

## P2P 网络中时变数据的近似聚集方法\*

程思瑶, 姜守旭<sup>+</sup>, 李建中

(哈尔滨工业大学 计算机科学与技术学院, 黑龙江 哈尔滨 150001)

### Approximate Aggregation of Time-Varying Data in P2P Networks

CHENG Si-Yao, JIANG Shou-Xu<sup>+</sup>, LI Jian-Zhong

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

+ Corresponding author: E-mail: jsx@hit.edu.cn

Cheng SY, Jiang SX, Li JZ. Approximate aggregation of time-varying data in P2P networks. *Journal of Software*, 2009,20(7):1800–1811. <http://www.jos.org.cn/1000-9825/3294.htm>

**Abstract:** With the wide application of peer-to-peer (P2P) technologies in many fields such as E-commerce, it is increasingly necessary to do aggregation queries in P2P networks. However, due to the large scale and decentralization of P2P networks it is rather difficult to do this kind of operation. Aggregation queries will become even more difficult in case that the data in P2P networks are time-varying which is often occurs in practice. The existing aggregation methods for data in P2P networks all assume that the data are time-invariant. If these methods are directly applied to P2P networks with time-varying data, some problems will arise because the data used in aggregation processing would have changed owing to the long time of aggregation. So, this paper proposes an approximate aggregation method for time-varying data in P2P networks based on uniform sampling. The theoretical analysis and experimental results show that this aggregation method outperforms the existing methods and can effectively be applied to P2P networks with time-varying data.

**Key words:** P2P network; time-varying data; uniform sampling; query processing; approximate aggregation

**摘要:** 随着 P2P 技术在电子商务等领域的广泛应用,对分布在 P2P 网络中的数据进行聚集操作的需求越来越迫切.但是,由于 P2P 网络的大规模及分散性,这种聚集操作的实现颇具挑战性,而且在很多应用中,P2P 网络中的数据往往是随时间变化的,这进一步增加了聚集操作的难度.现有 P2P 网络中的聚集算法均假定网络中的数据是非时变的,如果将其直接应用在存在时变数据的 P2P 网络中,则会因为其聚集时间过长而导致聚集过程中数据已经发生变化的问题.为此,提出了一种 P2P 网络中基于均衡采样的时变数据近似聚集算法,理论分析和实验结果表明,该聚集算法在处理时变数据时优于已有的算法,可以有效地应用于存在时变数据的 P2P 网络中.

**关键词:** P2P 网络;时变数据;均衡采样;查询处理;近似聚集

中图法分类号: TP392 文献标识码: A

\* Supported by the National Basic Research Program of China under Grant No.2006CB303000 (国家重点基础研究发展计划(973)); the National Natural Science Foundation of China under Grant Nos. 60533110, 60703012, 60773068, 60773063 (国家自然科学基金); the Program for New Century Excellent Talents in University of China under Grant No.NCET-05-0333 (新世纪优秀人才支持计划); the Heilongjiang Province FUND for Young Scholars of China under Grant No.QC06C033 (黑龙江省青年科技专项资金)

Received 2007-06-30; Accepted 2008-02-27

在过去的几年中,Peer-to-Peer(P2P)技术得到了飞速发展,尤其是像 BitTorrent, Freenet 和 KaZaA 这样的 P2P 文件共享系统更是获得了巨大的成功.目前,P2P 技术正在逐步与数据库技术、信息检索技术、语义信息集成技术相结合,在越来越多的领域得到广泛应用.P2P 系统的开放结构使其具有如下特性:适应性、自组织性、负载均衡性、容错性、可扩展性,并能通过大量副本实现高可用性.

随着 P2P 数据管理技术研究的不断发展,越来越多的 P2P 应用将产生大量的时变数据.所谓时变数据是指随着时间的推移不断发生变化的数据.如在 P2P 电子商务系统中,有关节点的交易额、交易次数、商品的售价或需求量、节点的可信度等数据就是时变数据;在 P2P 环境监测系统中,每个节点所保存的某个地区的温度、湿度等环境指标也是时变数据.此外,P2P 电信服务网络等应用系统也会产生大量与通信量有关的时变数据.

在上述存在时变数据的 P2P 数据管理应用中,经常需要对时变数据进行聚集运算.如在 P2P 电子商务系统中,某时间段内节点的平均可信度可以辅助用户确定交易伙伴的相对可信程度,某种商品的平均价格及总需求量可以辅助用户确定商品的售价,系统中的平均交易量和平均交易次数则可以辅助调整信任机制中信任度的收敛速度等;在 P2P 电信服务网络中,某时间段内系统中的平均流量同样可以辅助进行流量控制.

但是,由于 P2P 系统的大规模、动态性和分散性,对 P2P 网络中的时变数据执行聚集操作极具挑战性.已有的 P2P 网络中的聚集算法均假定网络中的数据是静态的<sup>[1-5]</sup>,即认为网络中的数据在整个聚集查询处理过程中是不变的.但是对存在时变数据的大规模 P2P 网络来说,网络中的数据很可能在查询请求转发及数据收集过程中已经发生了变化.因此,在利用这些算法计算时变数据的聚集值时,可能会因为使用了时间上失效的数据而产生较大的误差.所以,已有的 P2P 网络中的聚集算法不适合计算时变数据的聚集值.

为此,本文对 P2P 网络中时变数据的聚集方法展开了研究.由于 P2P 网络具有大规模、动态性等特点,所以不可能通过获取网络中所有时变数据的数值来计算其精确的聚集值,而只能采用采样的方法对其进行估计.对于 P2P 网络中均匀分布的数据来说,只需对节点进行足够多的均衡采样即可获得较为精确的聚集值.但对于分布不均匀的数据则需采用更为复杂的数据采样技术<sup>[6,7]</sup>,对此我们将在后续工作中再加以深入研究.不过,由于 P2P 网络中存在着大量均匀分布的时变数据,如某时间段内节点的可信度、节点的在线时间以及节点的交易统计信息等,所以,本文针对均匀分布的时变数据(如无特殊说明,本文所说的时变数据均指均匀分布的时变数据),提出了一种 P2P 网络中具有一定误差保证的、基于均衡采样的时变数据近似聚集算法——AUS(aggregation based on uniform sampling)算法,理论分析和实验结果表明,该聚集算法可以有效地应用于存在时变数据的 P2P 网络中.

本文第 1 节对已有的 P2P 网络聚集算法的研究成果进行分析.第 2 节给出 AUS 算法的基本思想.第 3 节给出 AUS 算法的数学原理.第 4 节讨论 AUS 算法的实现并对其性能进行分析.第 5 节通过实验验证 AUS 算法的有效性.第 6 节总结全文并对未来工作进行展望.

## 1 相关工作

早期 P2P 网络中的聚集查询算法主要着眼于如何获取网络的真实聚集值.这一阶段的算法主要分为如下 3 类:通过生成树进行聚集<sup>[1,2]</sup>、通过 DHTs 进行聚集<sup>[3]</sup>、利用 Gossip 协议进行聚集<sup>[4]</sup>.其中,前两类算法的动态适应能力较差,即一旦网络拓扑结构发生变化,则需重新构建聚集覆盖网络才能完成聚集计算,故这两类算法不能很好地适应动态性很强的 P2P 网络.利用 Gossip 协议进行聚集的算法虽然克服了前两类算法对网络拓扑过分依赖的缺点,并具有很好的动态适应性,但是由于这类算法与前两类算法一样,都致力于获取网络中的真实聚集值,所以需要遍历网络中的所有节点,这必将造成网络资源的巨大浪费.同时,这些算法也存在着可扩展性差、聚集处理时间过长等缺点.另外,在计算时变数据的聚集值时,由于聚集处理时间比较长,网络中的数据可能在聚集计算过程中已经发生了变化,所以,以上 3 类算法有可能使用时间上失效的数据.

