软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2011,22(9):2149−2165 [doi: 10.3724/SP.J.1001.2011.03951]

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

传感网络中误差有界的分段逼近数据压缩算法*

张建明[1], 林亚平[2+], 傅 明[1], 周四望[2]

[1](长沙理工大学 计算机与通信工程学院,湖南 长沙 410004)

[2](湖南大学 信息科学与工程学院,湖南 长沙 410082)

Piecewise Approximation Based Data Compression Algorithm with Error Bound in Wireless Sensor Networks

ZHANG Jian-Ming[1], LIN Ya-Ping[2+], FU Ming[1], ZHOU Si-Wang[2]

[1](School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha 410004, China)

[2](School of Information Science and Engineering, Hu'nan University, Changsha 410082, China)

+ Corresponding author: E-mail: yplin@hnu.cn

Zhang JM, Lin YP, Fu M, Zhou SW. Piecewise approximation based data compression algorithm with error bound in wireless sensor networks. *Journal of Software*, 2011,22(9):2149−2165. http://www.jos.org.cn/1000-9825/3951.htm

Abstract: Wireless sensor networks usually have limited energy and transmission capacity. A critical and practical demand is to online compress sensor data streams continuously. This paper makes the following contributions. First, using the built-in buffer of sensor node, a piecewise constant approximation based data compression algorithm with infinite norm error bound is presented, which is named PCADC-sensor and is a near online algorithm. Second, with infinite norm and square norm error bound respectively, this study proposes two online piecewise linear approximation based data compression algorithms in sensor node, named PLADC-sensor. A necessary and sufficient condition of PLA uniform approximation is given. Third, a piecewise linear representations based data compression algorithm in cluster head or sink, named PLRDC-cluster is presented. It does not need raw sensory data and can be applied to calculate aggregate functions. Last, the experiments on real-world sensor dataset show that the proposed algorithms match the sensor data stream model and can achieve significant data reduction.

Key words: wireless sensor networks; data compression; segmentation; uniform approximation; square approximation

摘 要: 无线传感器网络通常能量、带宽有限.一个关键而实用的需求是,在保证数据质量的情况下,对持续到达的采样数据进行在线式压缩.主要贡献:① 利用传感器节点内置的缓冲区,提出了单传感器节点上基于分段常量逼近的准在线式数据压缩算法(PCADC-sensor),并给出了在无穷范数误差度量下的实现;② 提出了单传感器节点上基于分段线性逼近的在线式数据压缩算法(PLADC-sensor).分别在无穷范数和 2 范数误差度量的情况下给出了计算 PLA 的两种简单快速算法,推导了分段线性一致逼近的充要条件;③ 簇头或基站无需接收原始采样数据,提出了基于原始数据的分段线性表示的压缩算法(PLRDC-cluster),推导了同一节点不同时段、不同节点相同时段两种情况下

* 基金项目: 国家自然科学基金(60973031, 60973127); 湖南省科技计划(2010FJ6005); 长沙理工大学人才引进基金
收稿时间: 2009-09-30; 修改时间: 2010-03-04; 定稿时间: 2010-10-09

的计算公式.实验结果表明,这些算法较好地匹配了传感器数据流模型,显著减少了冗余数据传输.

关键词: 传感器网络;数据压缩;分段;一致逼近;平方逼近
中图法分类号: TP393 文献标识码: A

无线传感器网络(WSN)能够协作地实时监测、感知、采集网络分布区域内的各种环境或监测对象的数据,并对这些数据进行处理,获得详尽而准确的信息,传送到需要这些信息的用户[1].WSN 是物联网时代的关键支撑技术.组成 WSN 的传感器节点体积小、价格廉,能量和通信带宽有限.直接将节点采集的原始数据传输到基站是不可行的,一是带宽不够,二是能量将很快耗尽.无线通信消耗了大部分能量,研究[2]表明,将 1 位数据传输 100 米的耗能可以执行 3000 条 CPU 指令.因此,在传感器节点上要有效减少传输的数据量,从而延长网络寿命.在基站或数据中心,将源源不断的原始数据直接存档也是不可行的,需要将其精简近似表示,以减少存储的数据量.这些压缩表示最好能直接(或解压缩后)用于相应的监控、数据分析任务.

受噪音、节点失效、无线通信不可靠等因素影响,感知数据的获取、处理和传输中常常存在一定的误差.如果用户并不需要非常精确的结果,那么通过牺牲数据精度,可以达到减少通信和存储数据量的目的.针对具体应用场景,寻找能效、数据精度/粒度、通信延时的一个折衷.例如:使用聚集函数(aggregate function)可节能,但丢失了数据中大量的原始结构,只提供粗糙的统计量,掩饰了令人感兴趣的局部变化;使用小波概要(wavelet synopses)、分段逼近、直方图等的节能效果不如聚集函数,但可以更好地恢复出近似原始数据.

传感器节点持续产生的数据随时间流逝而变化,可以直观表示为一条平面曲线.平面曲线的逼近表示在模式识别、图像处理、图形学等领域非常重要,可达到特征提取、数据压缩、降噪等目的.除非曲线足够简单,否则多项式逼近或级数展开逼近效果不好,出现 Runge 现象.分段逼近可以较好地解决这个问题.数值分析中的分段插值不但要求在插值节点上函数值相等,还要求导数值也相等(Hermite 插值),甚至要求高阶导数也相等(样条插值).我们通过低阶分段拟合来进行函数逼近,段数远小于采样数据的个数,而每段只需 2 个或 3 个数据来表示,实现有损数据压缩;不要求相邻各段连续,也不要求采样数据与逼近数据完全相等.

1 相关工作

时间序列的近似表示本质上就是降维,同时要保证误差有界,已有较丰富的研究成果.由于逼近子空间的不同构造(也就是基的选择或构造)以及范数的不同选取,形成了不同的数值逼近方法.根据逼近子空间的不同分为两类:① 使用分段不连续函数来逼近,包括离散小波变换(DWT)[3]、分段线性逼近(piecewise linear approximation,简称 PLA)[4,5]、分段常量逼近(piecewise constant approximation,简称 PCA)[6,7]、分段聚集逼近(piecewise aggregate approximation,简称 PAA)[8]、符号化的聚集近似(symbolic aggregate approximation,简称 SAX)[9].时间序列经过 DWT 变换后得到一个等长的小波系数序列,由于小波变换可以集中能量、消除相关性,因此小波系数序列中大部分细节分量值很小.精心挑选小波系数的一个子集,得到小波概要,以后可根据小波概要快速响应用户查询,并保证查询结果的精度.PLA 用分段线性函数(即若干线段)来表示时间序列.PAA 将包含 n 个数据的时间序列等分为 p 段($p<<n$),每段长度均为 n/p,用落在此段中数据的平均值代表本段.与 PAA 类似,PCA 也是用一系列段来近似时间序列,每段用段内数据的均值表示,但是 PCA 各段的段长是可变的.SAX 是时间序列的第 1 种符号化表示,SAX 首先计算时间序列的 PAA 表示,然后对 PAA 系数进行量化,最后将每个量化级用单个字符表示;② 使用低阶连续函数来逼近,包括奇异值分解(SVD)、离散傅立叶变换(DFT)、样条函数、非线性回归、Chebyshev 多项式.对传感器网络而言,SVD 计算开销太大.DFT 和 Chebyshev 多项式逼近类似于 DWT,都是用正交函数作为基进行函数(即时间序列)展开和逼近,不同在于基函数的选择.根据众多前人的研究,没有哪种方法在不同应用领域性能都优秀.

数据压缩就是要消除数据间的冗余,即相关性.某个传感器节点采集的某个物理量在某段时间内的数据之间具有相关性(时间相关性);在某一时刻,某个传感器节点采集的不同物理量属性数据之间具有相关性(多属性相关性);在某一时刻,地理上较近的不同传感器节点采集的某个物理量数据之间具有相关性(空间相关性).数据

的相关性越高,数据压缩的效果越好.

采用分段逼近消除数据空间相关性的论文我们暂未见到,因为一般采用分布式压缩算法降低数据的空间相关性.而综合能耗、精度、延时等因素考虑,分段逼近不适合于分布式计算.WSN 中,通常将空间邻近的节点组织成簇,这样,簇头收集到的传感数据集便存在较强的空间相关性.对存在空间相关性的传感数据集进行小波变换,不仅可以提高低频小波系数的能量聚集比,还提高数据编码的效率,使得传感数据集的压缩比得以增加.Ciancio 等人[10]研究小波压缩的分布式算法,通过在邻近的节点间交换信息,在数据传送到汇聚节点前分布式挖掘网络中数据的空间相关性,极大地减少了冗余数据的传输.虽然分布式压缩算法有效减少了网络中冗余数据的传输,然而节点间需要交换信息,由此产生的能量消耗、网络延时等代价尚需要在理论上进行进一步的定性分析.我们针对任意支撑长度的小波函数,给出了一种基于环模型的分布式时-空小波数据压缩算法[11],该算法可以同时消除传感器网络中数据的时间和空间相关性.实验表明,分布式二级小波变换的性能略优于分布式一级小波变换,因为虽然进行第 2 级小波变换会增加额外的耗能与延时,却取得了更好的去相关性.分布式小波变换[10,11]中,不同传感器节点间的同级变换可以并行,但各级变换之间必须串行.随着分解级数的提高,数据的空间相关性在降低.因此,分布式算法不能总是依靠提高小波分解级数来提高性能,那会不断增加通信开销和延时.分布式压缩算法都建立在节点之间具有空间相关性的假设之上,且实现代价较大.如果空间相关性低甚至不存在,分布式算法几乎没有压缩效果反而有实现开销,得不偿失.另一方面,在人工放置传感器节点时,希望用最少的节点监测最大的范围,通常使节点的放置尽量彼此独立.例如,在智能大厦中,每个房间装有温度传感器,由于每个房间使用状况不同(空调的开关、门窗的开闭、人员的多少),相邻房间内温度也有较大差别,即监测值不具有空间相关性.因此,当节点监测值之间不存在空间相关性或空间相关性不稳定时,我们设计在各节点上独立运行的算法是更好的选择.

