通过已知的网络拓扑结构以及网络节点属性等信息,预测网络中尚未产生连边的两个节点之间产生链接的可能性[1].链路预测包括对未知链接和未来链接的预测,其中,对未来链接的预测考虑链接的时间因素,即认为两个节点在不同时间上产生的链接是不同的.

链路预测问题也可以归结为节点相似性问题,然而在传统的求相似性的算法中只考虑了节点属性等外部信息,而没有考虑网络的拓扑信息.文献[2]根据 Web 网页的内容进行万维网的链路预测.文献[3]用节点间的相似性进行链路预测,相似性只用到了节点的属性信息.此外,也有文献同时利用了网络的拓扑信息和节点属性信息来进行链路预测.文献[4]基于网络的拓扑结构和节点的属性信息建立了一个局部的条件概率进行链路预测.文献[5]用传统的分类模型预测科学文献的引用关系,该模型不仅用到了引文网络信息,还有作者信息、期刊信息以及文章内容等外部信息.概率模型和传统的分类模型可以同时考虑网络的结构信息和节点属性信息,但是计算的复杂性以及节点外在属性信息在获取上的难度,造成了该类方法应用的局限性.

利用网络的拓扑结构进行链路预测通用性强,对于任何网络都适用,而且网络拓扑结构的获取相对容易,所以基于网络结构进行链路预测的方法受到越来越多的关注.文献[6]给出了 10 种计算复杂度为 $O(N\langle k\rangle^2)$的基于节点局部信息的相似性指标,并且通过实验说明 RA 算法在 10 种算法中表现得最出色.基于路径的相似性指标有局部路径(LP)指标[7]、Katz 指标[8]和 LHN-II 指标[9],计算复杂度分别为 $O(N\langle k\rangle^3)$,$O(N^3)$和 $O(N^3)$.此外,文献[10]给出了 6 种基于随机游走的相似性指标,计算复杂度为 $O(N^3)$和 $O(N\langle k\rangle^s)$,其中,s 为随机游走的步数.实验说明,基于路径的相似性指标优于基于节点局部信息的相似性指标,而基于随机游走的相似性指标是 3 类相似性指标中预测准确率最高的.

由于算法的计算复杂度较高和单台计算机内存的限制,传统的链路预测算法并不能够有效地处理大规模的复杂网络,从而需使用并行环境来解决这一问题.由 Google 提出的 MapReduce 是一种简洁抽象的并行计算模型,Apache 基金会在此基础之上实现了开源的 Hadoop 并行平台,已在学术界和工业界广泛应用.文献[11]提出了 3 种基于 MapReduce 的设计模式,用以加快复杂网络中图挖掘算法的处理速度.文献[12]描述了基于 MapReduce 的图挖掘工具 PEGASUS,它实现了大部分图挖掘中的典型算法,能处理 PB 级数据.尽管基于路径的相似性指标和基于随机游走的相似性指标在链路预测上的准确率较高,但是它们的时间复杂度也随之上升,而且需要考虑高阶邻居信息和网络拓扑信息,并不适合用 MapReduce 实现.然而,基于节点局部信息的相似性指标只需要知道节点的一阶邻居信息,因此可以将网络划分为以每个节点为中心的局部子图,从而适合于并行处理.

本文针对大规模复杂网络的特点,在开放并行平台 Hadoop 之上实现了基于节点局部信息的相似性指标,通过实验数据说明了算法的正确性,并在 10 个超大规模网络上进行了实验分析.此外,还分析了网络拓扑结构对链路预测 AUC 评价指标的影响.综上所述,本文的主要贡献如下:

(1) 使用 MapReduce 并行计算模型,在 Hadoop 平台上实现了基于节点局部信息的并行链路预测算法,包括 9 种节点相似性指标.算法在稀疏网络上的计算复杂度为 $O(N/U)$,N 为网络节点数,U 为 Hadoop 集群节点数.算法非常适用于超大规模的复杂网络.通过在标准数据集和大规模数据集上的实验,表明了并行算法的正确性和高效性;

(2) 给出了链路预测 AUC 评价指标的上下界.通过实验表明,AUC 上下界的中值和 AUC 的实际值很接近,说明了 AUC 算法虽然比较的是测试边集和不存在边集中预测分数值的大小,然而 AUC 却取决于测试边集和不存在边集中预测分数值是否为 0 的比例;

(3) 分析了网络拓扑结构对 AUC 值的影响.发现网络平均聚集系数对 AUC 值的影响最大,并且 AUC 值随着网络平均聚集系数的升高而增加.通过将网络社团化并将链路预测可视化展示后,发现基于节点局部信息相似性指标预测出的链路绝大部分都在聚集系数相对较高的子图之内.

本文第 1 节介绍链路预测的相关概念和性质,给出基于局部信息的相似性指标和评价方法.第 2 节介绍

MapReduce 并行计算模型并提出基于该模型的并行链路预测算法,进行时间和空间复杂度的分析,并且给出计算过程示例.第 3 节在标准数据集上进行实验对并行算法进行有效性验证,并在大规模数据集上验证并行算法的高效性.第 4 节对链路预测 AUC 评价指标进行分析,给出 AUC 的上下界;此外,还对链路预测进行可视化展示,并分析网络拓扑结构对链路预测的影响.第 5 节是本文的总结,并提出下一步的工作.

1 基于局部信息的链路预测

1.1 相关概念和性质

定义 $G(V,E)$为无向网络,其中,V 为节点集合,E 为边集合.网络总的节点数为 N,边数为 M.此网络共有 $N(N-1)/2$ 个节点对,即全集 U.给定一种链路预测的方法,对每个未连接的节点对$(x,y)\in U-E$ 赋予分数值 S_{xy},然后按照该分数值从大到小排序,排在最前面的节点对出现链接的概率最大.

基于局部信息的链路预测一共有 10 种方法[6],表 1 总结了 10 种基于局部信息的相似性指标的定义公式.对于网络中的节点 x,定义它的邻居为$\Gamma(x)$,$k(x)=|\Gamma(x)|$为节点 x 的度;S_{xy} 为点对(x,y)的预测分数,即相似性.与其他 9 种指标不同,PA 指标并没有用到共同邻居的信息.为保持一致性,本文讨论的基于局部信息的相似性指标不包括 PA 指标.

