软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2015,26(10):2614−2630 [doi: 10.13328/j.cnki.jos.004806]

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

面向数据流的多粒度时变分形维数计算*

倪志伟[1,2], 王 超[1,2,3], 胡汤磊[1,2], 倪丽萍[1,2]

[1](合肥工业大学 管理学院,安徽 合肥 230009)

[2](教育部过程优化与智能决策重点实验室(合肥工业大学),安徽 合肥 230009)

[3](安徽农业大学 信息与计算机学院,安徽 合肥 230036)

通讯作者: 王超, E-mail: warton_cc@163.com, http://www.hfut.edu.cn

摘 要: 在大数据时代,数据流是一种常见的数据模型,具有有序、海量、时变等特点.分形是许多复杂系统的重要特征,分形维数是度量系统分形特征的重要指标量.数据流作为动态的复杂系统,其上的分形维数应具有动态、时变、多粒度等特性.提出了多粒度时变分形维数的概念,并设计了基于小波变换技术的数据流多粒度时变分形维数算法.该算法通过对数据流进行离散小波变换,并利用多粒度小波变换树结构在内存中保存数据流的概要信息,可以同时在不同的时间粒度上实时地计算数据流时变分形维数.该方法具有较低的计算复杂度,实验结果表明:该方法可以有效地监控数据流分形维数在不同粒度上的时变特征,深刻地揭示数据流的演化规律.

关键词: 数据流;分形维数;小波变换;多粒度;时变性

中图法分类号: TP311

中文引用格式: 倪志伟,王超,胡汤磊,倪丽萍.面向数据流的多粒度时变分形维数计算.软件学报,2015,26(10):2614−2630. http://www.jos.org.cn/1000-9825/4806.htm

英文引用格式: Ni ZW, Wang C, Hu TL, Ni LP. Multi-Granularity and time-varying fractal dimension on data stream. Ruan Jian Xue Bao/Journal of Software, 2015,26(10):2614−2630 (in Chinese). http://www.jos.org.cn/1000-9825/4806.htm

Multi-Granularity and Time-Varying Fractal Dimension on Data Stream

NI Zhi-Wei[1,2], WANG Chao[1,2,3], HU Tang-Lei[1,2], NI Li-Ping[1,2]

[1](School of Management, Hefei University of Technology, Hefei 230009, China)

[2](Key Laboratory of Process Optimization and Intelligent Decision-Making of the Ministry of Education (Hefei University of Technology), Hefei 230009, China)

[3](School of Information and Computer, Anhui Agricultural University, Hefei 230036, China)

Abstract: In the era of big data, data stream is a common data model with characteristics such as orderly, massive and time-varying. Fractal is an important feature of many complex systems, and is mainly represented by fractal dimension. Data stream can be viewed as a dynamic and complex system, and its fractal dimension should also have characteristics of dynamic, time-varying and multi-granularity. This paper presents a method of measuring multi-granularity and time-varying fractal dimension on a data stream based on discrete wavelet transform. The method can simultaneously measure the time-varying fractal dimension on a data stream by using the summary information from wavelet transforming of the data stream saved in a multi-granularity wavelet transforming tree in memory. This method has low computational complexity, and effectively reveals the evolution of a data stream. Experimental results show that it can effectively monitor the time-varying characteristic of fractal dimension on a data stream at different granularity.

Key words: data stream; fractal dimension; wavelet transform; multi-granularity; time-varying

在大数据时代,信息具有大量化、多样化和快速化等特点,许多应用领域会快速、实时地产生大量的数据,例如股票市场中众多股票的实时交易信息、气象传感器网络采集的观测数据等.数据流模型可以准确地描述这

* 基金项目: 国家自然科学基金(71271071, 71301041); 国家高技术研究发展计划(863)(2011AA040501)
收稿时间: 2014-02-20; 修改时间: 2014-05-09, 2014-09-16, 2014-12-09; 定稿时间: 2014-12-16

类依照特定的时间戳有序排列、具有海量规模(可能无限)、实时到达、不断变化等特点的序列数据.对这些数据流的相关信息进行实时分析与精确处理,是快速、合理地做出决策的根本依据.