采用分段逼近来消除数据多属性相关性的核心思想是,选取某属性为基础信号,与之相关度较大的其他信号用分段逼近来表示.Deligiannakis[12]最先研究了多维数据流的多属性相关性.时间序列本身不是线性的,但不同序列在一段时间内可能存在线性相关.他提出的 SBR 算法将根据同一传感器节点上所有序列的数据分布特征挑出的基信号为自变量,用线性回归模型分段逼近其他序列.原始的多个时间序列可以表示为基信号加上一些回归参数.此方案能够消除多属性相关性和时间相关性,但仍有缺陷:① 需要不断将基信号更新传送到基站;② 压缩过程中未考虑数据误差,只是在满足数据压缩的条件下最大程度地压缩数据,没有保证压缩结果重构后误差有界;③ 回归参数的求解是基于 L_2 误差的.Gandhi 提出的 GAMPS[7]框架同时对多个数据流进行压缩,可确保重构出的数据 L_∞误差有界.GAMPS 根据各个数据流的相关性进行动态分组,相关大的数据流放在同一组以获得最佳压缩效果.每个组内挑选出一个基信号,采用 PCA 近似表示;其他信号除以基信号得到比例信号(ratio signal).比例信号比较平坦,再用 PCA 逼近可获得较高压缩率.但是,GAMPS 不是在线算法,只适用于基站或数据中心对传感器数据进行存档处理,不适用于将数据从传感器收集到基站.GAMPS 的静态/动态分组算法被转化为设施选址问题,而这是一个 NP 完全问题.我们提出了基于自适应回归的误差有界的多属性数据压缩算法[13],自适应指该算法可根据误差限和压缩收益自动选择传输原始数据还是传输回归系数,自动确定每次参与回归计算的数据个数.当多属性相关性减小或不稳定时,其压缩效果也很理想.

采用分段逼近来消除数据时间相关性是研究最广泛的[4−6].在许多应用中,用户获得精确值成本太高.设计单遍扫描算法、能实时给出近似查询结果就成为数据流模型下数据处理的目标.Lazaridis 等人[6]采用 PCA 来逼近传感器数据,提出了确保重构 L_∞误差有界的 PMC-MR 和 PMC-MEAN 两个在线压缩算法.PMC-MR 是段数最少的最优分段方法,但它不易应用于相似度检索系统中,且当数据分布偏向中值线某一边时,平均误差较大;PMC-MEAN 减少了平均误差,但使用了更多的段.Soroush 等人[5]采用分段线性逼近传感器数据,定义了确保重构 L_∞和 L_2 误差有界的两个问题(PLA-PointBound 和 PLA-SegmentBound),巧妙地提出对应的在线压缩算法.但 PLA-PointBound 问题需要求各线段包含原始数据的个数个多边形的交集,计算开销太大.

本文采用 2 范数和无穷范数作为误差度量,研究计算简单、误差有界的分段逼近算法.主要工作是:① 利用传感器节点内置缓冲区,提出了单节点上基于分段常量逼近的准在线数据压缩算法(PCADC-sensor),给出了 L_∞

误差有界的一种实现.在这种方式下,计算 PLA 的时间复杂度较大,故采用 PCA;② 提出了单节点上基于分段线性逼近的在线式数据压缩算法(PLADC-sensor),推导了分段线性一致逼近的充要条件.分别在 L_∞和 L_2 误差度量下给出了计算 PLA 的两种简单快速算法,而没有采用逼近效果平均稍差的 PCA;③ 簇头或基站不需要接收其他传感器节点的原始采样数据,提出了直接基于分段线性表示(PLR)的数据压缩算法(PLRDC-cluster),推导了同一节点不同时间段、不同节点相同时间段这两种情况下的计算公式;④ 模拟实验表明,算法能够显著减少冗余数据的传输,比分布式算法具有更好的实时性和更低的开销.

2 预备知识

2.1 数据流分段逼近模型

设某个传感器节点采集的某个物理量的数据流为 $s[0],\ldots,s[N-1],\ldots$,它是持续到达、潜在无限的.对这些数据进行分段,每段用一个常数(如均值)或一根线段(如斜率、截距)来表示.在保证每段内原始数据与逼近数据的误差小于预定误差限的条件下,在线式输出各段,从而极大地减少待传输的数据量.

考察某段时间内采集的 N 个数据为 $S=(s[0],\ldots,s[N-1])$,分段重构出的近似数据为 $\tilde{S}=(\tilde{s}[0],...,\tilde{s}[N-1])$,记误差 $e[i]=s[i]-\tilde{s}[i]$,$e=(e[0],\ldots,e[N-1])$.

定义 1(2-范数误差). $L_2(S,\tilde{S})=\|e\|_2=\sqrt{\sum_{0\leqslant i<N}e[i]^2}$.$L_2$ 误差体现了两个向量间的整体累积误差,也叫 Euclidean 距离.L_2 误差随数据个数(向量维数)的增加单调非递减.对于给定 L_2 误差限,评价重构数据逼近原始数据的近似程度时,还与时间窗口大小有关.我们用分段逼近数据流时,要求每段的 L_2 误差满足误差限要求.

定义 2(2-范数平均误差). $\bar{L}_2(S,\tilde{S})=\sqrt{\left(\sum_{0\leqslant i<N}e[i]^2\right)\Big/N}$.$\bar{L}_2$ 误差体现了两个向量间的平均累积误差,消除了数据个数的影响,在实际应用中较有意义.

定义 3(∞-范数误差). $L_\infty(S,\tilde{S})=\|e\|_\infty=\max_{0\leqslant i<N}|e[i]|$.$L_\infty$误差限保证了每个重构出来的数据与对应原始采样之间的误差有界.

数据流误差有界的分段逼近,要求原始数据和重构数据之差在某种度量意义下有上界.相似度越大,误差越小,距离越短.采用 L_2 误差作为距离度量时,称为平方逼近或均方逼近;采用 L_∞误差为距离度量时,称为一致逼近或均匀逼近.在实际应用中,一致逼近更有意义.

2.2 数据流分段算法

数据流分段算法可分为批处理(batch)方式和在线(online)方式两类:在线方式的逼近效果相对不好,但只要对数据顺序扫描一次;批处理方式可以产生高质量的逼近,但需要多次扫描整个数据集.各种资源都有限的传感器节点无法存储所有数据,采用在线方式更好.已有的数据流分段算法可以分为 3 类[4]:

(1) 滑动窗口分段(sliding window).过程是:① 将数据流的第 1 个数据作为第 1 个段的最左点,此时,第 1 个段的长度为 1;② 计算将该段右边相邻的 1 个数据纳入该段后的拟合误差,如果小于误差限,段长增加 1.继续考察该段右边相邻的数据,不断向右增加该段的长度.当到第 i 个数据点时,拟合误差超过了误差限,则第 1~第 i–1 个数据构成了第 1 段;③ 第 2 段将第 i 个数据作为最左点,段长为 1,类似地不断执行步骤②,依次得到后面各段.这种方法简单、直观,是一种在线算法.值得指出的是,当得到某个段以后,对应的数据可以马上释放,适用于传感器节点等存储空间小的情况.

(2) 自顶向下分段(top-down).这种方法逼近效果好、时间复杂度高,是一种批处理算法.过程是:① 将全部数据看成 1 段,依次考察将其分成 2 段的各种可能的划分方法,将其在最佳位置分成 2 段;② 分段后,分别测试这 2 个子段的拟合误差是否小于用户预定义的误差限.如果都满足误差限,则算法结束;否则,递归地将相应子段继续按步骤①再划分成 2 段,直到所有最新划分出的子段都满足误差限.

(3) 自底向上分段(bottom-up).这是自顶向下分段的逆过程:① 对长度为 N 的时间序列建立全部数据的最

精确的逼近,即建立 N 个段(PCA)或(N+1)/2 个段(PLA);② 依次计算序列中顺序相连的两个段如果合并的拟合误差,挑选出最小的合并误差;③ 若最小误差小于预定误差限,则将对应的相连两段进行合并,合并后的新段替换原来的两个段,计算新段与其前后段如果合并的拟合误差,再转步骤②;否则,算法结束.

3 单传感器节点上基于分段常量逼近的数据压缩算法

滑动窗口分段只考虑当前数据,没有纵观全局的能力,压缩效果较差.随着技术的发展,出现了各方面性能都更强大的传感器节点,存储能力有了显著增强,如 Crossbow 公司的 Imote2 节点有 32MB 数据存储空间.节点上可以缓存最近一段时间采集的大量数据,从而可以多次访问这些数据,取得比单纯滑动窗口分段更好的分段逼近效果.我们提出了单传感器节点上基于分段常量逼近的误差有界的数据压缩算法 PCADC-sensor (piecewise constant approximation based data compression algorithm with error bound in sensor node).PCADC-sensor 利用传感器节点所带缓冲区数据进行自底向上分段,但是增量式计算的,是一种准在线算法.虽然在这种方式下也可用 PLA 逼近,但反复计算 PLA,时间复杂度较大,故采用 PCA.