Table 1 Ten similarity indices based on local information

表 1 10 种基于节点局部信息的相似性指标

Indices	Definition	Indices	Definition
Common neighbours (CN)	$S_{xy}=\|\Gamma(x)\cap\Gamma(y)\|$	Hub depressed index (HDI)	$S_{xy}=\frac{\|\Gamma(x)\cap\Gamma(y)\|}{\max\{k(x),k(y)\}}$
Salton index	$S_{xy}=\frac{\|\Gamma(x)\cap\Gamma(y)\|}{\sqrt{k(x)\times k(y)}}$	Leicht-Holme-Newman index (LHN-I)	$S_{xy}=\frac{\|\Gamma(x)\cap\Gamma(y)\|}{k(x)\times k(y)}$
Jaccard index	$S_{xy}=\frac{\|\Gamma(x)\cap\Gamma(y)\|}{\|\Gamma(x)\cup\Gamma(y)\|}$	Preferential attachment index (PA)	$S_{xy}=k(x)\times k(y)$
Sorenson index	$S_{xy}=\frac{2\|\Gamma(x)\cap\Gamma(y)\|}{k(x)+k(y)}$	Adamic-Adar index (AA)	$S_{xy}=\sum_{z\in\Gamma(x)\cap\Gamma(y)}\frac{1}{\log k(z)}$
Hub promoted index (HPI)	$S_{xy}=\frac{\|\Gamma(x)\cap\Gamma(y)\|}{\min\{k(x),k(y)\}}$	Resource allocation index (RA)	$S_{xy}=\sum_{z\in\Gamma(x)\cap\Gamma(y)}\frac{1}{k(z)}$

为了测试算法的准确性,将已知的边 E 分为两部分:训练集 E^T 和测试集 E^P.在计算预测分数值的时候只能使用训练集中的信息.显然,$E=E^T\cup E^P$,且 $E^T\cap E^P=\varnothing$.在此,将属于 U 但不属于 E 的边定义为不存在的边集 $\bar{E}$.衡量链路预测算法的指标包括 AUC[13]、准确率和召回率.他们对链路预测算法衡量的侧重点不同,AUC 是从整体上衡量算法的精确度,准确率只考虑分数值排在前 L 位的边是否预测准确,召回率则更多考虑的是预测准确的边占真实存在边的比例.

定义 1(AUC,曲线下面积). AUC 用来衡量链路预测算法的精准度,定义为

$$AUC=\frac{n'+0.5n''}{n} \tag{1}$$

分别从 E^P 和 $\bar{E}$ 中随机选取一条边比较预测分数值的大小,n 为比较的次数,n'为 E^P 中边的分数值大于 $\bar{E}$ 中边的分数的次数,n''为两分数值相等的次数.

定义 2(Precision,准确率). 准确率作为另一种指标用来衡量链路预测算法的精准度,定义为

$$Precision=\frac{m}{L} \tag{2}$$

从所有预测的边中选出预测分数值排在前 L 的边,m 为在前 L 条边中预测准确的边数.

定义 3(Recall,召回率). 召回率是预测准确的边数占实际存在但未发现的边数的比例,定义为

$$Recall=\frac{m}{|E^P|} \tag{3}$$

其中,$|E^P|$是测试集的边数.

定义 4(σ,抽取因子). 抽取因子是 L 和预测分数值非 0 的边个数的比率,定义为

$$\sigma = \frac{L}{nonzero(\overline{E} \cup E^P)} \tag{4}$$

其中,*nonzero* 函数返回的是边集中预测分数值不为 0 的边的个数.

AUC 可以理解为 E^P 中边的分数值比随机选择的 $\overline{E}$ 中边的分数值高的概率.显然,如果所有分数都是随机产生的,那么 AUC 为 0.5.因此,AUC>0.5 的程度衡量了算法在多大程度上比随机选择的方法精确[10].准确率可以理解为预测分数值排在前 L 的边实际存在的概率.召回率衡量的是查全率.

定义 5(*Adj*,邻接表). 图 $G=(V,E)$的邻接表由一个包含$|V|$个列表的数组 *Adj* 所组成,其中,每个列表对应于 V 中的一个顶点.对于每一个 $u\in V$,邻接表 $Adj[u]$包含所有满足条件$(u,v)\in E$ 的顶点 v[14].

定理 1. 对于无向图 $G=(V,E)$,在基于局部信息的相似性指标中,若边(u,v)的分数值不为 0,那么点 u,v 必然同时出现在 $Adj[w]$中,w 为 u,v 的共同邻居;否则,u,v 不同时出现在任意 $Adj[i]$中,$i\in V$.

证明:若边(u,v)的分数值不为 0,则点 u,v 必有共同邻居 w,即$(w,u)\in E$ 且$(w,v)\in E$,那么点 u,v 共同出现在$Adj[w]$中;否则,u,v 没有共同邻居,即不存在节点 i 使得$(i,u)\in E$ 且$(i,v)\in E$,所以 u,v 不共同出现在任意 $Adj[i]$中,$i\in V$.

定理 2. 对于链路预测 AUC 评价指标,若 E^P 中边分数值非 0 的概率为 p_1, $\overline{E}$ 中边分数值为 0 的概率为 p_2,则 $p_1p_2+\left(\frac{1-p_1}{2}\right)p_2 \leqslant AUC \leqslant p_1+\left(\frac{1-p_1}{2}\right)p_2$.

证明:每次分别从 E^P 和 $\overline{E}$ 中随机选取一条边,设为 e_1 和 e_2,比较它们的分数值:e_1 分数值非 0 且 e_2 分数值为 0 的概率为 p_1p_2,e_1 和 e_2 分数值同时为 0 的概率为$(1-p_1)p_2$,e_1 和 e_2 分数值同时不为 0 的概率为 $p_1(1-p_2)$.

在最坏情况下,E^P 中非 0 的分数值都小于 $\overline{E}$ 中非 0 的分数值,因此在公式(1)中,$\frac{n'}{n}=p_1p_2$且$\frac{n''}{n}=(1-p_1)p_2$,所以 $AUC=\frac{n'+0.5n''}{n} \geqslant p_1p_2+\left(\frac{1-p_1}{2}\right)p_2$;

在最好情况下,E^P 中非 0 的分数值都大于 $\overline{E}$ 中非 0 的分数值,因此在公式(1)中,$\frac{n'}{n}=p_1p_2+p_1(1-p_2)=p_1$且$\frac{n''}{n}=(1-p_1)p_2$,所以 $AUC=\frac{n'+0.5n''}{n} \leqslant p_1+\left(\frac{1-p_1}{2}\right)p_2$.