针对早期 P2P 网络中聚集查询算法对网络资源消耗大及计算时间长等缺点,Arai 等人<sup>[5]</sup>借鉴了大规模数据库中有关近似聚集处理(approximate query processing,简称 AQP)的研究成果<sup>[8-11]</sup>,提出了一种 P2P 网络中基于随机采样的近似聚集算法(本文称其为 Arai 算法).这种算法的主要思想是通过在网络中的数据进行两次采样来

估计全局聚集值.第 1 次采样首先在网络中选取少量样本,并将样本方差与误差阈值相比较以确定第 2 次采样的样本容量,然后进行 2 次采样并将两次采样所获得的数据求均值,从而得到具有一定误差保证的全局聚集值.可见 Arai 算法通过采样的方式解决了早期算法网络资源消耗大及计算时间长的缺点.但是,由于 P2P 网络的规模较大,所以 Arai 算法所采用的基于随机游走的采样过程耗时较长(例如,在 1 000 000 个节点的 P2P 网络中,如果端到端的平均耗时为  $2.5s^{[12]}$ ,则一次随机游走采样过程约需 30 分钟左右<sup>[13]</sup>),使得两次采样过程之间可能间隔较长时间,于是,在估计 P2P 网络中时变数据的聚集值时,第 1 次采样的数据可能已经失效.此外,作为一种近似估计方法,Arai 算法没有给出估计值与真实值的误差超出误差阈值范围的概率(下文称为失败概率).

在其他分布式环境中,虽然有些聚集算法考虑了时间因素对网络中数据的影响,例如传感器网络中的实时聚集算法<sup>[14-17]</sup>、无线网络的实时聚集算法<sup>[18,19]</sup>等,但是,由于这些聚集算法大都是由某个能力强大的节点(如基站节点等)来统一管理网络中的时变数据并计算聚集值,而对于大规模 P2P 网络来说,几乎不可能利用一台或几台机器来保存网络中所有时变数据的变化情况,故这些聚集算法不适合应用到 P2P 网络环境中.

## 2 AUS 算法的基本思想

鉴于已有的聚集算法不能较为准确地计算较短时间段内 P2P 网络中时变数据的聚集值,本文将对此展开研究.Arai 等人<sup>[5]</sup>的研究表明:在无时变数据的 P2P 网络中采用基于随机采样的近似聚集可以大大节省网络资源并缩短聚集查询的时间,但这类方法的准确性很大程度上取决于对 P2P 网络中节点的采样是否均衡这一点上.因此,本文将基于均衡采样方法<sup>[13,20]</sup>来构造 P2P 网络中时变数据的近似聚集方法.

### 2.1 预备知识

所谓 P2P 网络中的均衡采样是指在大规模 P2P 网络中利用某种方法对网络中的节点进行选取,使得网络中每个节点被选中的概率均为  $1/N$ ,  $N$  为网络中的节点数.考虑到无结构 P2P 网络可以抽象成一个大规模图,而在大规模图中性能最好的采样方法是基于随机游走的采样方法<sup>[21]</sup>,因此,P2P 网络中的采样方法也大都建立在随机游走的基础上,下面对 P2P 网络中的采样方法进行简要的介绍.为便于讨论,首先定义如下一些相关概念:

**定义 1(P2P 网络中的随机游走过程).** 设有 P2P 网络  $G=(V,E)$ ,  $G$  是一个无向连通图.令随机变量  $X_k$  表示  $t_k$  时刻在  $G$  中访问的节点,  $k=1,2,\dots,n$ .随机变量序列  $X_0, X_1, \dots, X_k, X_{k+1}$  称为 P2P 网络中的一个随机游走过程,且  $P(X_{k+1}=j|X_k=i, X_{k-1}=i_{k-1}, \dots, X_0=i_0) = P(X_{k+1}=j|X_k=i) = p_{ij}$ , 其中,  $p_{ij}$  是从节点  $i$  经一跳到达节点  $j$  的一步转移概率.

**定义 2(基于随机游走的采样过程).** 令  $X_0, X_1, \dots, X_l$  表示 P2P 网络中长为  $l$  的某个随机游走过程,如果对  $\forall j, i_1, i_2 \in V$ , 均有  $P(X_l=j|X_0=i_1) = P(X_l=j|X_0=i_2)$  成立,则称  $X_0, X_1, \dots, X_l$  为基于随机游走的采样过程.

由于 P2P 网络中的随机游走过程是一个 Markov 过程<sup>[22]</sup>,而且 P2P 网络还是一个非二部图<sup>[13]</sup>,因此,P2P 网络的随机游走过程一定会收敛,其收敛速度为  $O(\log N)^{[23]}$ .该结果可以保证在 P2P 网络中进行基于随机游走的采样是可行的,但这种基于随机游走的采样能否达到均衡则取决于 P2P 网络中邻居节点间转移概率的设置.

P2P 网络中最初的采样方法大都是基于简单随机游走来进行的<sup>[22]</sup>,所谓简单随机游走是指对网络中的任意节点  $i$  和  $j$ ,如果  $j$  是  $i$  的邻居节点,则  $i$  到  $j$  的转移概率为  $1/d_i$ , 否则为 0.但是,这种采样方法选中网络中任一节点  $k$  的概率与节点  $k$  的度成正比,所以不是均衡采样.于是,Awan 等人<sup>[13]</sup>提出通过将简单随机游走的转移概率矩阵对称化以达到均衡采样的目的,给出了 3 种方法,分别是 MD(maximum-degree)方法、MH(metropolis-hastings)方法和 RWD(random weight distribution)方法.实验结果表明,RWD 方法的动态适应性、执行效率和精度相对较高.本文的研究将基于 RWD 方法展开,下一节我们将给出 P2P 网络中基于单次均衡采样的时变数据近似聚集算法即 AUS 算法的基本思想.

### 2.2 AUS 算法基本思想

众所周知,数据库中的聚集查询可以用如下方式来表示:

```
SELECT aggregation-op columns FROM T WHERE selection-condition
```

其中,aggregation-op 可以是任意的聚集操作符,如 SUM,COUNT,AVG 等,columns 是  $T$  的任意数字类型的列,或是包含多个列的表达式,selection-condition 决定哪些元组应参与聚集.在本文中,我们主要以 SUM 操作为例来讨论 AUS 算法是如何进行近似聚集查询的.此外,由于每个节点内部如何进行聚集运算属于传统数据库解决的问题,因此,本文假设每个节点可以提供基于某种查询条件的时变数据的聚集值,于是可令  $f(i,t)$  表示在时刻  $t$ 、节点  $i$  中关于某种查询条件的时变数据的 SUM 值.下面给出 AUS 方法的基本思想:

首先,由请求聚集查询的节点  $p$  同时并行发起  $m$  条基于随机游走的均衡采样过程(RWD 均衡采样过程),设这些采样过程所选中的节点集合为  $V_m = \{p_1, p_2, \dots, p_m\}$ .对  $\forall i(1 \leq i \leq m)$ ,当  $p_i$  被选中时,将把本节点的 SUM 值  $f(p_i, t_i)$  返回给节点  $p$ ,其中  $t_i$  为  $p_i$  被选中的时刻.