采用双循环队列来组织节点上采样数据缓冲区数组 S,如图 1 所示,S 大小为 *arraysize*.定义队头指针指向队列首元素,队尾指针指向队列尾元素的下一个位置.正在进行分段处理的数据为以 *datafront* 为队头指针、*datarear* 为队尾指针的处理队列(processing queue).在分段计算的同时,传感器又并发采集了若干数据,为以 *datarear* 为队头指针、*readin* 为队尾指针的缓冲队列(buffering queue).带缓冲区的分段算法的基本框架是:① 最开始,两个队列皆空,*datafront*=*datarear*=*readin*,以后两个队列也可以通过队头指针等于队尾指针来判定队空;② 随着采样的进行,*readin* 指针不断循环后移.当采集到 *arraysize*/4 个数据后,将 *readin* 赋值给 *datarear*,相当于把缓冲队列里的全部数据传到了处理队列;③ 处理队列中的数据按自底向上分段(见算法 1),以取得更好的逼近效果;④ 缓冲队列可并发继续存储新来采样.如果(*readin*+1)%*arraysize* 等于 *datafront*,则说明整个数组空间已满,强制转步骤⑤;⑤ 发送处理队列中计算出的第 1 个分段的表示数据,将对应采样数据清除出处理队列.在保证处理队列长度不超过 *arraysize*/2 的条件下,将缓冲队列中尽量多的数据纳入处理队列;⑥ 若处理队列长度小于 *arraysize*/4,转步骤②.重复以上步骤③~步骤⑥.

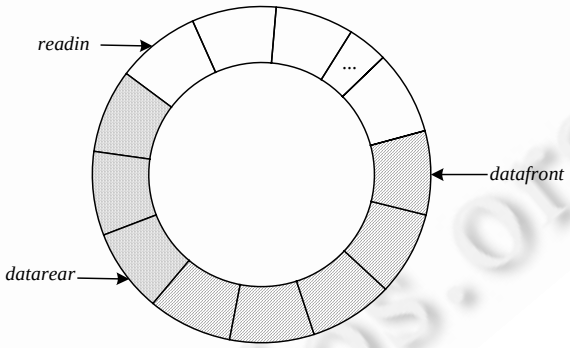

Fig.1 Data buffer organization of sensor node (double circular queue)

图 1 传感器节点的采样数据缓冲区(双循环队列)

数组空间大小、分段处理速度、采样频率将影响本算法性能.数组空间大的好处有:① 处理队列越长,越接近纯批处理方式的自底向上分段,压缩效果越好;② 当数据变化缓慢总是满足误差限时,整个处理队列中数据合并后只有 1 段,输出后希望缓冲队列中的数据也已经足够多,从而达到理想的压缩效果.上述算法保证了处理队列长度属于[*arraysize*/4,*arraysize*/2],缓冲队列有机会存放新来的采样.段长越长,数组空间就要越大,一般取平均段长的 12~20 倍为 *arraysize*.数组空间大的缺点有:① 处理队列越长,造成分段速度越慢,从而可能导致缓冲队列溢出,故采用了计算相对简单的 PCA;② 数组里数据越多,接收方得到数据的时延越大.易知,设 a 为待压

缩时间序列的长度,n 为处理队列的长度,l 为段的平均长度,则共有 $O(a/l)$段,而算法 1 的时间复杂度为 $O(n*l)$,则 PCADC-sensor 算法时间复杂度为 $O(a*n)$.

基于上述带缓冲区的分段算法基本框架,我们研究分段结果满足 L_∞误差有界的算法.算法 1 实现了一致逼近的分段.处理队列中包含的数据个数为(*datarear*−*datafront*+*arraysize*)%*arraysize*,记为 N.处理队列中的段用 SEG 类型表示是合并计算的需要;每次输出的是第 1 段,这个段的 *ts*[0].*count* 个原始数据均用 *ts*[0].*mean* 近似表示.真正输出时,用两个数据即可表示一个段.

算法 1. 处理队列中数据的分段算法.

Input:无穷范数误差限 *eps*,缓冲区数组 *s*;

Output:分段常量表示 *ts*.

```
typedef struct Seg{
   float mean;     //用本段包含的数据点的平均值代表本段
   int count;      //本段中数据点的个数
   float max,min; //本段中数据点的最大值、最小值
} SEG;
SEG ts[N];         //处理队列中有 N 个数据,将用若干常量段表示
int ts_length=0;  //段的个数
```

1. 将处理队列中的 N 个数据划分成满足误差限的 M 段,每段包含 1 个或相邻的 2 个数据,即有 $M\geqslant(N+1)/2$.
2. 假设将相邻两段合并成新段,依次计算所有 $M-1$ 种可能的新段所包含的数据个数、是否满足误差限.

   //假设将段 *ts*[*i*]和 *ts*[*i*+1]合并,描述新段的有关信息为:

   //新段覆盖的数据个数 *count*=*ts*[*i*].*count*+*ts*[*i*+1].*count*;

   //新段的均值 *mean*=(*ts*[*i*].*count***ts*[*i*].*mean*+*ts*[*i*+1].*count***ts*[*i*+1].*mean*)/*count*;

   //新段的最小数据值 min=(*ts*[*i*].min<*ts*[*i*+1].min)?*ts*[*i*].min:*ts*[*i*+1].min;

   //新段的最大数据值 max=(*ts*[*i*].max>*ts*[*i*+1].max)?*ts*[*i*].max:*ts*[*i*+1].max;

   //如果|*mean*−min|<*eps* 并且|*mean*−max|<*eps*,则将段 *ts*[*i*]和 *ts*[*i*+1]合并后满足误差限,可以合并.
3. 选择满足误差限且所包含的数据个数最多的新段 *k*,将其对应的原来的两段 *ts*[*k*]和 *ts*[*k*+1]真正合并.

   //将代表新段的 SEG 类型数据保存在 *ts*[*k*]中;

   //删除 *ts*[*k*+1],将 *M* 的值减 1.
4. 重复步骤 2、步骤 3,直至满足误差限的合并都已完成.

SWAB 算法[4]先用滑动窗口分段得到一些段,再用分段性能更佳的自底向上分段对它们进行半全局的合并.SWAB 算法将“数据段”进行再合并,但如果这些分段的误差已经接近误差限,合并很难再进行,且这些分段不一定是原来数据的最佳分段方式.通过设计缓冲区管理机制,我们的算法是将“数据点”进行合并.

4 单传感器节点上基于分段线性逼近的数据压缩算法

为消除数据时间相关性,我们基于滑动窗口分段,提出了单传感器节点上基于分段线性逼近的误差有界的在线式数据压缩算法 PLADC-sensor(piecewise linear approximation based data compression algorithm with error bound in sensor node).范数的不同选取,形成了如下两种不同的数值逼近方法.

4.1 满足L_∞误差限的一致逼近

一致逼近比平方逼近更难解决,即传统的 Chebyshev 逼近问题.现有如采用线性规划的分段方法都是静态的批处理方式,关注每段的最大误差点,需要反复迭代,不适合数据流.我们的算法以一种贪婪的在线方式进行,每段尽可能向右多包含一些数据.记直线方程为 $y=k*x+b$,其中,k 为斜率,b 为截距.

设传感器在时刻 t_i 采集的数据为 s_i,即$(t_1,s_1),(t_2,s_2),\ldots,(t_i,s_i)$,考虑到采样时间间隔相等,则不妨记为$(1,s_1)$,

$(2,s_2),\ldots,(i,s_i)$.每个数据点允许的误差限为 eps,则 s_1 在拟合直线上的近似值 $\tilde{s}_1 \in [s_1-eps, s_1+eps]$,如图 2 所示.当数据流中第 1 个数据 s_1 到达时,保存它,它是本段的第 1 个数据.当第 2 个数据 s_2 到达时,根据 s_1 和 s_2 计算满足误差限的直线的斜率区间 k_{12},在纵坐标上截距的区间 b_{12}.当第 i 个数据 s_i 到达时,根据 s_1 和 s_i 计算满足误差限的直线的斜率区间 k_{1i},在纵坐标上截距的区间 b_{1i}.

记 $k_{1i}=[k_i^{\min},k_i^{\max}]$,易知 $k_i^{\min}=((s_i-eps)-(s_1+eps))/(i-1), k_i^{\max}=((s_i+eps)-(s_1-eps))/(i-1)$.

例如,k_{12}=[$s_2-s_1-2*eps, s_2-s_1+2*eps$],如图 2 所示,区间端点分别为经过 A 点和 B 点的直线的斜率.

记 $b_{1i}=[b_i^{\min},b_i^{\max}]$,因为直线的斜率最大(小)时截距最小(大),则,

$$b_i^{\min}=(i(s_1-eps)-(s_i+eps))/(i-1), b_i^{\max}=(i(s_1+eps)-(s_i-eps))/(i-1).$$

例如,b_{12}=[$2s_1-s_2-3\times eps, 2s_1-s_2+3\times eps$],如图 2 所示,区间端点分别为经过 B 点和 A 点的直线的截距.

易知,$(1,s_1),(2,s_2),\ldots,(i,s_i)$满足要求的逼近直线存在的必要非充分条件是 $\bigcap_{j=1}^{i} k_{1j}\neq\varnothing$ 或 $\bigcap_{j=1}^{i} b_{1j}\neq\varnothing$.