综上所述, $p_1p_2+\left(\frac{1-p_1}{2}\right)p_2 \leqslant AUC \leqslant p_1+\left(\frac{1-p_1}{2}\right)p_2$. □

2 并行链路预测算法

2.1 MapReduce并行计算模型

Hadoop 是一个用来处理海量数据的分布式系统,其核心组件是 MapReduce 和分布式文件系统(HDFS). MapReduce 组件是运行在大规模集群上的分布式数据处理模型,MapReduce 处理分为两个阶段:Map 阶段和 Reduce 阶段.每个阶段以 key-value 对作为输入和输出.Hadoop 将输入数据分割成小数据块,为每个数据块创建一个 Map 任务,数据将以 key-value 对的形式输入,经过 Map 函数处理后,MapReduce 框架将对 Map 输出按 key 值进行排序,再输入到 Reduce 任务中.所以,key 值相同的数据将被送往同一个 Reduce 任务,Reduce 任务再将 key 值相同的数据归并成一条数据进行处理.MapReduce 工作机制如图 1 所示[15].

MapReduce 模型并不允许随机读取图中的节点和边的信息,因此在图算法中,MapReduce 特别适合于那些以节点为中心、并且只用到节点局部信息的算法.基于节点局部信息的相似性指标只用到节点的一阶邻居信息,因此特别适合于 MapReduce 模型.

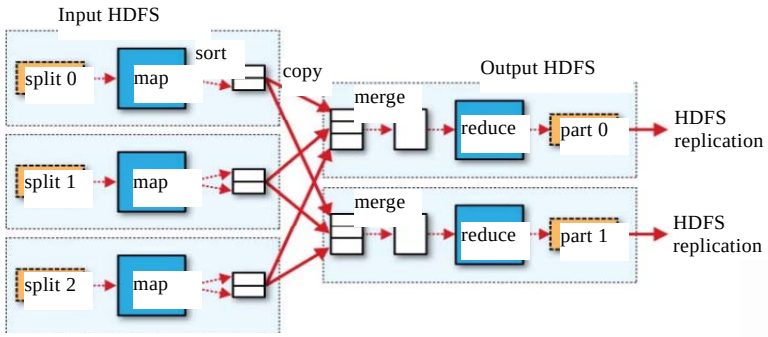


Fig.1 MapReduce data flow with multiple mapreduce tasks

图 1 具有多个 reduce 任务的 MapReduce 数据流

2.2 并行算法实现及其分析

本文在 MapReduce 计算模型上实现了基于节点局部信息的 9 种相似性指标(除了 PA),因为这 9 种指标都用到了节点的共同邻居信息,因此在并行算法实现上具有一致性.并行链路预测算法步骤如下所示,并给出了核心步骤 4 的 MapReduce 伪代码.

并行链路预测算法.

输入:$G(V,E)$的边集 E(文件 A);

输出:AUC 值.

1. 得到网络的节点数 N,将网络的边集 E 分割为训练集 E^T(文件 B)和测试集 E^P(文件 C);
2. 得到 E^T 和 E^P 中边的个数,得到 E^T 中所有节点的度;
3. 将 E^T 中的点对用邻接表表示;
4. 对于 E^T 的邻接表,根据相似性指标得到两两点对的预测分数值(文件 D);
5. 得到 E^P 中所有边的预测分数值(文件 D∩文件 C)和 $\overline{E}$ 中所有边的预测分数值(文件 D~文件 A);
6. 分别从 E^P 和 $\overline{E}$ 中随机选取一条边进行比较,重复 n 次,根据公式(1)计算出 AUC 值.

步骤 4. 根据相似性指标得到两两点对的预测分数值

Mapper

输入:图 $G(V,E)$的邻接表;

输出:图 $G(V,E)$两两点对之间的共同邻居.

```
1. 处理数据行 Key: x Value: Γ[1~n]                    //Γ[1~n]为点 x 的 n 个邻居
2. FOR i←1 TO n–1                                     //从Γ[n]中选取不同的两个邻居节点
3.     FOR j←i+1 TO n
4.         IF ID(Γ[i])>ID(Γ[j])
5.             设置 Key: (Γ[j],Γ[i]) Value: x          //将 ID 小的节点放前面,避免重复计算
6.         ELSE 设置 Key: (Γ[i],Γ[j]) Value: x         //点 x 是点Γ[i]和Γ[j]的共同邻居
7.         输出(Key,Value)
```

Reducer

输入:图 $G(V,E)$两两点对之间的共同邻居;

输出:图 $G(V,E)$两两点对的预测分数值.

```
1. 处理数据行 Key: (Γ[i],Γ[j]) Value: Iterator(x1,x2,x3,…,xm)   //x[m]是点Γ[i]和点Γ[j]的共同邻居
2. 设置 Key: (Γ[i],Γ[j])
3. 设置Γ[i],Γ[j]的共同邻居数为 m
```

4. 根据不同的指标计算预测分数值,若计算 LHN-I 指标,则设置 Value: m/(degree($B[i]$)·degree($B[j]$))

5. 输出(Key,Value) //输出点对的预测分数值

从表 1 可以看出,基于局部信息的链接预测算法中,所有分数值的计算都与共同邻居有关(除了 PA 指标),所以,如果两个节点没有共同邻居,那么预测分数值肯定为 0.定理 1 说明:如果节点 a 和 b 有共同邻居 c,那么 a 和 b 肯定同时存在于 $Adj[c]$中;反之,如果节点 a 和 b 没有共同邻居,那么 a 和 b 将不会同时出现在任何一个单链表中,所以只要计算单链表中两两点对的预测分数值,而无需对所有点对进行计算.如果网络中节点的平均度为 $\langle k\rangle$,$\langle k\rangle=2M/N$,那么对于每个节点的单链表,所需计算的点对数目为$\langle k\rangle(\langle k\rangle-1)/2$,所以整个邻接表总共计算的点对数目是$\langle k\rangle(\langle k\rangle-1)N/2$.因此,基于局部信息的链接预测算法的时间复杂度为 $O(N\langle k\rangle^2)$,或为 $O(M\langle k\rangle)$;空间复杂度为 $O(N\langle k\rangle)$,或为 $O(M)$.