当  $p$  收集完  $m$  个数据后,按式(1)计算出  $\hat{I}_m(\Delta t)$ ,并将  $\hat{I}_m(\Delta t)$  作为时间区间  $\Delta t$  内,整个 P2P 网络关于某种查询条件的时变数据 SUM 值的近似估计,其中,  $t_{\max} = \max\{t_1, \dots, t_m\}$ ,  $t_{\min} = \min\{t_1, \dots, t_m\}$ ,  $\Delta t = [t_{\min}, t_{\max}]$ ,  $N$  为网络规模.

$$\hat{I}_m(\Delta t) = \frac{f(\eta_1, t_1) + f(\eta_2, t_2) + \dots + f(\eta_m, t_m)}{m} N \quad (1)$$

由 AUS 方法的基本思想可以看出,AUS 算法的有效执行需要解决如下两个关键问题:

(1) 由于 AUS 算法意在解决 P2P 网络中的时变数据的聚集问题,所以需要时间区间  $\Delta t$  (即  $[t_{\min}, t_{\max}]$ ) 足够小,以保证在此时间区间内 P2P 网络中时变数据的变化很小(即可以忽略不计).我们将如何保证时间区间  $\Delta t$  足够小的问题称为样本数据的时效问题.

(2) 由于 AUS 算法是采用随机采样的方式对总体聚集值进行估计的,故其估计精度取决于样本容量  $m$  的大小.于是查询发起者如何才能按照用户的精度要求来确定近似聚集所需的样本容量就成为又一个关键问题,我们称该问题为样本容量的确定问题.

### 3 AUS 算法的数学原理

本节主要讨论如何解决 AUS 算法设计过程中所遇到的样本数据的时效问题及样本容量的确定问题.

#### 3.1 样本数据的时效性保证

为便于讨论,我们首先给出如下概念:

**定义 3(P2P 网络聚集算法的样本数据返回时间区间、时间差).** 设  $\mathcal{A}$  为 P2P 网络中的一个聚集算法,而  $\{f(i_1, t_1), f(i_2, t_2), \dots, f(i_n, t_n)\}$  为  $\mathcal{A}$  计算全局聚集值时所要使用的数据集合,其中  $t_k$  为算法  $\mathcal{A}$  在数据收集阶段访问节点  $i_k$  并获取数据的时刻,而  $f(i_k, t_k)$  为时刻  $t_k$ 、节点  $i_k$  中关于某种查询条件的时变数据的 SUM 值,令  $t_{\max} = \max\{t_1, \dots, t_n\}$ ,  $t_{\min} = \min\{t_1, \dots, t_n\}$ , 则称  $[t_{\max}, t_{\min}]$  为  $\mathcal{A}$  的样本数据返回时间区间,称  $|t_{\max} - t_{\min}|$  为  $\mathcal{A}$  的样本数据返回时间差.

一个 P2P 网络聚集算法的样本数据返回时间差越小,利用它来计算时变数据聚集值的效果就越好.结合第 1 节的分析可知,已有的 P2P 网络中的聚集方法的样本数据返回时间差都很大,虽然 Arai 算法的样本数据返回时间差相对较小,但是对一个具有  $10^6$  个节点的 P2P 网络来说,若端到端的平均耗时为  $2.5s^{[12]}$ ,它的样本数据返回时间差也将超过 30 分钟.正是因为已有的 P2P 网络中的聚集方法的样本数据返回时间差过大,才导致利用它们计算时变数据的聚集值时会产生较大的计算误差.

那么 AUS 算法的样本数据返回时间差有多大呢?根据第 2.2 节的分析,在 AUS 方法中我们采用基于随机游走的 RWD 均衡采样方法来获得样本节点集合  $V_m = \{p_1, p_2, \dots, p_m\}$ ,而且对  $\forall i(1 \leq i \leq m)$ ,当  $p_i$  被某个均衡采样过程选中时,将把本节点的 SUM 值返回给查询发起者  $p$  用以完成近似聚集运算.令  $t_i$  表示  $p_i$  被某个均衡采样过程选中并将本地的 SUM 值返回给查询发起者  $p$  的时刻,令  $t_{\max} = \max\{t_1, \dots, t_n\}$ ,  $t_{\min} = \min\{t_1, \dots, t_n\}$ , 则根据定义 3, AUS 算法的样本数据返回时间差为  $|t_{\max} - t_{\min}|$ .因为随机游走采样过程中所选的路径是不同的,所以 AUS 算法样本数据返回时间差(即  $|t_{\max} - t_{\min}|$ )将大于 0,下面我们利用实验来考察该时间差的大小.

由文献[13]的研究结果可知,对于节点数在  $10^3 \sim 10^9$  之间的 P2P 网络,均衡采样所需的随机游走深度大约在

300~1000 之间.我们利用 Netlogo 软件<sup>[24]</sup>分别模拟了 3 种不同规模的 P2P 网络(网络 1 具有 5 000 个节点、10 000 条边;网络 2 具有 20 000 个节点、50 000 条边;网络 3 具有 50 000、 $10^5$  条边),网络中端到端的延迟不超过 3s<sup>[12]</sup>.在这 3 个网络中完成 50 个深度为  $l$  的随机游走过程  $\{Pr_1, Pr_2, \dots, Pr_{50}\}$ , 令  $t_i^{rw}$  为  $Pr_i (1 \leq i \leq 50)$  的终止时刻(即随机游走深度达到  $l$  的时刻),  $t_{\max}^{rw} = \max\{t_1^{rw}, \dots, t_{50}^{rw}\}$ ,  $t_{\min}^{rw} = \min\{t_1^{rw}, \dots, t_{50}^{rw}\}$ ,  $|\Delta t^{rw}| = t_{\max}^{rw} - t_{\min}^{rw}$ , 我们考察了当随机游走深度  $l$  在 300~1000 之间发生变化时  $|\Delta t^{rw}|$  的大小变化情况, 实验结果如图 1 所示.

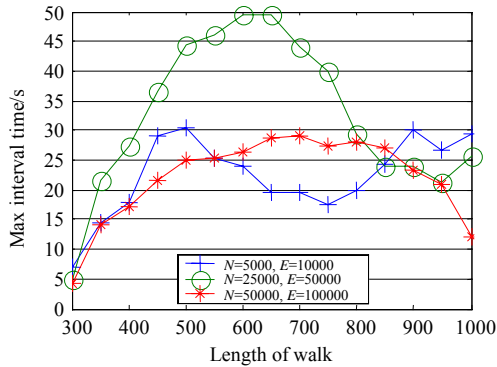


Fig.1 Relation between  $|\Delta t^{rw}|$  and length of random walk

图 1  $|\Delta t^{rw}|$ 与随机游走深度的关系

### 3.2 样本容量的确定

由于 AUS 算法是一种基于随机采样的近似估计算法, 所以需要根据精度要求来确定样本容量. 下面我们给出一个衡量 AUS 算法准确性的精度标准, 即 AUS 方法的误差保证.

假设  $\varepsilon$  是用户给出的相对误差的阈值,  $\delta$  是用户给出的失败概率的阈值, 则我们用如下术语  $\varepsilon$ - $\delta$  成功来衡量近似估计的精度.

**定义 4 ( $\varepsilon$ - $\delta$ 成功).** 设  $1 > \varepsilon > 0, 1 > \delta > 0$ ,  $\hat{I}_m(\Delta t)$  是  $I(\Delta t)$  的估计. 如果概率误差  $P(|\hat{I}_m(\Delta t) - I(\Delta t)| / I(\Delta t) \geq \varepsilon) \leq \delta$ , 则我们称  $\hat{I}_m(\Delta t)$   $\varepsilon$ - $\delta$  成功地估计了  $I(\Delta t)$ , 简称  $\hat{I}_m(\Delta t)$  是  $\varepsilon$ - $\delta$  成功的.

本节的主要任务是给出一种样本容量的确定办法, 以便 AUS 算法能够在其样本数据返回时间区间内时,  $\varepsilon$ - $\delta$  成功地估计网络中时变数据的近似聚集值. 下面, 我们首先给出本节所使用的一些符号的含义:

设  $Con$  表示某种查询条件, 而  $f(i, t)$  表示在时刻  $t$ , 节点  $i$  中关于查询条件  $Con$  的时变数据的 SUM 值.