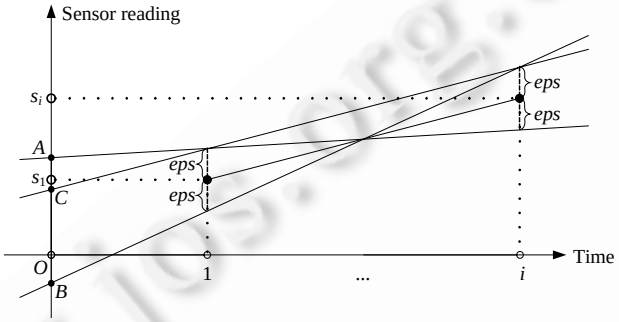

Fig.2 Ranges of possible slope and intercept

图 2 直线斜率和截距的可能范围

下面分析满足要求的逼近直线存在的充要条件.实际上,前述计算 b_{1i} 上下界的方法是很宽松的,是任意 $\tilde{s}_1\in[s_1-eps,s_1+eps]$ 对应的截距区间的并集.对于某条具体的直线,$\tilde{s}_1$ 只能取一个值,其截距区间并没有那么大.图 2 表明了根据 s_1 和 s_i 计算逼近直线的情况,根据前述 b_{1i} 的计算公式,其截距区间为 BA;但如果 $\tilde{s}_1$ 具体取值为 s_1+eps,其截距区间仅为 CA.

我们先将 $\tilde{s}_1$ 固定视为常数,根据 $\tilde{s}_1$ 和 s_i 计算满足误差限的直线的斜率范围 $nk_{1i}=[nk_i^{\min},nk_i^{\max}]$,其中,

$$nk_i^{\min}=((s_i-eps)-\tilde{s}_1)/(i-1), nk_i^{\max}=((s_i+eps)-\tilde{s}_1)/(i-1).$$

因直线经过点 $(1,\tilde{s}_1)$,则可算出对应的截距范围 $nb_{1i}=[nb_i^{\min},nb_i^{\max}]$,其中,

$$nb_i^{\min}=(i*\tilde{s}_1-s_i-eps))/(i-1), nb_i^{\max}=(i*\tilde{s}_1-s_i+eps)/(i-1).$$

因此,对 s_1 和 s_i 而言,满足以下 4 个不等式就可得到满足误差限的所有直线:

$$s_1-eps\leqslant\tilde{s}_1\leqslant s_1+eps \tag{1}$$

$$nb_i^{\min}\leqslant b\leqslant nb_i^{\max} \tag{2}$$

以 $\tilde{s}_1$ 为横轴,对应的直线截距 b 为纵轴,得到图 3.平行四边形里的任何一点代表一条对 s_1 和 s_i 而言符合误差要求的逼近直线,将这个平行四边形记为 $poly(1,i)$.注意,图 2 中的 A,B,C 与图 3 中的是相互对应的.

公式(1)对应了图 3 中平行四边形的两条与横坐标轴垂直的边,所有平行四边形 $poly(1,j)$(其中,j=2,…i)的这两条边都共线.公式(2)对应了两条斜边,$poly(1,j)$斜边的斜率为 $j/(j-1)$,即所有平行四边形斜边的斜率为(1,2]中的某个值,且依次减少.

定理 1(分段线性一致逼近的充要条件). 通过点 $(1,\tilde{s}_1)$ 和点$(0,b)$的直线是数据流 $s_1,s_2,\ldots,s_i$ 在 eps 误差限下的一致逼近(图 2)当且仅当点 $(\tilde{s}_1,b)$ 属于 $\bigcap_{j=2}^{i} poly(1,j)$ (图 3).

证明:(必要性⇒).证明其等价的逆否命题即可.对于任意 $(\tilde{s}_1,b)\notin\bigcap_{j=2}^{i} poly(1,j)$,则存在一个 $k(2\leqslant k\leqslant i)$,有 $(\tilde{s}_1,b)$ 不属于 $poly(1,k)$.根据 $poly(1,k)$的定义,即通过点 $(1,\tilde{s}_1)$ 和点$(0,b)$的直线将不能在误差限下逼近 s_1 或 s_k.

(充分性⇐)用反证法.假设 $(\tilde{s}_1,b)\in\bigcap_{j=2}^{i} poly(1,j)$,但通过点 $(1,\tilde{s}_1)$ 和点$(0,b)$的直线不能逼近 $s_k(1\leqslant k\leqslant i)$.此时有两种情况:如果 $k=1$,即 $|\tilde{s}_1-s_1|>eps$,则有 $(\tilde{s}_1,b)$ 不属于 $poly(1,2)$;如果 $k\neq 1$,则 $(\tilde{s}_1,b)$ 不属于 $poly(1,k)$.这两种情况都与假设矛盾,即假设不成立. □

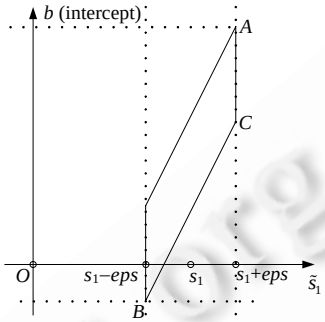

Fig.3 Parallelogram $poly(1,i)$ denoting possible domain of $(\tilde{s}_1,b)$ when s_1 and s_i meet the uniform approximation

图 3 s_1 和 s_i 满足一致逼近时 $(\tilde{s}_1,b)$ 的合法范围记为 $poly(1,i)$

根据定理 1 和第 2.2 节滑动窗口分段的框架,我们就可以得到一个各分段不连续的数据压缩算法.即寻找满足 $\bigcap_{j=2}^{i} poly(1,j)\neq\varnothing$ 且 $\bigcap_{j=2}^{i+1} poly(1,j)=\varnothing$ 的 i,将 s_1 到 s_i 压缩成一段.可以用数学归纳法证明这种方式产生的段数是最少的.但这个需要计算平行四边形的交集,计算量仍然不小.算法 2 给出了一种近似算法,我们记其为 PLADC-sensor-inf 算法.这个基于斜率的分段线性逼近算法求出的逼近直线满足 L_∞误差限要求,是充分非必要条件.其时间复杂度为 $O(n)$,空间复杂度为 $O(1)$.其中,n 为待压缩时间序列的长度.

算法 2. 基于斜率的分段线性逼近.

Input:无穷范数误差限 *eps*;

Output:分段线性表示 k,(i,s_i).

//根据 s_i 和 s_j 计算满足误差限的直线的斜率范围 $nk_{ij}=[nk_j^{\min},nk_j^{\max}]$,其中,$i<j$

//其中, $nk_j^{\min}=((s_j-eps)-s_i)/(j-i), nk_j^{\max}=((s_j+eps)-s_i)/(j-i)$

1. 读第 1 个数据 s_1,保存 s_1,将 s_1 作为第 1 段的第 1 个元素;
2. 读第 2 个数据 s_2,计算 $k=nk_{12}$,丢弃数据 s_2;
3. $i=1$; $j=2$ //满足误差限的直线起点为 i,终点为 j;
4. While (1){

   j++;

   读第 j 个数据 s_j,保存数据 s_j,计算 nk_{ij},计算 $k1=k\cap nk_{ij}$;

   If ($k1$!=∅) {丢弃数据 s_j; $k=k1$;}

   Else {

      取斜率区间中点值为斜率 k,经过本段起点(i,s_i)的直线作为当前段输出; //即每个段用 3 个数据表示

      $i=j$; //新段起点的序号

      j++;

      读第 j 个数据 s_j,计算 $k=nk_{ij}$,丢弃数据 s_j;

}

}

4.2 满足L_2误差限的平方逼近

我们求 $y=k\times x+b$,使$(1,s_1),(2,s_2),\ldots,(i,s_i)$对应的 L_2 误差满足误差限 *eps*,且 *i* 尽可能地大.这是标准的一元线性回归,可采用众所周知的最小二乘法来求解.

对点(j,s_j)而言,代表 *j* 时刻传感器读数为 s_j,其用分段直线逼近的估计值为 $\tilde{s}_j=k*j+b$.则 L_2 误差的平方为 $f=(\sum_{j=1}^{i}(k*j+b-s_j))^2$,当 $\frac{\partial f}{\partial b}=0$ 且 $\frac{\partial f}{\partial k}=0$ 时,误差取最小值,即 $\begin{cases}\sum_{j=1}^{i}(k*j+b-s_j)=0\\ \sum_{j=1}^{i}[(k*j+b-s_j)*j]=0\end{cases}$,解方程组:

$$k=\frac{i*\sum_{j=1}^{i}(j*s_j)-\sum_{j=1}^{i}j*\sum_{j=1}^{i}s_j}{i*\sum_{j=1}^{i}j^2-(\sum_{j=1}^{i}j)^2},b=\frac{\sum_{j=1}^{i}s_j-k*\sum_{j=1}^{i}j}{i} \tag{3}$$

当按公式(3)计算出 *k* 和 *b* 时,*f* 取到最小值.注意,$\bar{L}_2$ 误差也取到最小值.为便于计算,将 *f* 展开为

$$f=\sum_{j=1}^{i}((k*j-s_j)+b)^2=\sum_{j=1}^{i}[k^2j^2-2kjs_j+s_j^2+2(k*j-s_j)b+b^2]$$

$$=k^2\sum_{j=1}^{i}j^2-2k\sum_{j=1}^{i}(j*s_j)+\sum_{j=1}^{i}s_j^2+2b\sum_{j=1}^{i}(k*j-s_j)+i*b^2$$

又因为取最小值时有 $\sum_{j=1}^{i}(k*j+b-s_j)=0$,即 $\sum_{j=1}^{i}(k*j-s_j)=-i*b$,代入上式得到:

$$f_{\min}=k^2\sum_{j=1}^{i}j^2-2k\sum_{j=1}^{i}(j*s_j)+\sum_{j=1}^{i}s_j^2-i*b^2 \tag{4}$$

随着时间的流逝,依次计算点集$\{(1,s_1),(2,s_2),\ldots,(i,s_i)\}(i=3,4,5,\ldots)$的拟合能否满足误差限,得到第 1 段.注意,公式(3)、公式(4)可以采用递推的方式进行计算,极大减少了计算量.算法 3 是采用滑动窗口方式进行分段线性平方逼近的算法,我们记其为 PLADC-sensor-2 算法.采取算法 3 中的递推计算方式,每采集一个新数据点,在 *O*(1)的时间内可以判定出是否可以加入已有线段.整个算法的时间复杂度为 *O*(*n*),空间复杂度为 *O*(1).