分形是指系统的局部与整体具有某种方式的相似性,即自相似性,在小尺度上的细节信息与大尺度上的总体信息具有严格的或统计意义上的相似性[1].分形维数是描述分形的主要参数,是表示分形对相应嵌入空间的填充程度的统计量,其大小刻画了系统的复杂程度.因此,分形维数可以作为数据流分布变化的一个关键特征,为更深层次的知识发现工作做铺垫,如基于分形技术的数据流聚类分析[2]、突变点检测[3]、概念漂移检测[4]等.

分形维数作为反映数据集分布情况的特征统计量,在数据挖掘领域已经得到了广泛的应用.Faloutsos 指出,网络数据、传感数据、图数据、医学数据、地理数据、金融数据等数据集都适合使用分形技术进行描述和分析[1].因此,分形在数据挖掘的不同领域都有着广泛的应用.通过文献分析我们发现,当前的分形维数计算方法存在以下几个问题:

1. 传统的分形维数计算方法不能满足数据流环境的特点,因此不能直接应用到数据流分形维数计算中.

传统的分形维数有多种计算方法:

(1) 盒计数法[5]及其改进方法,如 FD3 算法[6]、Z-order 算法[7]、最小聚类体积覆盖方法[8]等:通过计算非空盒子内的数据分布情况与划分盒子的对应尺度间的双对数曲线的斜率近似分形维数;

(2) G-P 算法[9]:使用延迟时间嵌入方法重构相空间,并在相空间中计算对应尺度下的数据分布情况,得出关联分形维数;

(3) 小波方差法[10]:对数据集进行小波变换,估计小波方差和相关尺度的双对数曲线的斜率,进而计算出分形维数;

(4) Hguchi 分形维数算法[11]:针对时间序列数据的分形维数计算方法.首先,将原时间序列拆分,构建 k 条新的时间序列;然后,计算 k 条新时间序列的长度及其均值;最后,计算长度和均值对应的双对数曲线的斜率,得出 Higuchi 分形维数.

以上方法只适用于静态数据集的分形维数的计算,需要在完整的数据集上运算,且部分算法需要多遍读取数据集.不能满足数据流挖掘的要求.

例如:文献[12,13]将分形维数的变化作为属性约减和特征选择的评估准则;文献[14]将分形维数作为模糊聚类的聚类标准;文献[15]提出了有效分形维数的概念,并通过实验验证了有效分形维数可以有效地提高文本特征提取和分类准确度;文献[16]通过对出租车运行轨迹数据集的分形维数的分析,发现了一些有趣的模式,其中,在计算分形维数时都采用了盒计数法,对数据集进行了多次扫描,不能满足数据流挖掘的实时性要求;文献[17]提出了基于分形维数和香农熵的分类准则,可以有效地识别和分类人类 Y 染色体基因.其中,分形维数的计算采用了 Hguchi 分形维数算法,需要扫描完整的数据集,不能满足数据流挖掘的未来数据不可预知性的要求;文献[18]提出了基于分形的聚类层次优化算法,利用聚类对应的多重分形维数及聚类合并之后多重分形维数的变化程度来度量初始聚类之间的相似程度,并最终生成反映数据自然聚集状态的聚类层次树.其中,在计算分形维数时采用了 Z-order 算法,只需扫描数据集 1 遍,但是需要扫描完整的数据集,并且忽略了数据的时序特征,不能满足数据流挖掘的时序性要求.

2. 现有的数据流分形维数计算方法大多是基于盒计数法的,需要对数据空间进行划分.但是数据流是动态变化的,相应的数据空间也是动态变化的.因此,需要根据数据流的变化情况不断地调整数据空间的划分,增加了计算的复杂度.