Barábasi 和 Albert 提出了真实网络中节点度的指数定律,即节点度数为 k 的概率和 k 成反比[16],网络中大部分节点的度都很小.而现实世界中的大型网络绝大部分都是稀疏图,平均度$\langle k\rangle$是一个很小的常数,所以对于稀疏的大型网络来说,算法的时间复杂度为 $O(N)$.而对于并行链路预测算法,数据会分发到多个处理单元上,所以时间复杂度为 $O(N/U)$,U 为处理单元的个数.

2.3 计算过程示例

图 2 给出了一个示例图 $G(V,E)$和 MapReduce 的输入/输出过程,采用 CN 指标计算两两点对的预测分数值.首先由 map 读入示例图的邻接表,对于每个 $Adj[v]$,map 每次获取其中的一个节点对(u,w),记录他们的共同邻居 v 一次;然后在 reduce 中统计节点对(u,w)的共同邻居总数,即在 CN 指标下的预测分数值.Reduce 的输出为点对的预测分数值,所有剩下的点对预测分数值都为 0.

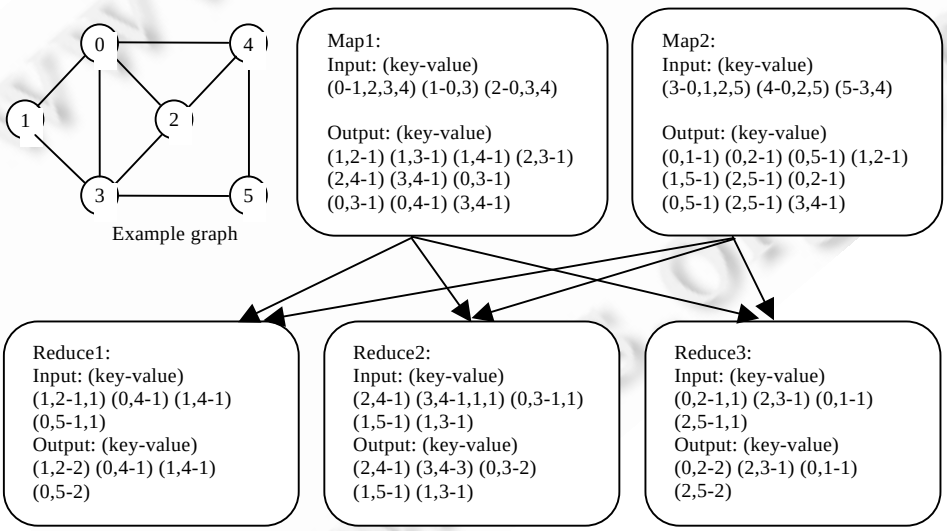


Fig.2 I/O of Map and Reduce tasks

图 2 Map 和 Reduce 任务的输入/输出

3 实验结果及其分析

3.1 数据集介绍

数据集包括 3 个典型小型网络和 10 个大型复杂网络.3 个典型小型网络包括科学家合作网络(NS)[17]、美国电力网络(Grid)[18]以及美国航空网络(USAir)[19],它们的统计性质见表 2[6].其中,N,M 分别表示网络的节点数和边数;Nc 为网络的最大联通集团,例如,332/1 表示 USAir 网络中有 1 个连通集团,最大连通集团包含 332 个节

点;*e* 为网络的效率;*ACC* 为网络的平均聚集系数;*r* 为同配系数;⟨*k*⟩为网络的节点平均度.NS 网络初始有 1 589 个节点,但本文并不考虑其中的 128 个孤立节点.

Table 2 Basic topological features of three example networks

表 2 3 个实验网络的拓扑性质

Datasets	*N*	*M*	*Nc*	*e*	*ACC*	*r*	⟨*k*⟩
NS	1 461	2 742	379/268	0.016	0.878	0.462	3.754
Grid	4 941	6 594	4941/1	0.063	0.107	0.003	2.669
USAir	332	2 126	332/1	0.406	0.794	−0.208	12.807

超大型网络数据来自 Stanford 大型网络数据集[20],本文从中选取了 9 种不同类型的网络,共 10 个数据集.选取的数据集名称和建网规则见表 3,数据集中包含了社交网络、Internet 网络和引文网络等.10 个大型网络的统计性质见表 4.其中,*M* 为转化为无向图后网络的边数,*N*(SCC)和 *M*(SCC)为最大强连通分量的点和边数占整个网络的比例,*D* 为网络直径,*T* 为图中三角形的个数占长度为 2 的路径个数的比例.数据集中最大的网络 cit-Patents 包含百万个节点、千万条边,最小的网络也有数万个节点和边,不同网络的拓扑性质也有较大差异.

Table 3 Ten large-scale example networks

表 3 10 个大规模实验网络

Network type	Network meaning	Datasets
Social networks	Nodes represent people and edges interactions	soc-Epinions1
Web graphs	Nodes represent Web pages and edges are hyperlinks	Web-Google, Web-NotreDame
Collaboration networks	Nodes represent scientists and edges them co-authoring a paper	ca-AstroPh
Citation networks	Nodes represent papers and edges represent citations	cit-Patents
Internet peer-to-peer networks	Nodes represent computers and edges communication	p2p-Gnutella24
Communication networks	Node represent emails and edges communication	email-Enron
Amazon networks	Nodes represent products and edges link co-purchased products	amazon0601
Road networks	Nodes represent intersections and edges road connecting them	roadNet-CA
Signed networks	Who-Trust-Whom network with edges represent trustiness	soc-sign-epinions