算法 3. L_2 误差有界的分段线性逼近.

Input:2 范数误差限 *eps*(实际使用中采用定义 2 较好);

Output:分段线性表示.

//定义 $sum\_j(x,y)=\sum_{j=x}^{y}j, sum\_jj(x,y)=\sum_{j=x}^{y}j^2, sum\_s(x,y)=\sum_{j=x}^{y}s_j, sum\_js(x,y)=\sum_{j=x}^{y}(j*s_j)$,

//定义 $sum\_ss(x,y)=\sum_{j=x}^{y}s_j^2$

1. 读入第 1 个数据 s_1,保存;
2. 读入第 2 个数据 s_2;
3. 计算 *sum_j*(1,2),*sum_jj*(1,2),*sum_s*(1,2),*sum_js*(1,2),*sum_ss*(1,2),
   将它们分别保存在变量 *sum_j*,*sum_jj*,*sum_s*,*sum_js*,*sum_ss* 里;
4. 计算通过$(1,s_1),(2,s_2)$两点的直线的斜率 *old_k*、截距 *old_b*,丢弃数据 s_2;
5. *m*=1; *i*=2; //满足误差限的直线起点为 *m*,终点为 *i*;
6. While (1){
   i++;
   读入第 *i* 个数据 s_i,保存数据 s_i;
   sum_j+=*i*; *sum_jj*+=*i**i*; *sum_s*+=s_i; *sum_js*+=*i*s_i;
   根据公式(3)的递推形式 $k=\frac{(i-m+1)*sum\_js-sum\_j*sum\_s}{(i-m+1)*sum\_jj-sum\_j*sum\_j}, b=\frac{sum\_s-k*sum\_j}{i-m+1}$,计算出 *k*,*b*;
   sum_ss+=s_i*s_i;
   根据公式(4)的递推形式 $f=k^2*sum\_jj-2k*sum\_js+sum\_ss-(i-m+1)*b^2$,计算 L_2 误差的平方 *f*;

实际使用中,误差采用定义 2 较有物理意义,即 f 还需要除以本段对应的数据个数($i-m+1$)后再开平方;
If ($f<eps$) {丢弃数据 s_i;$old\_k=k$;$old\_b=b$;}
Else {
取 old_k 为斜率、old_b 为截距,开始时间为 m 的直线作为当前段输出; //即每个段用 3 个数据表示
丢弃数据 s_m; //已经输出段的起点
$m=i$; //新段起点的序号
i++;
读入第 i 个数据 s_i;
计算 $sum\_j(m,i)$,$sum\_jj(m,i)$,$sum\_s(m,i)$,$sum\_js(m,i)$,$sum\_ss(m,i)$,
将它们分别保存在变量 sum_j,sum_jj,sum_s,sum_js,sum_ss 里;
计算通过(m,s_m),(i,s_i)两点的直线的斜率 old_k、截距 old_b,丢弃数据 s_i;
}
}

5 簇头或基站上基于分段线性表示的数据压缩算法

传感器节点采用 PLADC-sensor 算法得到其采集数据的分段线性表示(PLR),除用于数据压缩外,还可广泛用于数据挖掘.簇头或基站利用来自多个节点的 PLR 直接进行进一步的数据压缩,也可计算 SUM,AVG 等聚集函数值,不需要原始的传感数据.我们提出了簇头或基站上基于分段线性表示的误差有界数据压缩算法 PLRDC-cluster(piecewise linear representation based data compression algorithm with error bound in cluster head).算法 2 用斜率 k、起点(i,s_i)表示一段;算法 3 用斜率 k、截距 b、开始时间 m 表示一段.这两种 PLR 表示是等价的,由一种表示形式可以导出另一种.下面采用第 2 种 PLR 来表示数据流.

5.1 相同节点不同时间段内平方逼近的数据压缩

记某传感器原始数据流$(1,s_1),(2,s_2),\ldots,(i,s_i),\ldots$,已采用上节算法表示为$(t_1,k_1,b_1),\ldots,(t_i,k_i,b_i),\ldots$.其中,$t_i$ 表示第 i 段的开始时间,k_i 表示第 i 段的斜率,b_i 表示第 i 段的截距.(t_i,k_i,b_i)为段 S_i 的 PLR.

已知 $S_1,S_2,\ldots,S_j$ 共 j 个段的 PLR 表示,现用一个段作为$[t_1,t_{j+1})$时间内这个传感器数据的更粗糙的 PLR 表示.由于簇头或基站上是按时间顺序接收 S_j 的,采取将新来段不断归并到已有段的方式,我们只要研究由相邻两小段 S_1,S_2 的 PLR 计算出合并后大段 S_{12} 的 PLR 的算法即可.注意,此时无法访问原始的采样数据 s_i.

方法 1:基于 S_1,S_2 两段对应的原始采样数据,最小化 L_2 误差进行线性拟合,得到新大段.记近似值 $\tilde{s}_i=k*i+b$,过去有 W 次采样 $s_i,i\in[t_b,t_e],W=t_e-t_b+1$,拟合成一条直线,参考平方逼近现有计算公式[14]:

$$k=\sum_{i=t_b}^{t_e}\left(\frac{i-\bar{t}}{v_t}\right)(s_i-\bar{s}) \tag{5}$$

$$b=\bar{s}-k*\bar{t} \tag{6}$$

其中,时间平均值 $\bar{t}=\left(\sum_{i=t_b}^{t_e}i\right)\Big/W=(t_b+t_e)/2$,采样平均值 $\bar{s}=\left(\sum_{i=t_b}^{t_e}s_i\right)\Big/W$,$v_t=\sum_{i=t_b}^{t_e}(i-\bar{t})^2$.

为便于说明,段 S_1 用 (t_b^1,t_e^1,k_1,b_1) 表示,段 S_2 用 (t_b^2,t_e^2,k_2,b_2) 表示,合并后的大段 S_{12} 用(T_b,T_e,K,B)表示.

段 S_1 的采样个数、时间平均值、采样平均值分别用 $W_1,\bar{t}_1,\bar{s}_1$ 表示;类似地,段 S_2 的为 $W_2,\bar{t}_2,\bar{s}_2$,段 S_{12} 的为 W_{12},$\bar{T},\bar{S}=(\bar{s}_1*W_1+\bar{s}_2*W_2)/(W_1+W_2)$.显然,$[T_b,T_e]=[t_b^1,t_e^2]$.下面研究 K,B 的计算.

将 v_t 展开计算,利用公式 $\sum_{i=1}^{n}i^2=\frac{1}{6}n(n+1)(2n+1)$,可得 $v_t=\frac{1}{12}(W^3-W)$.即,其值与具体时间无关,只取决于时间窗口的大小.对段 S_{12} 而言,将 $W_{12}=t_e^2-t_b^1+1$ 代入即可计算出 v_t.根据公式(5),考察段 S_{12}:

$$K=\sum_{i=t_b^1}^{t_e^2}\left(\frac{i-\overline{T}}{v_t}\right)(s_i-\overline{S})=\sum_{j=1}^{2}\sum_{i=t_b^j}^{t_e^j}\left(\frac{i-\overline{T}}{v_t}\right)(s_i-\overline{S})=\sum_{j=1}^{2}\left\{\sum_{i=t_b^j}^{t_e^j}\left(\frac{i-\overline{t}_j+\overline{t}_j-\overline{T}}{v_t}\right)(s_i-\overline{S})\right\}$$

$$=\sum_{j=1}^{2}\left\{\sum_{i=t_b^j}^{t_e^j}\left(\frac{i-\overline{t}_j}{v_t}\right)(s_i-\overline{S})+\sum_{i=t_b^j}^{t_e^j}\left(\frac{\overline{t}_j-\overline{T}}{v_t}\right)(s_i-\overline{S})\right\} \tag{7}$$

$$=\sum_{j=1}^{2}\left\{\frac{W_j^3-W_j}{W_{12}^3-W_{12}}*k_j+6*\frac{2\sum_{i=1}^{j-1}W_i+W_j-W_{12}}{W_{12}^3-W_{12}}*W_j*(\overline{s}_j-\overline{S})\right\}$$

$$B=\overline{S}-K*\overline{T} \tag{8}$$

合并后的直线可依据公式(7)、公式(8)计算,计算过程中,要用到原来各段的采样平均值 $\overline{s}_i$.因此,在传感器节点向簇头或基站传数据的 PLR 表示时,如果还要进一步压缩或进行其他数据挖掘,段 S_i 的 PLR 修正为 $(t_i,k_i,b_i,\overline{s}_i)$.