设  $m$  为网络中某个节点  $p$  同时发起的均衡采样过程数(即 AUS 方法中的样本容量).  $V_m = \{p_1, p_2, \dots, p_m\}$  表示这  $m$  个均衡采样过程所选中的节点集合,  $t_i$  为  $p_i$  被选中的时刻. 令  $t_{\max} = \max\{t_1, \dots, t_m\}$ ,  $t_{\min} = \min\{t_1, \dots, t_m\}$ ,  $\Delta t = [t_{\min}, t_{\max}]$ . 根据第 3.1 节的讨论, 对于  $\forall i \in V, \forall t \in \Delta t$ , 均有  $f(i, t) = f(i, \Delta t)$ , 其中  $f(i, \Delta t)$  表示节点  $i$  在时间区间  $\Delta t$  内关于查询条件  $Con$  的时变数据的 SUM 值, 而在  $\Delta t$  时间区间内, 整个 P2P 网络关于查询条件  $Con$  的时变数据的真实 SUM 值为  $I(\Delta t) = \sum_{i \in V, t \in \Delta t} f(i, t) = \sum_{i=1}^N f(i, \Delta t)$ , 其中,  $N$  为网络中的节点数.  $\hat{I}_m(\Delta t)$  表示 AUS 算法利用  $m$

个样本按照式(1)计算出的近似聚集值.

下面的定理 1 与定理 2 可以保证当样本容量满足式(2)时, 利用 AUS 算法按照式(1)计算出的  $\hat{I}_m(\Delta t)$   $\varepsilon$ - $\delta$  成功地估计了  $I(\Delta t)$  (真实聚集值), 其中,  $\phi_{\frac{\delta}{2}}$  是  $N(0, 1)$  分布的上侧  $\delta/2$  分位数,  $M(\Delta t), m(\Delta t)$  分别为网络中  $f(i, \Delta t)$  的上限值和下限值, 即  $M(\Delta t) = \sup_{i \in V} (f(i, \Delta t))$ ,  $m(\Delta t) = \inf_{i \in V} (f(i, \Delta t))$ .

由图 1 可以看出,  $|\Delta t^{rw}|$  不超过 50s, 而根据该实验的设置, 我们知道  $|\Delta t^{rw}|$  的大小约等于 AUS 算法的样本数据返回时间差, 故 AUS 算法的样本数据返回时间差约为 50s 左右. 可见, AUS 算法的样本数据返回时间差远远小于已有的 P2P 网络中的聚集方法的样本数据返回时间差, 这也说明 AUS 算法具有更强的时效性.

同时, 由于对大多数 P2P 系统来讲, 需经过一系列的操作才能引起网络中的数据发生变化(如在 P2P 电子商务系统中, 节点需要与其他节点发生交易才能引起自身的交易额、可信度等数据发生变化), 而这些操作往往需要花费几分钟, 因此在 AUS 算法样本数据返回的时间区间内 P2P 网络中的数据变化很小, 为便于讨论, 我们假定在该时间区间内网络中的数据是不变的.

$$m \geq \frac{\phi_{\frac{\varepsilon}{2}}^2}{\varepsilon^2} \left( \frac{M(\Delta t)}{m(\Delta t)} - 1 \right) \quad (2)$$

**定理 1.**  $E(\hat{I}_m(\Delta t)) = I(\Delta t)$ .

证明:因为  $V_m = \{p_1, p_2, \dots, p_m\}$  是由节点  $p$  并行发起的  $m$  个均衡采样过程获得的,所以  $X_{t_1} = p_1, \dots, X_{t_m} = p_m$  是相互独立的,而且对  $\forall i(1 \leq i \leq m)$ ,  $P(X_{t_i} = p_i) = 1/N$ , 其中,  $N$  为网络中的节点数.于是,

$$E(Nf(p_i, \Delta t)) = \sum_{p_i=1}^N Nf(p_i, t_i)P(X_{t_i} = p_i) = \sum_{p_i=1}^N Nf(p_i, \Delta t)P(X_{t_i} = p_i) = \sum_{p_i=1}^N Nf(p_i, \Delta t)(1/N) = \sum_{p_i=1}^N f(p_i, \Delta t) = I(\Delta t).$$

从而,

$$\begin{aligned} E(\hat{I}_m(\Delta t)) &= E\left(\frac{f(p_1, t_1) + f(p_2, t_2) + \dots + f(p_m, t_m)}{m} N\right) = E\left(\frac{f(p_1, \Delta t) + f(p_2, \Delta t) + \dots + f(p_m, \Delta t)}{m} N\right) \\ &= (1/m)(E(Nf(p_1, \Delta t)) + \dots + E(Nf(p_m, \Delta t))) = I(\Delta t). \end{aligned}$$

故  $E(\hat{I}_m(\Delta t)) = I(\Delta t)$ . □

**定理 2.** 当样本容量满足式(2)时,利用 AUS 方法对全局聚集值的估计是  $\varepsilon$ - $\delta$  成功的.

证明:根据定理 1 及其证明过程,  $X_{t_1} = p_1, \dots, X_{t_m} = p_m$  是相互独立的,且  $E(\hat{I}_m(\Delta t)) = I(\Delta t)$ , 从而有,

$$\begin{aligned} \text{var}(\hat{I}_m(\Delta t)) &= (1/m) \text{var}(f(p_i, t_i)N) = (1/m) \text{var}(f(p_i, \Delta t)N) = (1/m)(E(Nf(p_i, \Delta t))^2 - E^2(Nf(p_i, \Delta t))) \\ &= (1/m) \left( \sum_{p_i=1}^N f^2(p_i, \Delta t)N^2 \frac{1}{N} - I(\Delta t)^2 \right) \end{aligned} \quad (3)$$

又因为  $M(\Delta t) = \sup_k (f(k, \Delta t))$ , 所以有  $f^2(p_i, \Delta t) \leq M(\Delta t)f(p_i, \Delta t)$ , 将该结果代入式(3)可得:

$$\begin{aligned} \text{var}(\hat{I}_m(\Delta t)) &= (1/m) \left( \sum_{p_i=1}^N f^2(p_i, \Delta t)N - I(\Delta t)^2 \right) \\ &\leq (1/m) \left( M(\Delta t) \sum_{p_i=1}^N f(p_i, \Delta t)N - I(\Delta t)^2 \right) \\ &= (1/m)(NM(\Delta t)I(\Delta t) - I(\Delta t)^2) \end{aligned} \quad (4)$$

由于 P2P 网络的规模较大(节点数一般都大于 10 000),即使按很低的精度要求对其进行采样所需的样本容量也要远大于  $30^{[5]}$ .根据中心极限定理,当  $m \geq 30$  时,  $\hat{I}_m(\Delta t)$  的近似服从  $N(E(\hat{I}_m(\Delta t)), \text{var}(\hat{I}_m(\Delta t)))^{[25]}$ .由定理 1 可知,  $E(\hat{I}_m(\Delta t)) = I(\Delta t)$ , 故  $\hat{I}_m(\Delta t) \sim N(I(\Delta t), \text{var}(\hat{I}_m(\Delta t)))$ , 于是有,

$$P\left(|\hat{I}_m(\Delta t) - I(\Delta t)| \geq \phi_{\frac{\varepsilon}{2}} \sqrt{\text{var}(\hat{I}_m(\Delta t))}\right) \leq \delta \quad (5)$$