Table 4 Basic topological features of ten large-scale example networks

表 4 10 个大规模实验网络的拓扑性质

ID	Datasets	*N*	*M*	⟨*k*⟩	*N*(SCC)	*M*(SCC)	*D*	*T*	*ACC*	Average AUC
G1	P2P-Gnutella24	26 518	65 369	4.93	0.240	0.351	10	0.00410	0.0094	0.509
G2	email-Enron	36 692	183 831	10.02	0.918	0.984	12	0.08531	0.4970	0.922
G3	ca-AstroPh	18 772	198 110	21.11	0.954	0.995	14	0.31800	0.6306	0.972
G4	soc-Epinions1	75 879	405 740	10.69	0.425	0.872	13	0.06568	0.2283	0.836
G5	soc-sign-epinions	131 828	711 783	10.80	0.314	0.825	14	0.08085	0.2424	0.844
G6	Web-NotreDame	325 729	1 117 563	6.86	0.166	0.204	46	0.08767	0.4540	0.814
G7	amazon0601	403 394	2 443 408	12.11	0.980	0.975	21	0.16560	0.4179	0.889
G8	roadNet-CA	1 965 206	2 766 607	2.82	0.996	0.998	850	0.06039	0.0464	0.531
G9	Web-Google	875 713	4 322 051	9.87	0.497	0.670	22	0.05523	0.6047	0.901
G10	cit-Patents	3 774 768	16 518 948	8.75	0.000	0.000	22	0.06714	0.0919	0.663

3.2 并行算法有效性验证

验证数据选取前面提到的科学家合作网络、美国电力网络以及美国航空网络.很多关于链路预测的文献都针对这 3 个数据集做了实验,因此,将本文并行链路预测算法的结果和它们进行对比可以验证并行算法的正确性.对于 3 种网络,并行链路预测算法的 AUC 计算结果见表 5.对于每一种网络,第 1 列数据来自文献[6];第 2 列数据由本文并行链路预测算法得出;第 3 列数据来自文献[21],它并没有对 RA 指标进行计算.可以看出,MapReduce 并行链路预测算法得出来的 AUC 值和另外两篇文献中的 AUC 值总体趋势保持一致,AUC 平均误差为 0.035,在可接受的范围之内.

抽取因子对准确率和召回率的影响如图 3 和图 4 所示.总体来看,随着抽取因子的增加,准确率逐渐降低,召回率逐渐升高.对于 NS 网络,当抽取因子<0.1 时,准确率大于 80%;而对于 USAir 网络,当抽取因子=0.05 时,准确率已经不到 50%,准确率随抽取因子的增长下降得很快,所以只有极少数预测分数值最高的边才能认为是预测

准确的边.当抽取因子=1 时,召回率并没有达到 100%,说明根据相似性指标,一部分真实存在的边分数值为 0;对于 NS 网络有近 25%的真实边分数值为 0,这说明基于局部信息的相似性指标并不能很好地体现出 NS 网络的拓扑结构,因此还有很大的提升空间.在相同的抽取因子下,USAir 网络准确率要低于 NS 网络,召回率要高于 NS 网络.对于 Grid 网络,AUC 的平均值只有 0.558,效果和随机算法相近,准确率最高才达到 7.67%.

Table 5 Comparison of AUC results in three papers

表 5 3 篇论文中的 AUC 计算结果比较

Indices	NS			Grid			USAir		
CN	0.933	0.890	0.9370	0.590	0.564	0.5881	0.937	0.948	0.9345
Salton	0.911	0.868	0.9371	0.585	0.561	0.5880	0.898	0.893	0.9075
Jaccard	0.933	0.883	0.9371	0.590	0.558	0.5880	0.901	0.895	0.8963
Sorenson	0.933	0.884	0.9371	0.290	0.555	0.5880	0.902	0.888	0.8963
HPI	0.911	0.868	0.9370	0.585	0.561	0.5880	0.857	0.853	0.8676
HDI	0.933	0.881	0.9370	0.590	0.557	0.5880	0.895	0.900	0.8896
LHN-I	0.911	0.865	0.9367	0.585	0.559	0.5880	0.758	0.769	0.7613
AA	0.932	0.881	0.9373	0.590	0.554	0.5880	0.925	0.923	0.9462
RA	0.933	0.883		0.590	0.555		0.955	0.954	
Average AUC	0.926	0.878	0.937	0.555	0.558	0.588	0.892	0.891	0.887

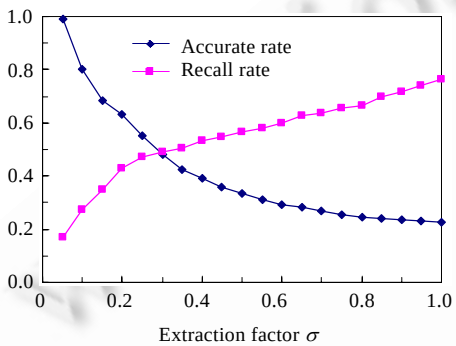


Fig.3 σ versus Precision and recall of NS network

图 3 NS 网络准确率和召回率随抽取因子σ的关系

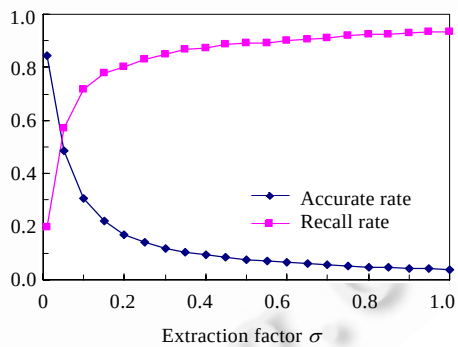


Fig.4 σ versus Precision and recall of USAir network

图 4 USAir 网络准确率和召回率随抽取因子σ的关系

3.3 大规模数据实验

使用并行链路预测算法在 10 个大型网络上进行实验.实验环境为两个 Hadoop 集群,每个集群由多个服务器组成,集群 1 的机器节点数为 6 个,集群 2 的节点数为 20 个.集群上的 Hadoop 版本为 0.20.1,每个机器节点的配置为一个 4 核 CPU,CPU 型号为 Intel(R) Xeon(R) CPU E5504 @ 2.00GHz,内存大小为 4GB,节点之间相互拷贝数据的测试速度为 11.2MB/s~12MB/s.链路预测结果见表 4,在 10 个大型网络中,9 种节点相似性指标得出的 AUC 值非常接近,故只列出每个网络的平均 AUC 值.其中,ca-AstroPh 网络的平均 AUC 值最高,达到 0.972,P2P-Gnutella24 网络的平均 AUC 最低,只有 0.509,仅和随机算法得出的 AUC 值相近.对于 10 个大型网络,AUC 值在不同相似性指标下的标准差平均值只有 0.004,而 3 个典型网络的 AUC 值标准差平均值为 0.02.所以,不同相似性指标对大型网络 AUC 值的影响要小于对小型网络的影响.