方法 2:直接基于段 S_i 的 PLR 表示(t_i,k_i,b_i)进行求解,即最小化线段 S_1,S_2 上近似数据点的 L_2 误差进行线性拟合,得到新大段.段 S_1 的采样个数用 $W_1=t_2-t_1$ 表示,段 S_2 的为 W_2.平方逼近即最小化如下的 f:

$$f=\sum_{t=t_1}^{t_1+W_1-1}[(K*t+B)-(k_1*t+b_1)]^2+\sum_{t=t_2}^{t_2+W_2-1}[(K*t+B)-(k_2*t+b_2)]^2.$$

求解如下方程(9)即可求出平方逼近的 K,B.

$$\begin{cases}\sum_{j=1}^{2}[(K-k_j)*\sum_{t=t_j}^{t_j+W_j-1}t^2+(B-b_j)*\sum_{t=t_j}^{t_j+W_j-1}t]=0\\ \sum_{j=1}^{2}[(K-k_j)*\sum_{t=t_j}^{t_j+W_j-1}t+(B-b_j)*W_j]=0\end{cases} \tag{9}$$

Palpanas 等人指出[15],根据线段 S_1,S_2 的 PLR 拟合出的线段与根据原始采样数据拟合出的线段 S_{12} 是相同的.因此,上述方法 1、方法 2 计算出的 K,B 应该相等.根据文献[15]中的定理 2,S_{12} 的平方逼近误差可用公式在 $O(1)$ 时间内算出,从而可以根据误差限在线的决定是否将 PLR 表示的段进行合并.

5.2 不同节点相同时间段内的数据压缩

求聚集函数 SUM 就是将来自不同节点同一时刻同一物理量的值相加.在进行数据挖掘和数据压缩中,PLR 是非常好的表示方式.设来自传感器节点 S_i 的数据的 PLR 为(t_i,k_i,b_i).

定理 2(PLR 的可加性). 设来自传感器节点 S_i 的物理属性 A 数据的 PLR 为(t_1,k_i,b_i),即在 t 时刻其采样数据近似为 $s_i^t=k_i*t+b_i$,$i=1,\ldots,N,t\in[t_1,t_2]$.$N$ 个节点覆盖区域的 A 物理量的 SUM 函数定义为 $SUM(t)=\sum_{i=1}^{N}s_i^t$.

SUM 可用 PLR 表示为(t_1,K,B),$SUM(t)=K*t+B$.其中,$K=\sum_{i=1}^{N}k_i,B=\sum_{i=1}^{N}b_i$.

证明:设传感器节点 S_i 的 PLR 对应原始的 $W=t_2-t_1$ 个数据,此传感器采样数据平均值 $\overline{s}_i=\left(\sum_{t=t_1}^{t_2-1}s_i^t\right)\Big/W$,$i=1,\ldots,N$.时间的平均值 $\overline{t}=\left(\sum_{t=t_1}^{t_2-1}t\right)\Big/W$.SUM 的平均值 $\overline{S}=\left(\sum_{t=t_1}^{t_2-1}SUM(t)\right)\Big/W=\left(\sum_{t=t_1}^{t_2-1}\sum_{i=1}^{N}s_i^t\right)\Big/W=\sum_{i=1}^{N}\overline{s}_i$.参考第 5.1 节的推导,有 $v_t=\sum_{t=t_1}^{t_2-1}(t-\overline{t})^2=\frac{1}{12}(W^3-W)$.

$$K=\sum_{t=t_1}^{t_2-1}\left[\frac{t-\overline{t}}{v_t}(SUM(t)-\overline{S})\right]=\sum_{t=t_1}^{t_2-1}\left[\frac{t-\overline{t}}{v_t}(\sum_{i=1}^{N}s_i^t-\sum_{i=1}^{N}\overline{s}_i)\right]$$

$$=\sum_{i=1}^{N}\left[\sum_{t=t_1}^{t_2-1}\left(\frac{t-\overline{t}}{v_t}(s_i^t-\overline{s}_i)\right)\right]=\sum_{i=1}^{N}k_i,$$

$$B=\overline{S}-K*\overline{t}=\sum_{i=1}^{N}\overline{s}_i-(\sum_{i=1}^{N}k_i)*\overline{t}=\sum_{i=1}^{N}(\overline{s}_i-k_i*\overline{t})=\sum_{i=1}^{N}b_i.$$ □

类似地,聚集函数 $AVG(t)=SUM(t)/N$ 可用 PLR 表示为$(t_1,K/N,B/N)$.

6 实 验

数据集由 Madden 等人提供(http://db.csail.mit.edu/labdata/labdata.html),包含 54 个 Mica2Dot 节点同时采集的 230 多万次数据,每个节点采集 4 种属性数据:温度、湿度、光强和电压.传感器节点存储的采样数据、回归系数、段的均值、段内采样数据之和等均需要 2 字节进行存储.采用 VC++ 6.0 在 PC 机上实现算法.数据压缩性能用空间节省率(space saving)度量,定义为压缩后减少的数据量除以原始数据大小.平方逼近采用定义 2 为误差度量更有物理意义.误差限采用绝对误差,但也可以使用相对误差.

6.1 传感器节点数据的分段常量一致逼近

用段中所有数据的均值及此段的结束时间(或开始时间、数据个数也可以)来表示一个段,比较 3 种可以保证 L_∞误差有界的分段常量逼近算法:① 重点测试本文的 PCADC-sensor 算法性能,该算法利用了传感器节点上的缓冲,是一种准在线算法.压缩性能应该介于后两者之间;② 静态分段的 BottomUp 算法[4],是一种脱机批处理算法.当一次性将全部数据置于处理队列时,算法 1 就是 BottomUp 的一种实现.每次在所有误差有界的合并中,挑选生成的新段最长的原来两段进行合并,也可以挑选新段的最大误差为最小的进行合并.实验中发现,这两种方式性能相差不大;③ PMC-MEAN[6]是一种基于滑动窗口的完全在线算法.

取数据集第 1 个传感器节点最开始的 8192 个温度数据进行压缩,温度数值范围为[17.1954,26.6328],均值为 20.9586,方差为 6.6199.图 4 是 PCADC-sensor 算法在不同缓冲数组大小 *arraysize*、误差限 *eps*(亦记为 ε)下的空间节省率 *SS*.误差限根据统计特征来选择.根据前述算法,处理队列长度 *pq_len* 不超过 *arraysize* 的一半.仿真实验中,每次将处理队列充满.由图 4 可见:当 *arraysize* 一定时,*SS* 随 *eps* 的增加而增加;但 *eps* 大到一定程度后,*SS* 不再增加.因为随 *eps* 的增加,平均段长在增加,则处理队列中容纳的段在减少;极端情况,每次处理队列中全部数据被压缩成一段(如 *arraysize*=16 且 *eps*=3 时),*SS* 为常数 1–2/*pq_len*;此时可通过同步增加 *arraysize* 来提高 *SS*.但是,*arraysize* 越大,分段计算越慢;*arraysize* 大到一定程度后再增加对压缩无用.因此,要根据数据本身的分布、误差限来选择合适的 *arraysize*.

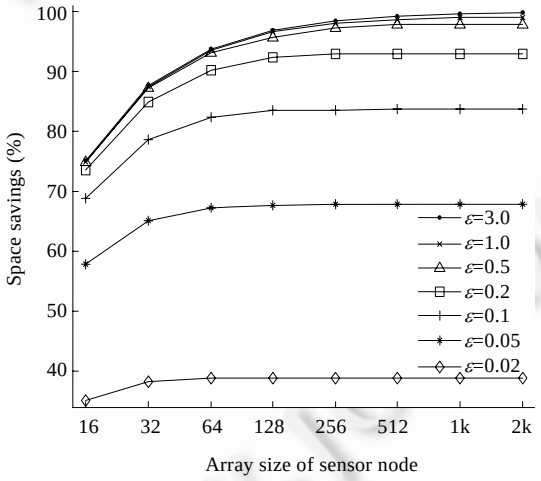

Fig.4 Compressing temperature data using PCADC-sensor

图 4 用 PCADC-sensor 算法压缩温度数据

取数据集第 1 个传感器最开始的 4096 个电压数据进行压缩,电压数值范围为[2.6396,2.7624],均值为 2.6892,方差为 0.000654.图 5 是这 3 种算法在不同误差限下的空间节省率.其中,PCADC-Sensor 采用两种 *arraysize* 分别进行实验,PCADC-sensor(1024)表示 *arraysize* 为 1024.误差限对各算法的压缩性能影响很大.极端情况,当$\|e\|_\infty$为 0 时,所有算法性能一样,将产生 4096 条线段,数据不仅没有被压缩,反而增加了 1 倍.当$\|e\|_\infty$很大

时,PCADC-sensor 算法每次将处理队列里全部数据作为一段输出;BottomUp 将待分段的全部数据视为一段;PMC-MEAN 会不断采样,段长不断增加,但一直没有输出.节点较长时间没有数据传出的话,用户或基站可能认为该节点已经死亡,这是要避免的.考察图 5,当 *eps*<0.005 时,各算法压缩性能一样.由于每段长度很短,脱机压缩的优势没有表现出来.逐渐加大误差限后,PCADC-sensor 与脱机分段 BottomUp 的压缩性能十分接近(性能曲线基本重叠),优于在线分段算法.PCADC-sensor 借助传感器节点本地缓冲,达到了较好的压缩效果.当 *eps* 取 0.02 时,BottomUp,PCADC-Sensor(1024),PMC-MEAN 分别将待分段数据分为 15 段、16 段、21 段,符合预期,脱机压缩性能更高.当 *eps*=0.035 时,BottomUp,PCADC-sensor(1024),PMC-MEAN 分别将待分段数据分为 5 段、10 段、7 段,这个现象是误差限增大使段的平均长度增加、缓冲数组里数据量不够引起的;此误差限下,PCADC-sensor(2048)算法分为 6 段,证实了这个推断.