因此,要想使  $\hat{I}_m(\Delta t)$  能够  $\varepsilon$ - $\delta$  成功地估计  $I(\Delta t)$ , 根据定义 4, 即使  $P\left(|(\hat{I}_m(\Delta t) - I(\Delta t))/I(\Delta t)| \geq \varepsilon_r\right) \leq \delta$  成立, 则只要下面的式(6)成立即可.

$$\phi_{\frac{\varepsilon}{2}} \sqrt{\text{var}(\hat{I}_m(\Delta t))} \leq \varepsilon_r I(\Delta t) \quad (6)$$

将式(4)代入式(6)即可得到式(7):

$$\phi_{\frac{\varepsilon}{2}}^2 (NM(\Delta t)I(\Delta t) - I(\Delta t)^2) \leq m\varepsilon_r^2 I^2(\Delta t) \quad (7)$$

对式(7)进行求解可得:

$$m \geq \frac{\phi_{\frac{\varepsilon}{2}}^2 (NM(\Delta t)I(\Delta t) - I(\Delta t)^2)}{\varepsilon^2 I^2(\Delta t)} = \frac{\phi_{\frac{\varepsilon}{2}}^2}{\varepsilon^2} \left( \frac{NM(\Delta t)}{I(\Delta t)} - 1 \right) \quad (8)$$

又因为  $I(\Delta t) = \sum_{i=1}^N f(i, t_i) = \sum_{i=1}^N f(i, \Delta t) \geq Nm(\Delta t)$ , 故只要  $m$  满足式(2), 则有  $P\left(|(\hat{I}_m(\Delta t) - I(\Delta t))/I(\Delta t)| \geq \varepsilon_r\right) \leq \delta$  成立, 即利用 AUS 方法计算出的  $\hat{I}_m(\Delta t)$  是  $\varepsilon$ - $\delta$  成功的. □

由定理 1 可知,样本容量  $m$  大小的确定不仅与  $\varepsilon, \delta$  有关,而且还与  $M(\Delta t)$  和  $m(\Delta t)$  ( $f(i, \Delta t)$  的上下限) 有关.  $M(\Delta t)$  和  $m(\Delta t)$  大小的确定与具体的应用有关. 如果用户的查询条件较为固定、 $M(\Delta t)$  和  $m(\Delta t)$  在某个较短的时间段  $\tau (\tau \gg |\Delta t|)$  内的波动较小时,则可以利用文献[26]中的方法对  $M(\Delta t)$  和  $m(\Delta t)$  进行周期性的统计;如果  $M(\Delta t)$  和  $m(\Delta t)$  与历史数据关系密切,则可以采用直方图来给出  $M(\Delta t)$  和  $m(\Delta t)$  的估计<sup>[27]</sup>;如果用户根据系统保存的历史记录较容易判断  $M(\Delta t)$  和  $m(\Delta t)$  的概率分布情况(例如,在 P2P 电子商务系统中,对某种商品价格上下限的判断、对节点可信度上下限的判断、对气温等环境指标的上下限的判断等),则可以采用经济学中的主观概率加权法<sup>[28]</sup> 来对  $M(\Delta t)$  和  $m(\Delta t)$  进行估计.

按照第 3.1 节和第 3.2 节所给出的方法即可解决第 2.2 节所提及的两个关键问题,下一节我们将给出 AUS 算法的实现及其性能分析.

#### 4 AUS 算法及其性能分析

下面首先给出 AUS 算法的伪码描述.

算法(AUS 近似聚集算法).

输入: Sink (发出聚集查询请求的某一特定节点);  $\Delta t$  (需要计算近似聚集值的时间段);

$\varepsilon, \delta$  (用户给出的相对误差及失败概率阈值);  $M(\Delta t), m(\Delta t), N$  (网络的规模).

输出:  $\hat{I}_m(\Delta t)$  ( $\Delta t$  时间区间内时变数据的近似聚集值)

1.  $\hat{I}_m(\Delta t) := 0$
2.  $\phi_{\frac{\delta}{2}} := N(0, 1)$  的上侧  $\frac{\delta}{2}$  分位数
3.  $m := \frac{\phi_{\frac{\delta}{2}}^2}{\varepsilon^2} \left( \frac{M(\Delta t)}{m(\Delta t)} - 1 \right)$  // 计算样本容量
4.  $V_m := \text{RWD}(\text{Sink}, m)$  //  $V_m$  是从 Sink 节点出发,同时发起  $m$  个 RWD 均衡采样过程所选中的节点集合
5. Fetch local sum from  $V_m$  // 从集合  $V_m$  中的节点处取回 sum 值
6.  $I := \text{average}(f(p_1, t_1), f(p_2, t_2), \dots, f(p_m, t_m))$  // where  $\{p_1, p_2, \dots, p_m\} = V_m$
7.  $\hat{I}_m(\Delta t) := N \times I$
8. return  $\hat{I}_m(\Delta t)$

由上述算法可以看出,算法的运行时间主要消耗在第 4 步和第 6 步上.由第 2.1 节的分析可知,P2P 网络中的均衡采样算法的时间复杂度为  $O(\log N)$ ,即第 4 步的时间复杂度为  $O(\log N)$ .而第 6 步计算均值的时间与样本容量的大小有关,根据第 3.2 节的分析,样本容量的大小仅与精度标准及网络中的数据分布有关,而与网络规模无关,所以第 6 步的时间复杂度为  $O(1)$ .从而有,AUS 算法的时间复杂度为  $O(\log N)$ .

其次,从统计学角度来看,AUS 算法属于一种估计方法,所以我们需要考察通过 AUS 算法计算的  $\hat{I}_m(\Delta t)$  是否是  $I(\Delta t)$  的无偏估计和相合估计.下文中所用的符号与第 3.2 节中的相同.

由定理 1 可知  $E(\hat{I}_m(\Delta t)) = I(\Delta t)$  成立,故  $\hat{I}_m(\Delta t)$  为  $I(\Delta t)$  的无偏估计.

下面我们通过定理 3 来证明  $\hat{I}_m(\Delta t)$  也是  $I(\Delta t)$  的相合估计.

定理 3.  $\hat{I}_m(\Delta t)$  为  $I(\Delta t)$  的相合估计.

证明:根据定理 2 证明过程中的式(4)有:

$$\text{var}(\hat{I}_m(\Delta t)) \leq \frac{1}{m} (NM(\Delta t)I(\Delta t) - I(\Delta t)^2) = \frac{N^2}{m} M^2(\Delta t) \left( \frac{I(\Delta t)/N}{M(\Delta t)} \left( 1 - \frac{I(\Delta t)/N}{M(\Delta t)} \right) \right) \quad (9)$$

令  $\gamma = \frac{I(\Delta t)/N}{M(\Delta t)}$ , 将  $\gamma = \frac{I(\Delta t)/N}{M(\Delta t)}$  代入式(9)可得:

$$\text{var}(\hat{I}_m(\Delta t)) \leq \frac{N^2}{m} M^2(\Delta t) \gamma(1-\gamma) \quad (10)$$

考虑到  $\gamma(1-\gamma) \leq \frac{1}{4}$ , 所以有:  $\text{var}(\hat{I}_m(\Delta t)) = \frac{N^2}{m} M^2(\Delta t) p(1-p) \leq \frac{M^2(\Delta t) N^2}{4m}$ , 根据 Chebyshev 不等式, 对于任意  $\varepsilon' > 0$ , 均有下式成立:

$$P(|\hat{I}_m(\Delta t) - I(\Delta t)| \geq \varepsilon') \leq \frac{\text{var}(\hat{I}_m(\Delta t))}{\varepsilon'^2} \tag{11}$$