并行链路预测算法在集群 1 上的运行时间如图 5 所示,图中横坐标为数据集的 ID,纵坐标为运行的时间,M*i*R*i* 为设置的 Map 和 Reduce 任务的个数.可以看出,随着 Map 和 Reduce 任务数的增加,链路预测算法的运行时间不断下降.通过设置 Map 和 Reduce 任务的个数为 1 来模拟单机的运行时间,从图中可以看出,并行链路预测算法和传统单机算法相比有巨大的优势,并且优势随着并行规模的增加而更加明显.图 6 为 5 个较大数据集在两个集群上的运行结果,在集群 1 和集群 2 上运行时的配置分别为 M6R6 和 M20R20.从图中可以看出,集群 2 的运行时间比集群 1 大幅下降,并且随着数据规模的增大,运行时间减少的比例越多,分别为 34.9%,38.2%,

51.3%和 59.0%.图 5 和图 6 体现了并行链路预测算法相对于传统链路预测算法的巨大优势.

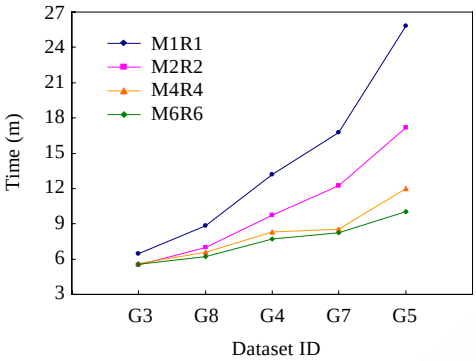


Fig.5 Running time comparing of different M*i*R*i*

图 5 不同 M*i*R*i* 运行时间对比

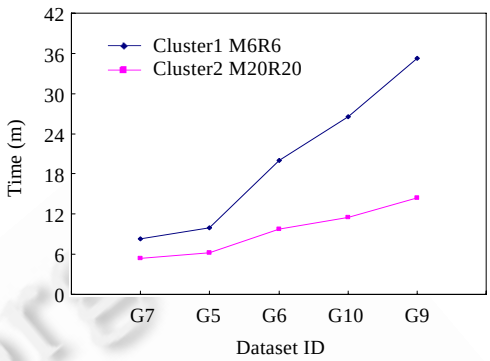


Fig.6 Running time comparing of two clusters

图 6 两个 Hadoop 集群运行时间对比

4 链路预测 AUC 评价指标分析

4.1 AUC上下界分析

对于一个待链路预测的网络,如果知道 E^P 中边的预测分数值非 0 的概率 p_1 和 $\overline{E}$ 中边的预测分数值为 0 的概率 p_2,则可以计算出该网络 AUC 值的上下界,如定理 2 所示: $p_1p_2+\left(\frac{1-p_1}{2}\right)p_2 \leqslant AUC \leqslant p_1+\left(\frac{1-p_1}{2}\right)p_2$.因此,AUC 的上下界可以用来估算 AUC 的真实值,AUC 上下界之间的差值为 $p_1(1-p_2)$,如果这个差值很小,那么 AUC 就能被准确估算.表 6 给出了 3 个典型网络的 E^P 和 $\overline{E}$ 中边的预测分数值是否为 0 的比例、AUC 上下界和 AUC 的真实值.在 NS 网络和 USAir 网络中,E^P 中边的分数值非 0 的比例要远高于分数值为 0 的比例,而在 $\overline{E}$ 中正好相反.对于 Grid 网络来说,E^P 中分数值非 0 的边只占到了总数 11.77%,这是导致 Grid 网络 AUC 平均值只有 0.558 的原因.

Table 6 Upper and lower bounds of AUC in three example networks

表 6 3 个典型网络的 AUC 值上下界

Datasets	Edge set	*M*	Non-Zero (%)	Zero (%)	Upper and lower bounds of AUC	Average AUC
NS	E^P	794	75.82	24.18	[0.8773,0.8789]	0.878
	$\overline{E}$	1 063 788	0.20	99.8		
Grid	E^P	1 980	11.77	88.23	[0.55846,0.55854]	0.558
	$\overline{E}$	12 197 676	0.07	99.93		
USAir	E^P	618	89.97	10.03	[0.7122,0.9373]	0.891
	$\overline{E}$	52 820	25.02	74.98		

在不考虑 E^P 和 $\overline{E}$ 中边预测分数值的大小而仅考虑分数值是否为 0 的情况下,AUC 上下界的中值和实际 AUC 值很接近.对于这 3 个网络,AUC 上下界的中值和实际 AUC 值的比率分别为 99.78%,100%和 92.56%.这说明 AUC 公式虽然比较的是分数值的大小,但是 AUC 值却取决于 E^P 和 $\overline{E}$ 中边预测分数值是否为 0 的比例.而准确率的计算只包含预测分数值排名靠前的 L 条边,这 L 条边中绝大多数的预测分数值都不为 0,所以准确率侧重的是预测分数值的大小.AUC 和准确率计算上的侧重点不同造成了 AUC 和准确率结果上的差异.

4.2 网络平均聚集系数对AUC的影响

为了更好地说明网络拓扑结构对 AUC 的影响,本文采用 NeSVA 软件[22]对链路预测进行可视化展示.对于每个网络,选取 AUC 值最高的相似性指标来进行可视化展示;对于 NS 网络采用 CN 指标;而对于 USAir 网络采用 RA 指标.由于 Grid 网络预测的准确率很低,故在此不做可视化展示.原始 NS 网络一共有 2 742 条边,图 7 中

较深颜色的边为 NS 网络的训练集,训练集一共有 1 914 条边,剩下的 828 条边作为测试集.在抽取因子为 0.05 时,L 为 141,较浅颜色的边即为选取的预测分数值在前 L 的边.此时,链路预测的准确率达到 99.29%,召回率为 16.91%.原始 USAir 网络一共有 2 126 条边,图 8 中较深颜色的边为 USAir 网络的训练集,训练集一共有 1 510 条边,剩下的 616 条边作为测试集.在抽取因子为 0.05 时,L 为 726,此时的准确率达到 48.48%,召回率为 57.14%.从图上也可以看出,基于局部信息的链路预测的特点,网络中聚集系数高的子图结构产生预测边的可能性要极大地高于聚集系数低的子图结构.