实验结果表明:① 为获得较高空间节省率,脱机处理的 BottomUp 算法应以节点缓冲区容量为上限,尽可能多地积累一次处理的数据.但其压缩性能并没有预计的好,增加的计算开销和时延限制了其使用;② 滑动窗口算法只考虑当前采集的数据能否纳入当前段,数据流只扫描一趟,空间开销极小.PMC-MEAN 算法的压缩性能与待处理的数据个数无关,实验验证了这点.在处理温度等变化较平缓的数据时,压缩性能比预计的好;③ PCADC-Sensor 算法在处理队列可以容纳 7 段左右时,就可达到脱机分段的压缩水平;按照双循环队列的方式来管理数据缓冲区,取得了延时和压缩性能的一个平衡.在处理电压等变化较剧烈的数据时,压缩性能优于 PMC-MEAN;④ PCADC-sensor 算法动态调整 *arraysize*,可获得压缩性能和计算开销的一个折中.

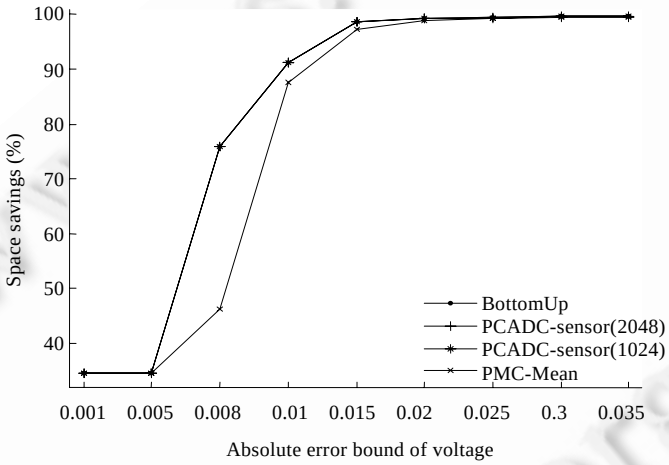

Fig.5 Space savings of three PCA algorithms

图 5 3 种分段常量逼近算法的压缩性能比较

6.2 传感器节点数据的分段线性逼近

算法 2 给出了一致逼近的近似算法 PLADC-sensor-inf,在线式单遍扫描数据流.用斜率、线段起点的横坐标和纵坐标共 3 个数据代表一段直线.采用这个算法,我们对数据集第 1 个传感器最开始的 4096 个温度、电压数据分别进行压缩,并与 BottomUp,PMC-MEAN 进行对比,如图 6 和图 7 所示.当 *eps* 较大时,三者性能接近.对于变化较平缓的温度数据,如图 6 所示,PLADC-sensor-inf 算法的压缩性能全都超过了 BottomUp 和 PMC-MEAN 算法.对于变化剧烈的电压数据,如图 7 所示,PLADC-sensor-inf 没有明显优势,在误差限为中等大小[0.005, 0.009],优于也是在线式的 PMC-MEAN.我们认为,PLADC-sensor-inf 与 BottomUp,PMC-MEAN 相比,压缩性能受数据本身分布的影响更大.虽然前者可用任意线段逼近原函数,而后两者只能用水平线段逼近原函数,但是前者需要 3 个数据代表 1 段,而后两者只需 2 个数据代表 1 段.

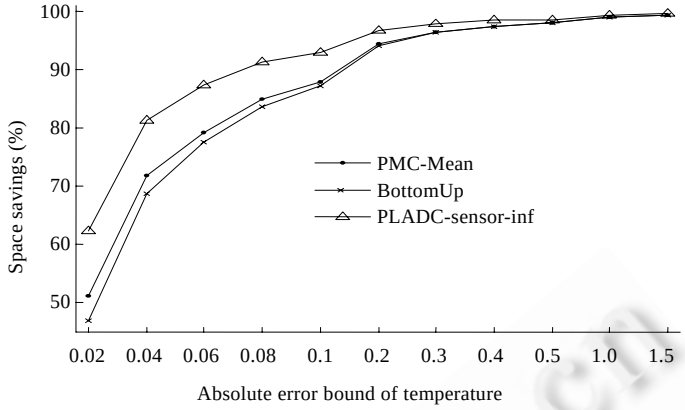

Fig.6 Space savings of three compressing temperature data algorithms

图 6 压缩温度数据的 3 种算法的空间节省率

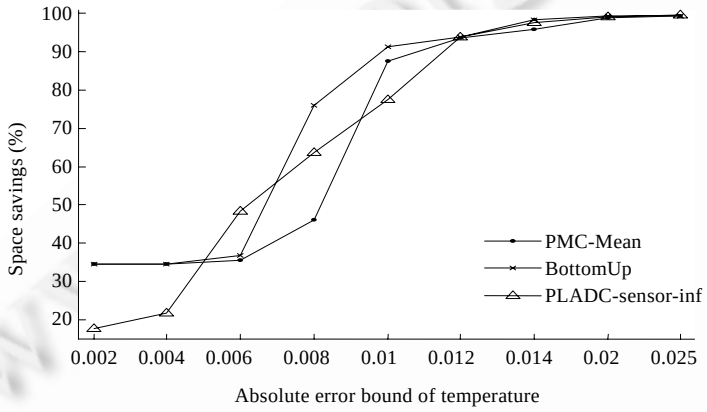

Fig.7 Space savings of three compressing voltage data algorithms

图 7 压缩电压数据的 3 种算法的空间节省率

算法 3 给出了平方逼近的递推式在线算法 PLADC-sensor-2,用斜率、线段起点的横坐标和本段在纵轴上的截距共 3 个数据代表一段直线.采用这种算法,我们对数据集第 1 个传感器最开始的 4096 个温度数据进行压缩.误差限用 2-范数平均误差度量.图 8 说明了不同误差限下,PLADC-sensor-2 算法的压缩性能.

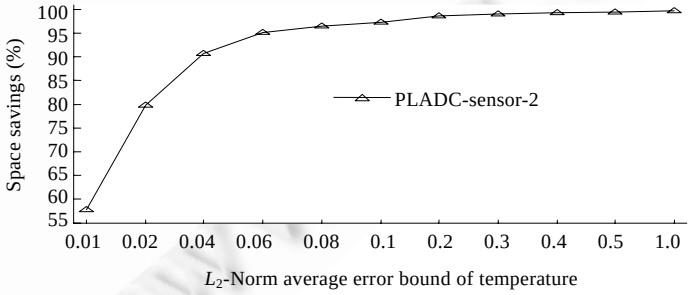

Fig.8 Space savings of PCADC-sensor-2 compressing temperature data

图 8 用 PLADC-sensor-2 压缩温度数据的空间节省率

图 9 的左、右子图分别说明了误差限为 0.2,0.5 时,温度数据用 PLADC-sensor-2 算法压缩,原始数据和压缩

数据的对比.误差限为 0.2 时,用 18 根不连续的线段逼近原始数据,空间节省率为 98.68%;误差限为 0.5 时,用 7 根不连续的线段逼近原始数据,空间节省率为 99.49%.注意,图中基本垂直于时间轴的线段不算,因为不代表逼近数据;绘图时把两个不连续的线段连接起来了而已.

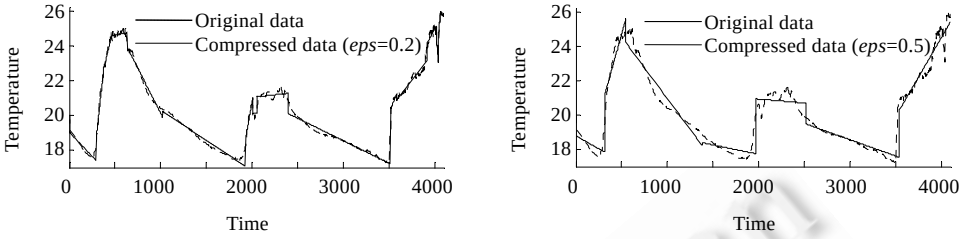

Fig.9 Comparison before and after compression with different error bounds (online least square approximation)

图 9 不同误差限下压缩前后对比(在线式平方逼近)

6.3 簇头或基站上基于分段线性表示的数据压缩

簇头或基站收集到某传感器节点相邻的不同时间段的 PLR 后,在平方逼近意义下将段合并而进一步压缩,即用一个 PLR 代表更长时间内的数据.从数据集第 1 个传感器的温度采样中取出第 605 个~618 个数据,将其编号为 0~13.采用算法 3 得出,此 14 个数据被分成 2 段,各有 7 个数据,其 PLR 分别为(0,−0.0315,24.8555),(7,0.0371,24.5426),如图 10 的上子图.如第 5.1 节所述,如果在基站或簇头还要进一步压缩,段 S_i 的 PLR 修正为 $(t_i,k_i,b_i,\overline{s_i})$,即每段的采样数据平均值也需要传送给基站或簇头.$\overline{s_1}$为 24.7610,$\overline{s_2}$为 24.9136.根据公式(7)、公式(8),得到原数据更粗糙的逼近表示为(0,0.017123,24.7260),如图 10 中下子图所示.这与直接基于原始采样数据计算平方逼近的结果是相同的,验证了公式(7)、公式(8)的有效性.另外,若段 S_i 的 PLR 用(t_i,k_i,b_i)表示,解公式(9)的方程组也可求得段合并后的 PLR.