将式(10)代入式(11)可得:  $P(|\hat{I}_m(\Delta t) - I(\Delta t)| \geq \varepsilon') \leq \frac{N^2 M^2(\Delta t)}{4m\varepsilon'^2}$ . 由于  $M(\Delta t) \ll \infty, N \ll \infty$ , 所以,  $\lim_{m \rightarrow \infty} P(|\hat{I}_m(\Delta t) - I(\Delta t)| \geq \varepsilon') = 0$ , 而  $P(|\hat{I}_m(\Delta t) - I(\Delta t)| \geq \varepsilon') \geq 0$ , 故  $\lim_{m \rightarrow \infty} P(|\hat{I}_m(\Delta t) - I(\Delta t)| \geq \varepsilon') = 0$ , 即  $\hat{I}_m(\Delta t)$  是  $I(\Delta t)$  的相合估计.  $\square$

### 5 实验及结果分析

本节通过实验将 AUS 算法与 Arai 算法加以对比. 我们利用 NetLogo<sup>[24]</sup> 模拟了一个具有 10 000 个节点的 P2P 网络. NetLogo 是人工智能领域中较为流行的多智能体仿真软件, 利用它可以很容易地为独立和相似的智能体建模. 实验中, 两种算法所调用的均衡采样方法都是 RWD 均衡采样算法<sup>[13]</sup> (RWD 算法采用的随机游走深度为  $25 \log N$  ( $N$  为网络中节点的个数)), 模拟网络中设定端到端的平均时间延迟为 2s. Arai 算法第 1 次采样的预取样本容量为 100. 由于 Arai 算法中未对失败概率  $\delta$  进行讨论, 所以本实验只考察 Arai 算法的样本容量和估计误差随  $\varepsilon_r$  的变化情况. PC 机的基本配置为 Pentium IV 2.4G CPU, 512M 内存, 100M 网卡, 操作系统为 Windows XP.

#### 5.1 实验1: 静态数据的聚集实验

本实验主要对比两种算法在静态数据的聚集运算中的样本容量及实际误差.

静态数据的生成方法为: 模拟网络中每个节点保存具有单一属性的元组, 该属性的取值范围为 1~300, 每个节点保存 10~20 个这样的元组. 两种算法所执行的聚集操作均为 “select sum(A) from peer p”. 由于利用采样方式对真实值进行估计的过程是一个随机过程, 故我们采用重复 50 次实验然后取均值的方法来消除实验数据的随机性. 实验结果如图 2 和图 3 所示.

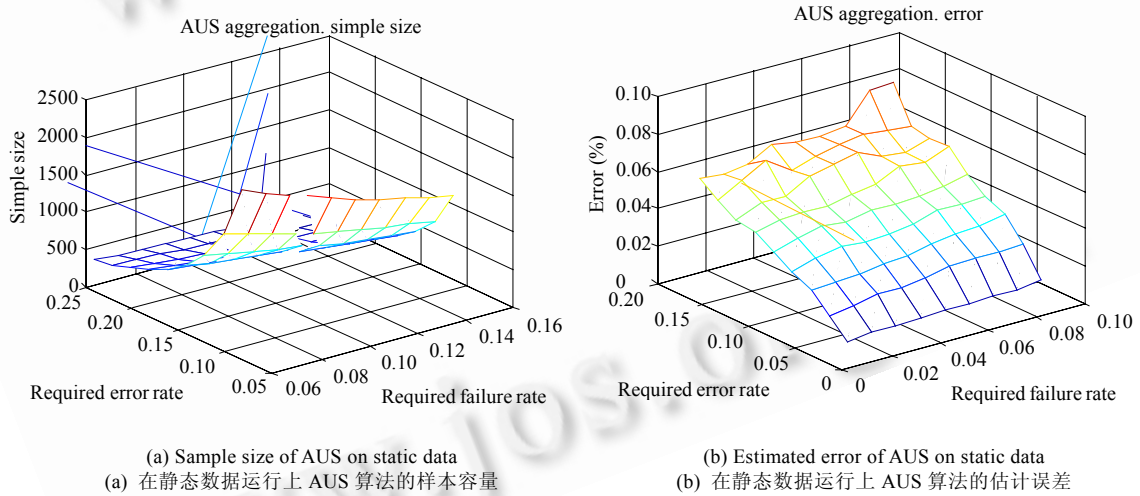


Fig.2 Sample size and estimated error of AUS on static data

图 2 在静态数据上运行 AUS 的样本容量与估计误差

从图 2(a)和图 3(a)可以看出, AUS 算法的样本容量略大于 Arai 算法, 这是因为 AUS 算法仅仅通过网络中数据的取值范围(上限值及下限值)来对所需的样本容量进行估计, 并没有利用网络中其他的数据信息. 而 Arai 算法在对样本容量估计的过程中通过预取的方式利用了网络中的部分数据信息. 但是, 正是由于 Arai 算法需要预取一定量的数据, 才会导致其执行时间较长. 由图 2(b)可以看出, AUS 算法实际产生的误差低于 8%, 且都能够满



足预定的误差限制.而由图 3(b)可以看出,Arai 算法在精度要求较高时(例如错误率在 6%以下)会出现实际误差不能满足预定误差标准的情况.这是因为当精度要求较高时 Arai 算法对样本容量的大小估计不足而造成的.但是,当精度要求较低时(例如,错误率在 10%~20%),Arai 算法既能满足误差阈值,又能使所需样本容量较小.

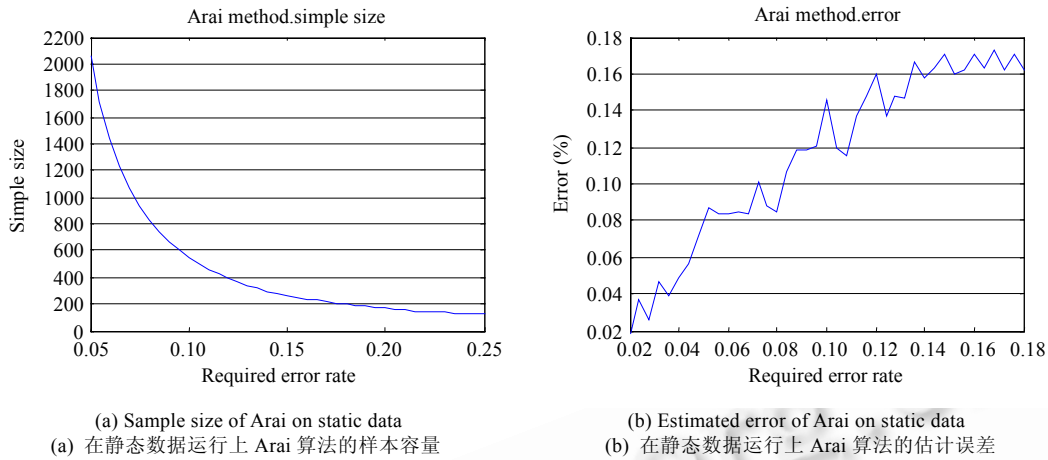


Fig.3 Sample size and estimated error of Arai on static data  
图 3 在静态数据上运行 Arai 的样本容量与估计误差

综上所述,在计算 P2P 网络中静态数据的聚集值时,如果所需的近似聚集结果的精度要求较高,则 AUS 算法的性能较好,而如果精度要求较低,则 Arai 更加节省网络资源.

### 5.2 实验2:时变数据的聚集实验

本实验主要对比两种算法在计算时变数据的聚集值时的样本容量及实际误差.

实验中我们通过运行 P2P 电子商务系统中一种简单的信任机制来获得实验所需数据:网络中的节点按交易情况对其邻居节点进行评价,节点的可信度是根据其他节点对该节点的历史评价来计算的.易见节点的可信度是时变数据.我们利用 AUS 及 Arai 算法对该类数据执行 SUM 聚集操作,并将这两种算法的执行结果进行了对比.实验结果如图 4 和图 5 所示.