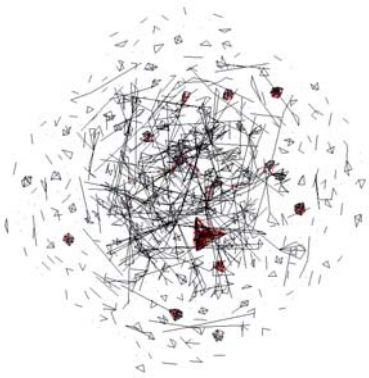


Fig.7 NS network after link prediction

图 7 链路预测后的 NS 网络

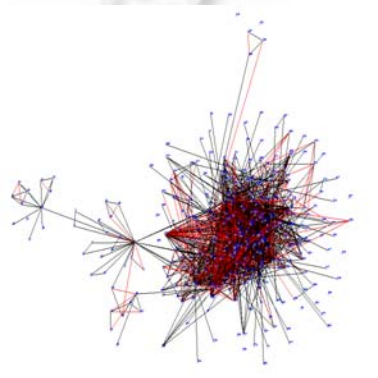


Fig.8 USAir network after link prediction

图 8 链路预测后的 USAir 网络

网络的拓扑性质和 AUC 值之间的 Pearson 相关系数如图 9 所示,相关系数的计算由表 4 得出.其中,网络直径 D 和 AUC 是负相关,网络的平均度和平均聚集系数对 AUC 值的影响较大,相关系数分别为 0.78 和 0.90,故网络平均聚集系数是影响 AUC 值最重要的因素.由表 4 可以得到 10 个大型网络的平均聚集系数和 AUC 平均值之间的关系.网络平均聚集系数对 AUC 的影响如图 10 所示,P2P-Gnutella24 网络的 ACC 为 0.009 4,AUC 平均值为 0.509,与随机算法产生的 AUC 值非常接近.随着 ACC 增加到 0.2 以上,AUC 值高于 0.8,有大幅度的提升.网络平均聚集系数越高,代表着网络中三角形占三元组的比例越高.如果一个节点对和其他节点形成的三角形数目越多,那么该节点对的共同邻居数也越多,节点对的链接预测分数值就越大.总体上来讲,AUC 值随着网络平均聚集系数的增加而升高.由此可知,若网络的聚集系数很小,则不适合用基于节点局部信息的相似性指标进行链路预测,而应该采用基于路径或随机游走的相似性指标.从并行链路预测算法的运行时间来看,聚集系数越高,导致链路预测分数值非零的节点对越多.故在其他网络拓扑性质相同的情况下,聚集系数越高的网络运行时间越长.

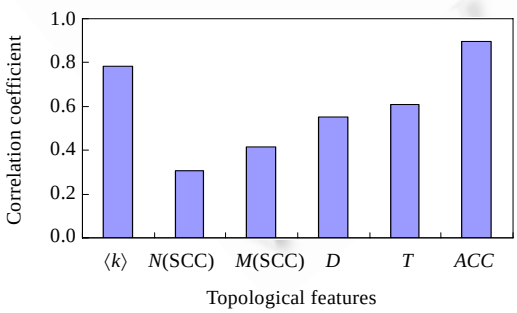


Fig.9 Correlation between topological features and AUC

图 9 网络拓扑性质和 AUC 值之间的相关系数

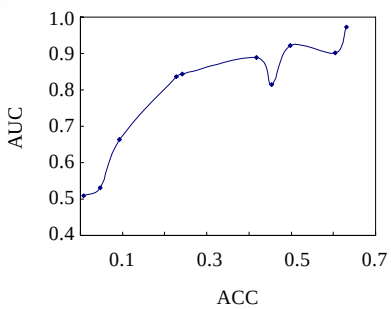
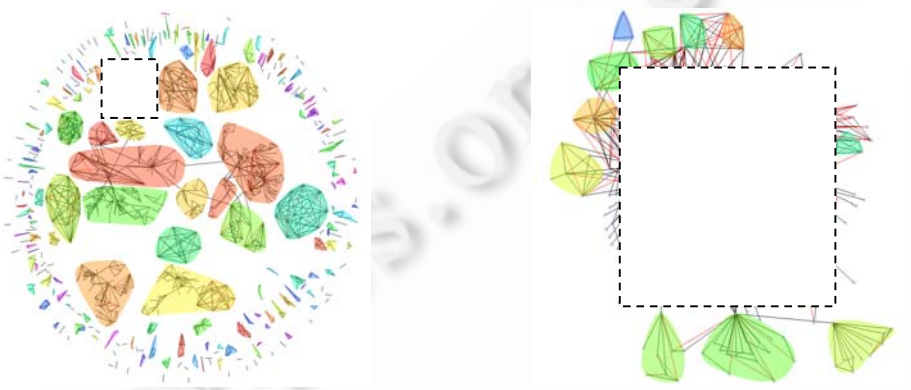


Fig.10 AUC under different value of ACC

图 10 网络平均聚集系数对 AUC 的影响

对于 NS 网络和 USAir 网络,采用 GN 算法[23]对网络进行社团划分,使用 NeSVA 软件展示后结果如图 11 所示.图中每一个社团使用不同的颜色进行表示,图中较深颜色的边代表训练集,链路预测算法预测的边为较浅颜色,此时的抽取因子为 0.05.图 11(a)中,NS 网络中有 268 个连通集团,所以 NS 网络的社团数目非常的多.设虚线方框中的社团为 c_1,可以看到,预测的边大部分都在 c_1 之内.图 11(b)中,USAir 网络只有一个联通集团,在虚线方框中存在一个非常大的社团,定义为 c_2.同样,预测的边大部分都在 c_2 之内.定义社团的聚集系数为社团内所有节点聚集系数的平均值,可以看出,c_1 和 c_2 相比于其他社团来说,其中的边更加密集而且社团的聚集系数较高.从此也可以看出,网络中聚集系数高的子图结构产生预测边的可能性要极大地高于聚集系数低的子图结构.