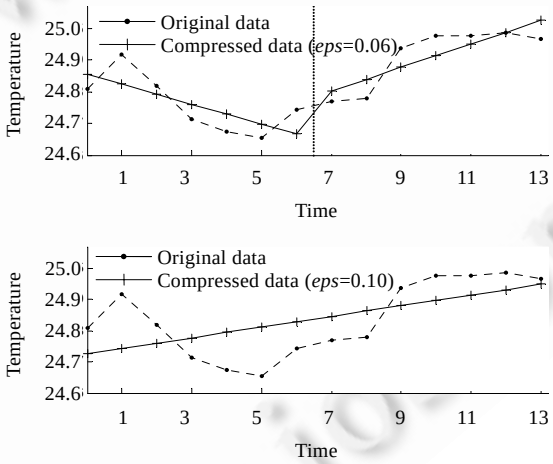

Fig.10 Merging sequential time intervals PLRs of same sensor node

图 10 同一传感器在相邻时间段的 PLR 的合并

图 10 中下面的子图显示误差较大,存在局部信息特性丢失.可解释为:① 图中时间坐标和物理量坐标的单位太小,分辨率太高,显得误差较大;② 直接基于传感数据的 PLR 根据公式(7)、公式(8)不断进行压缩,新 PLR 的重构误差会不断增大.根据文献[15]中定理 2,可根据误差限在线式的决定是否将 PLR 进行合并,从而保证了 2 范数误差有界;③ 许多数据流应用中,数据的作用是随时间衰减的,称为数据流的遗忘特性.如股票价格,对近几天可能关心其具体的价格,对一年前的主要关心其大致波动范围和趋势.显然,PLR 适合于在簇头、基站、数据

中心等地方表示较久过去的数据.随着时间的流逝,可不断进行 PLR 合并.

簇头或基站收集到了不同传感器节点的 PLR,用相同时段内的 PLR 计算聚集函数.为使不同节点的 PLR 对应相同时段,可能先需要求出各个传感器节点的在指定时段内的 PLR:如果传感器指定时间段内只有一个 PLR 则直接使用;否则,利用第 5.1 节的方法进行 PLR 合并.考察数据集在 2004 年 2 月 28 日凌晨 1 点开始的 10 分钟内,取第 1 个、第 8 个 Mote 的温度采样作为实验数据.这两个传感器都采集了 8 次数据,分别将其编号为 0~7.采用算法 3,将各传感器的 8 个数据各自拟合成一段,Mote1 的温度(记为 S_1)的 PLR 为(0,–0.07875,19.5572),Mote8 的温度(记为 S_8)的 PLR 为(0,–0.04305,19.0721),如图 11 中左面子图所示.根据第 5.2 节定理 2,得到聚集函数 $SUM(t)=S_1(t)+S_8(t)$的逼近表示 PLR 为(0,–0.1218,38.6293),如图 11 中右面子图所示.这与直接基于同一时刻 Mote1 和 Mote8 原始温度采样之和计算平方逼近的结果是相同的,验证了定理 2 的有效性.

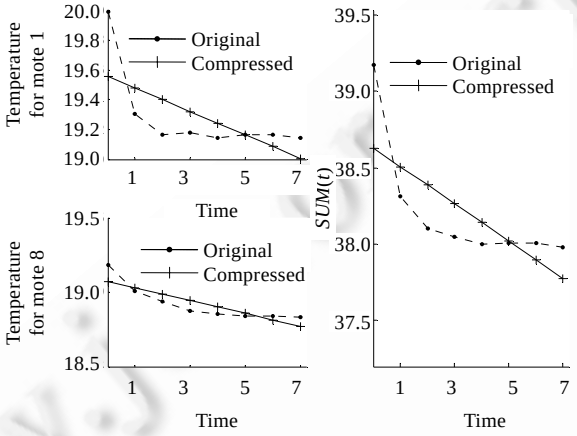

Fig.11 Aggregation same time interval PLRs of different sensor nodes

图 11 不同传感器在相同时间段的 PLR 的聚集

7 结 论

针对传感器节点的数据采样之间不存在空间相关性或空间相关性不稳定,本文设计了在各节点上独立运行的误差有界的在线式分段逼近数据压缩算法,主要消除数据的时间相关性.这些压缩表示(分段线性表示)除了可以保证误差有界的重构出近似原始数据外,还可以直接应用在簇头和基站进行进一步压缩或计算聚集函数,这也部分消除了空间相关性.根据分段线性一致逼近的充要条件分段时,实现计算简单的算法还值得进一步研究;本文为了减少计算复杂度提出了一个近似实现,压缩性能不如基于充要条件的实现.

References:

[1] Li JZ, Gao H. Survey on sensor network research. Journal of Computer Research and Development, 2008,45(1):1−15 (in Chinese with English abstract).

[2] Pottie GJ, Kaiser WJ. Wireless integrated network sensors. Communications of the ACM, 2000,43(5):51−58. http://dx.doi.org/10.1145/332833.332838

[3] Pang CY, Zhang Q, Hansen D, Maeder A. Unrestricted wavelet synopses under maximum error bound. In: Proc. of the EDBT. New York: ACM Press, 2009. 732−743. http://dx.doi.org/10.1145/1516360.1516445

[4] Keogh E, Chu S, Hart D, Pazzani M. An online algorithm for segmenting time series. In: Proc. of the ICDM. Piscataway: IEEE Press, 2001. 289−296. http://dx.doi.org/10.1109/ICDM.2001.989531

[5] Soroush E, Wu K, Pei J. Fast and quality-guaranteed data streaming in resource-constrained sensor networks. In: Proc. of the MobiHoc. New York: ACM Press, 2008. 391−400. http://dx.doi.org/10.1145/1374618.1374670

[6] Lazaridis I, Mehrotra S. Capturing sensor-generated time series with quality guarantees. In: Proc. of the 19th Int'l Conf. on Data Engineering. Piscataway: IEEE Press, 2003. 429−440. http://dx.doi.org/10.1109/ICDE.2003.1260811

[7] Gandhi S, Nath S, Suri S, Liu J. GAMPS: Compressing multi sensor data by grouping and amplitude scaling. In: Proc. of the SIGMOD. New York: ACM Press, 2009. 771−783.

[8] Keogh E, Chakrabarti K, Pazzani M, Mehrotra S. Dimensionality reduction for fast similarity search in large time series databases. Knowledge and Information Systems, 2000,3(3):263−286. [doi: 10.1007/PL00011669]

[9] Lin J, Keogh E, Lonardi S, Chiu B. A symbolic representation of time series, with implications for streaming algorithms. In: Proc. of the ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery. New York: ACM Press, 2003. 2−11. http://dx.doi.org/10.1145/882082.882086

[10] Ciancio A, Pattem S, Ortega A, Krishnamachari B. Energy-Efficient data representation and routing for wireless sensor networks based on a distributed wavelet compression algorithm. In: Proc. of the IPSN. New York: ACM Press, 2006. 309−316. http://dx.doi.org/10.1145/1127777.1127824

[11] Zhou SW, Lin YP, Zhang JM, Ouyang JC, Lu XG. A wavelet data compression algorithm using ring topology for wireless sensor networks. Journal of Software, 2007,18(3):669−680 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/18/669.htm [doi: 10.1360/jos180669]

[12] Deligiannakis A, Kotidis Y, Roussopoulos N. Dissemination of compressed historical information in sensor networks. The VLDB Journal, 2007,16(4):439−461. [doi: 1007/s00778-005-0173-5]

[13] Zhang JM, Lin YP, Zhou SW, Tan XL. Self-Adaptive regression based multivariate data compression algorithm in wireless sensor networks. Computer Science, 2009,36(10A):269−274 (in Chinese with English abstract).

[14] Lim JJ, Shin KG. Energy-Efficient self-adapting online linear forecasting for wireless sensor network applications. In: Proc. of the MASS. Piscataway: IEEE Press, 2005. 372−379. http://dx.doi.org/10.1109/MAHSS.2005.1542822

[15] Palpanas T, Vlachos M, Keogh E, Gunopulos D. Streaming time series summarization using user-defined amnesic functions. IEEE Trans. on Knowledge and Data Engineering, 2008,20(7):992−1006. http://dx.doi.org/10.1109/TKDE.2007.190737

附中文参考文献:

[1] 李建中,高宏.无线传感器网络的研究进展.计算机研究与发展,2008,45(1):1−15.

[11] 周四望,林亚平,张建明,欧阳竞成,卢新国.传感器网络中基于环模型的小波数据压缩算法.软件学报,2007,18(3):669−680. http://www.jos.org.cn/1000-9825/18/669.htm [doi: 10.1360/jos180669]

[13] 张建明,林亚平,周四望,谭新良.传感器网络中多属性数据基于自适应回归的压缩算法.计算机科学,2009,36(10A):269−274.

张建明(1976－),男,湖南益阳人,博士,副教授,CCF 会员,主要研究领域为传感器网络中的数据管理,数据挖掘.

傅明(1961－),男,博士,教授,主要研究领域为计算机网络,数据挖掘.

林亚平(1955－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为计算机网络,机器学习.

周四望(1971－),男,博士,副教授,CCF 会员,主要研究领域为传感器网络中的信号处理,小波分析.