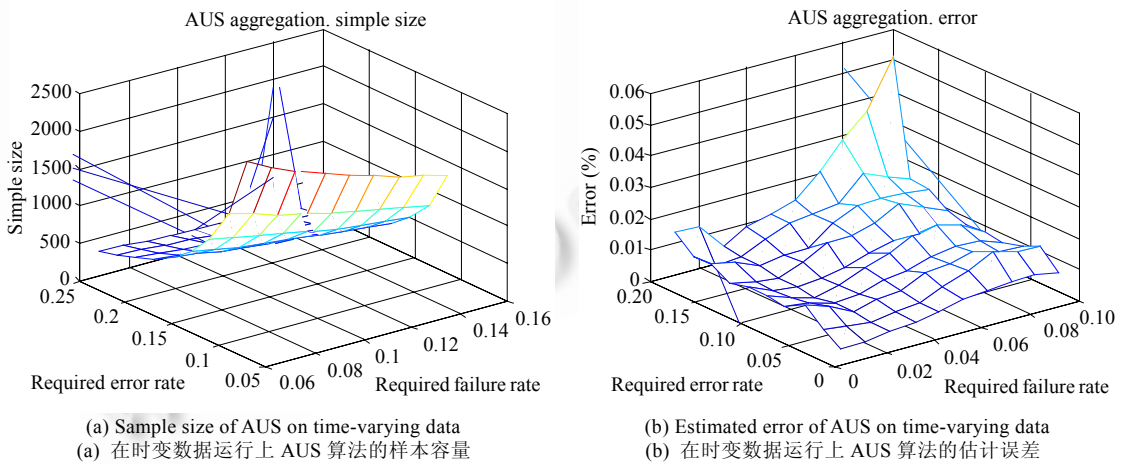


Fig.4 Sample size and estimated error of AUS on time-varying data  
图 4 在时变数据上运行 AUS 的样本容量与估计误差

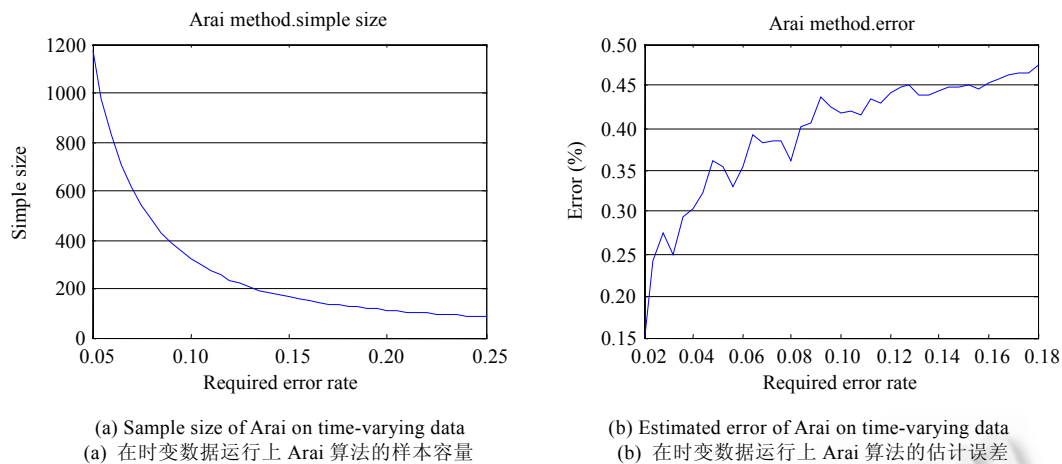


Fig.5 Sample size and estimated error of Arai on time-varying data

图 5 在时变数据上运行 Arai 的样本容量与估计误差

由图 4(b)可以看出,AUS 算法在计算时变数据的近似聚集值时所产生的误差在 6%以下.而且精度要求越高,AUS 算法的性能表现越突出.由图 5(b)可以看出,Arai 算法在计算过程中完全不能满足误差阈值,最高误差甚至达到 40%左右.这是因为 Arai 算法在计算过程中使用了时间上过期的数据,从而造成对所需样本容量大小的估计产生失误的缘故.如图 5(a)所示,Arai 算法利用过期的数据所估计的样本容量的大小几乎是 AUS 算法的 0.5 倍(如图 4(a)所示).综上所述,我们发现 Arai 算法不适合用来计算时变数据的近似聚集值,而 AUS 算法在进行时变数据的近似聚集时却表现出较好的性能.

### 5.3 实验3:时间复杂度实验

本实验主要对比两种算法的时间复杂度.实验结果如图 6 所示.

由图 6 可以看出,Arai 算法的时间复杂度约是 AUS 算法的 1.6 倍.这是因为 Arai 算法调用了两次均衡采样算法,而 AUS 算法仅调用了 1 次,所以理论上 Arai 算法的时间复杂度约是 AUS 的 2 倍,之所以未到 2 倍,是因为 AUS 算法的样本容量略大于 Arai 算法的样本容量,从而在采样过程中产生了较多的延迟.

综上所述,我们发现在计算 P2P 网络中静态数据的聚集值时,AUS 算法虽然比 Arai 算法的样本容量略大些,但其计算精度及时间复杂性均优于 Arai 算法.而在计算 P2P 网络中时变数据的聚集值时,Arai 算法可能产生很大的误差,而 AUS 算法能够较为准确地给出时变数据的近似聚集值.

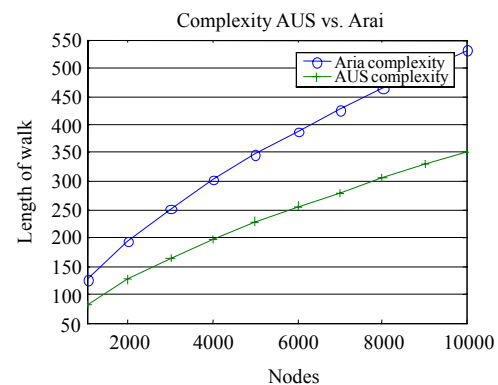


Fig.6 Comparison of time complexity between AUS and Arai

图 6 AUS 算法与 Arai 算法时间复杂性对比

## 6 结 论

由于目前 P2P 网络的聚集算法均不适用于求解时变数据的聚集值,所以本文提出了一种基于单次均衡采样的 P2P 网络时变数据的近似聚集算法——AUS 算法,并通过理论证明和实验验证说明了该算法的正确性和有效性.因为本文中提出的 AUS 算法估计的准确性取决于对网络中数据上、下限的预测,所以在将来的工作中,我们还需对其进行研究.