(a) NS 网络 (b) USAir 网络

Fig.11 Link prediction results and communities in networks

图 11 社团划分后的网络和网络链路预测的结果

5 结论及展望

本文提出了基于局部信息的并行链路预测算法,从图的邻接表得到所有预测分数值非 0 的节点对,进而计算网络的 AUC 值.并行链路预测算法采用 MapReduce 计算模型,时间复杂度为 $O(N/U)$.实验结果表明,并行链路预测算法相对于传统链路预测算法具有更高的效率,在大数据集下更为显著.本文还提出并证明了 AUC 评价指标的上下界,通过分析得出 AUC 上下界的中值与 AUC 实际值很接近,并得出 AUC 评价指标侧重于预测分数值是否为 0 而不是分数值的大小的结论.此外,本文还讨论了网络拓扑结构对 AUC 评价指标的影响.实验结果表明,网络平均聚集系数对 AUC 值的影响最大,并且 AUC 值随着网络平均聚集系数的增加而提高.

在下一步的工作中,将针对基于路径的相似性指标和基于随机游走的相似性指标的并行化进行研究.针对 MapReduce 模型在处理图数据时的不足,可以采用更适合图处理的 BSP 或 MPI 并行计算模型,并对它们的大规模并行化进行研究.针对 AUC 评价指标的不足,将进一步研究 AUC 的改进方法.同时,如何针对不同拓扑结构的网络采用不同的相似性指标进行链路预测以达到最好的预测效果,也是值得研究的问题.

References:

[1] Getoor L, Diehl CP. Link mining: A survey. ACM SIGKDD Explorations Newsletter, 2005,7(2):3−12. [doi: 10.1145/1117454.1117456]

[2] Dorogovtsev SN, Mendes JFF. Evolution of networks. Advances in Physics, 2002,51(4):1079−1187. [doi: 10.1080/ 000187301101 12519]

[3] Lin D. An information-theoretic definition of similarity. In: Proc. of the 15th Int'l Conf. on Machine Learning. San Francisco: Morgan Kaufman Publishers, 1998. 296−304. [doi: 10.1.1.55.1832]

[4] Madadhain JO, Hutchins J, Smyth P. Prediction and ranking algorithms for event-based network data. ACM SIGKDD Explorations Newsletter, 2005,7(2):23−30. [doi: 10.1145/1117454.1117458]

[5] Popescul A, Ungar LH. Statistical relational learning for link prediction. In: Proc. of the IJCAI-03 Workshop on Learning Statistical Models from Relational Data. New York: ACM Press, 2003. 81−87. [doi: 10.1.1.57.6234]

[6] Zhou T, Lü LY, Zhang YC. Predicting missing links via local information. The European Physical Journal B, 2009,71(4):623−630. [doi: 10.1140/epjb/e2009-00335-8]

[7] Lü LY, Jin CH, Zhou T. Similarity index based on local paths for link prediction of complex networks. Physical Review E, 2009, 80(4):046122. [doi: 10.1103/PhysRevE.80.046122]

[8] Katz L. A new status index derived from sociometric analysis. Psychometrika, 1953,18(1):39−43. [doi: 10.1007/BF02289026]

[9] Leicht EA, Holme P, Newman MEJ. Vertex similarity in networks. Physical Review E, 2006,73(2):026120. [doi: 10.1103/PhysRevE.73.026120]

[10] Lü LY, Zhou T. Link prediction in complex networks: A survey. Physica A, 2011,390(6):1150−1170. [doi: 10.1016/j.physa.2010.11.027]

[11] Lin J, Schatz M. Design patterns for efficient graph algorithms in MapReduce. In: Proc. of the 8th Workshop on Mining and Learning with Graphs. New York: ACM Press, 2010. 78−85. [doi: 10.1145/1830252.1830263]

[12] Kang U, Tsourakakis CE, Faloutsos C. PEGASUS: Mining peta-scale graphs. Knowledge and Information Systems, 2010,27(2): 303−325. [doi: 10.1007/s10115-010-0305-0]

[13] Hanely JA, McNeil BJ. The meaning and use of the area under a receiver operating characteristic (ROC) curve. Radiology, 1982, 143(1):29−36.

[14] Cormen TH, Leiserson CE, Rivest RL, Stein C. Introduction to Algorithms. 3rd ed., The MIT Press, 2001. 322−323.

[15] White T. Hadoop: The Definitive Guide. O'Reilly Media, 2009. 29−30.

[16] Barabási AL, Albert R. Emergence of scaling in random networks. Science, 1999,286(5439):509−512. [doi: 10.1126/science.286.5439.509]

[17] Newman MEJ. Finding community structure in networks using the eigenvectors of matrices. Physical Review E, 2006,74(3): 036104. [doi: 10.1103/PhysRevE.74.03.6104]

[18] Watts DJ, Strogatz SH. Collective dynamics of 'small-world' networks. Nature, 1998,393(6684):440−442. [doi: 10.1038/30918]

[19] Batagelj V, Mrvar A. Pajek datasets. http://vlado.fmf.uni-lj.si/pub/networks/data/

[20] Leskovec J. Stanford large network dataset collection. http://snap.stanford.edu/data/

[21] Dong YX, Ke Q, Wang B, Wu B. Link prediction based on local information. In: Proc. of the 2011 Int'l Conf. on Advances in Social Networks Analysis and Mining. Kaohsiung: IEEE Press, 2011. 382−386. [doi: 10.1109/ASONAM.2011.43]

[22] Ye Q, Wu B, Suo LJ, Zhu T, Han C, Wang B. TeleComVis: Exploring temporal communities in telecom networks. In: Proc. of the European Conf. on Machine Learning and Knowledge Discovery in Databases. Berlin: Springer-Verlag, 2009. 755−758. [doi: 10.1007/978-3-642-04174-7_57]

[23] Girvan M, Newman MEJ. Community structure in social and biological networks. Proc. of the National Academy of Science, 2002, 99(12):7821−7826. [doi: 10.1073/pnas.122653799]

饶君(1989－),男,江西南昌人,硕士,主要研究领域为图数据挖掘,并行计算.

东昱晓(1987－),男,硕士,主要研究领域为图数据挖掘,并行计算.

吴斌(1969－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为智能信息处理,图数据挖掘,云计算.