**References:**

- [1] Van Renesse R, Birman KP, Vogels W. Astrolabe: A robust and scalable technology for distributed system monitoring, management, and data mining. *ACM Trans. on Computer Systems*, 2003,21(2):164–206.
- [2] Bawa M, Garcia-Molina H, Gionis A, Motwani R. Estimating aggregates on a peer-to-peer networks. Technical Report, 2003-24, Stanford: Stanford University, 2003. 1–13.
- [3] Huebsch R, Chun B, Hellerstein J, Loo BT, Maniatis P, Roscoe T, Shenker S, Stoica I, Yumerefendi AR. The architecture of PIER: An Internet-scale query processor. In: Stonebraker M, Weikum G, DeWitt D, eds. *Proc. of the 2005 Conf. on Innovative Data Systems Research. Asilomar: VLDB*, 2005. 28–43.
- [4] Kempe D, Dobra A, Gehrke J. Gossip-Based computation of aggregate information. In: Tittsworth F, Torres A, eds. *Proc. of the 44th Annual IEEE Symp. on Foundations of Computer Science*. Cambridge: IEEE Computer Society, 2003. 482–491.
- [5] Benjamin A, Gautam D, Dimitrios G, Vana K. Approximating aggregation queries in peer-to-peer networks. In: Liu L, Reuter A, Whang KY, Zhang J, eds. *Proc. of the 22nd Int'l Conf. on Data Engineering*. Los Alamitos: IEEE Computer Society Press, 2006. 642–654.
- [6] Benjamin A, Song L, Dimitrios G. Efficient data sampling in heterogeneous peer-to-peer networks. In: Liu L, Reuter A, Whang K-Y, Zhang J, eds. *Proc. of the 7th IEEE Int'l Conf. on Data Mining (ICDM 2007)*. Omaha: IEEE Computer Society, 2007. 23–32.
- [7] Datta S, Kargupta H. Uniform data sampling from a peer-to-peer network. In: Li X, Fleisch B, Picco GP, eds. *Proc. of the 27th Int'l Conf. on Distributed Computing Systems (ICDCS 2007)*. Los Alamitos: IEEE Computer Society, 2007. 50–57.
- [8] Chaudhuri S, Das G, Srivastava U. Effective use of block-level sampling in statistics estimation. In: Weikum G, König AC, Deßloch S, eds. *Proc. of the 2004 ACM SIGMOD Int'l Conf. on Management of Data*. Paris: ACM Press, 2004. 287–298.
- [9] Babcock B, Chaudhuri S, Das G. Dynamic sample selection for approximate query processing. In: Alon Y, Halevy, *et al.*, eds. *Proc. of the 2003 ACM SIGMOD Int'l Conf. on Management of Data*. San Diego: ACM Press, 2003. 539–550.
- [10] Chaudhuri S, Das G, Datar M, Motwani R, Narasayya V. Overcoming limitations of sampling for aggregation queries. In: *Proc. of the 17th Int'l Conf. on Data Engineering (ICDE 2001)*. Washington: IEEE Computer Society, 2001. 534–542. <http://research.microsoft.com/pubs/76563/icde01.pdf>
- [11] Acharya S, Gibbons PB, Poosala V. Aqua: A fast decision support system using approximate query answers. In: Atkinson MP, Orłowska ME, Valduriez P, Zdonik SB, Brodie ML, eds. *Proc. of the 25th Int'l Conf. on Very Large Data Bases (VLDB)*. Edinburgh: Morgan Kaufmann Publishers, 1999. 754–757.
- [12] Rakesh K, Mario G. Analysis of end-to-end measurements to investigate delays on the Internet. In: *Proc. of the DIMACS Workshop on Performance of Real-Time Applications on the Internet*. 1996. <http://dimacs.rutgers.edu/Workshops/RealTime/Kushwaha.ps>
- [13] Awan A, Ferreira RA, Jagannathan S. Distributed uniform sampling in unstructured peer-to-peer networks. In: Klefstad D, Corsaro A, Mahmoud Q, eds. *Proc. of the 39th Annual Hawaii Int'l Conf. on System Sciences*. Washington: IEEE Computer Society, 2006.223.3.
- [14] Sharaf MA, Beaver J, Labrinidis A, Chryanthi PK. Tina: A scheme for temporal coherency-aware in-network aggregation. In: *Proc. of the 3rd ACM Workshop on Data Engineering for Wireless and Mobile Access*. New York: ACM Press, 2003. 69–76. <http://dmod.cs.uoi.gr/DandyPage/papers/mobide03.pdf>
- [15] Solis I, Obraczka K. Efficient continuous mapping in sensor networks using isolines. In: Cristina NR, Antonio C, Zhang WS, eds. *Proc. of the 2005 MobiQuitous*. San Diego: IEEE Computer Society, 2005. 325–332.
- [16] Pattem S, Krishnamachari B, Govindan R. The impact of spatial correlation on routing with compression in wireless sensor networks. *ACM Trans. on Sensor Networks*, 2008,4(4):24–57.
- [17] Silberstein A, Braynard R, Yang J. Constraint chaining: On energy-efficient continuous monitoring in sensor networks. In: Labrinidis A, Cooper BF, Loannidis Y, eds. *Proc. of the 2006 ACM SIGMOD Int'l Conf. on Management of Data*. Chicago: ACM Press, 2006. 157–168.
- [18] Navda V, Kashyap A, Ganguly S, Izmailov R. Real-Time video stream aggregation in wireless mesh network. In: *Proc. of the IEEE the 17th Int'l Symp. on Personal, Indoor and Mobile Radio Communication*. Helsinki: IEEE Computer Society, 2006. 1–7. <http://www.wings.cs.sunysb.edu/~vnavda/pubs/vnavda-ganges-pimrc06.pdf>

- [19] Tao Y, Kollios G, Considine J, Li F, Papadias D. Spatio-Temporal aggregation using sketches. In: Gray J, Shenory PJ, eds. Proc. of the 20th Int'l Conf. on Database Engineering (ICDE). Boston: IEEE Computer Society, 2004. 214–225.
- [20] Daniel S, Reza R, Nick D, Subhabrata S, Walter W. On unbiased sampling for unstructured peer-to-peer networks. In: Shekhar S, Saran R, eds. Proc. of the ACM/USENIX Internet Measurement Conf. (IMC 2006). Rio de Janeiro: ACM Press, 2006. 27–40.
- [21] Leskovec J, Faloutsos C. Sampling from large graphs. In: Labrinidis A, Cooper BF, Loannidis Y, eds. Proc. of the 12th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. New York: ACM Press, 2006. 631–636.
- [22] Gkantsidis C, Mihail M, Saberi A. Random walks in peer-to-peer networks. In: Zhang ZS, Low S, eds. Proc. of the IEEE INFOCOM 2004. Piscataway: IEEE Press, 2004. 120–130.
- [23] Lovasz L. Random walks on graphs: A survey. *Combinatorics, Paul Erdos is Eighty*, 1993,2(1):1–46.
- [24] Wilensky U. Netlogo. Center for connected learning and computer-based modeling. Evanston: Northwestern University, 1999. <http://ccl.northwestern.edu/netlogo>
- [25] Bernstein S, Bernstein R, Wrote; Shi DJ, Trans. *Elements of Statistics II: Inferential Statistics*. Beijing: Science Press, 2002. 83–84 (in Chinese).
- [26] Akbarinia R, Martins V, Pacitti E, Valduriez P. Top- $k$  query processing in the APPA P2P system. In: Dayde M, Palam JM, Couthino AL, Pacitti E, Lopes JC, eds. Proc. of the Int'l Conf. on High Performance Computing for Computational Science (VecPar). Berlin: Springer-Verlag, 2006. 158–171.
- [27] Michel S, Triantafillou P, Weikum G. Klee: A framework for distributed top- $k$  query algorithms. In: B-hm K, Jensen CS, Haas LM, Kersten ML, Larson P, Ooi BC, eds. Proc. of the 31st Int'l Conf. on Very Large Data Bases (VLDB). Trondheim: ACM Press, 2005. 637–648.
- [28] Xu GY, Hu QY. *Statistical Forecasting and Decision Making*. Shanghai: Shanghai University of Finance & Economics Press, 1998. 16–20 (in Chinese).

## 附中文参考文献:

- [25] Bernstein S, Bernstein R, 著; 史道济, 译. *统计学原理——推断统计学(下册)*. 北京: 科学出版社, 2002. 83–84.
- [28] 徐国祥, 胡清友. *统计预测与决策*. 上海: 上海财经大学出版社, 1998. 16–20.



程思瑶(1982—),女,黑龙江佳木斯人,博士生,主要研究领域为对等计算.



李建中(1951—),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为数据管理,传感器网络,并行计算,计算生物学.



姜守旭(1968—),男,博士,教授,CCF 高级会员,主要研究领域为对等计算,传感器网络与信任管理.