文献[19]设计了时空行为计量(spatio-temporal behavior meter,简称 STB-meter)方法,识别多维数据流的时空模式.该方法采用了多尺度层次结构的方法处理空间信息,采用基于分形的方法监控数据流的时序信息,在真实的气候数据集上的相关实验结果表明,STB-meter 方法可以有效地发现重要的气候模式和一些极端气候事件.文献[20]提出了 Tug-of-War 算法,将数据空间划分出的每个独立的单元格看作是一个事件,每个单元格的计数问题转化为求相关事件的二阶矩.Tug-of-War 算法只需扫描 1 遍数据,且可在不牺牲计算精度和速度的情况下将空间复杂度控制在 $O(1)$.文献[21]设计了基于数据流概要的分形维数计算方法,采用 Count-Min Sketch 和

Flajolet Martin Sketch 技术构建数据流的概要信息,利用概要信息计算数据流的分形维数.该方法在概率近似正确(probably approximate correct,简称 PAC)框架下以至少 $1-\delta$的概率保证概要信息的误差小于ε,并且在计算速度上快于 Tug-of-War 算法.其中,STB-meter 方法是一种存储数据分布信息的数据结构,与 Tug-of-War 算法及基于数据流概要的分形维数计算方法同属于基于盒计数法的分形维数计算方法,但它忽略了对数据流计算窗口内的时序特征和数据空间动态变化的考量.

3. 现有的数据流分形维数计算方法只能计算固定尺寸窗口内(即单一粒度)的局部分形维数,或是到目前为止观测到的所有数据流的整体的分形维数,不能在不同的粒度下计算数据流的分形维数并观察数据流在不同尺度下的变化情况.

文献[22−24]提出了 SID 算法,采用滑动窗口的方法对数据流进行分割,并设计了计数树结构记录数据的分布情况.SID 算法具有较低的计算复杂度,能够快速地计算数据流当前滑动窗口状态下的分形维数,但却无法获得数据流的整体分形维数和不同时间粒度内的分形维数变化情况.

由于数据流是有序实时到达的数据序列,未来的数据流模式具有不确定性和不可预知性.数据流模式的挖掘是一个由细节到整体的不断提升的过程,即是一个多粒度过程:先由较小的粒度开始挖掘,随着数据流的不断到来,再挖掘较大粒度上的模式,逐渐清晰地展示整个数据流在不同粒度下的模式.因此,本文将结合离散小波变换技术,通过计算不同粒度下数据流的小波谱,进而估算数据流的多粒度时变分形维数.同时,从宏观和微观的角度对数据流进行分析,更深刻地揭示数据流的演化规律.

因此,为了揭示这种演化规律,本文做了以下工作:(1) 基于数据流的特点,首次提出了多粒度时变分形维数的概念;(2) 介绍了数据流小波变换的融合延拓方法,并在此基础上设计了满足数据流挖掘相关要求的多粒度小波变换树,可以提取和保存数据流的概要和小波谱信息,实时地计算数据流多粒度时变分形维数;(3) 通过一系列实验验证了多粒度时变分形维数的概念和计算方法的有效性.

本文第 1 节介绍我们的研究所涉及到的相关概念和知识.第 2 节讨论数据流上分形维数的特点以及多粒度时变分形维数的计算方法.第 3 节通过实验验证本文概念和方法的有效性.第 4 节是对全文的总结并给出进一步的研究方向.

1 相关概念和知识

本文主要研究时间序列数据流的时变分形维数,盒计数法无法对整个时间序列数据流的数据空间进行划分,且对数据的时序性不敏感,因此不适合计算时间序列数据流的分形维数.本文将采用小波谱[25]方法计算多粒度时变分形维数.

(1) 基于小波变换的分形维数计算

随机过程 $f(x)$通过小波变换可被分解为低频的 $\sum_{k\in Z} a_{j_0,k}\phi_{j_0,k}(x)$ 部分(即全局特征)和高频的 $\sum_{j=j_0}^{\infty}\sum_{k\in Z} d_{j,k}\psi_{j,k}(x)$ 部分(即局部特征),其中, $a_{j_0,k}$ 和 $d_{j,k}$ 共同组成了频域中的小波系数[26].当小波函数的消失矩 R 满足条件:$R>\gamma/2$,信号 $f(x)$在时间戳 $2^j k$ 和频率 $2^{-j}v_0$(v_0 与使用的小波基函数有关)处的能量为$|d_{j,k}|^2$,特定尺度 j 的小波能量谱可由公式(1)计算得出:

$$P(2^{-j}v_0)=\frac{1}{n_j}\sum_k |d_{j,k}|^2 \tag{1}$$

其中,n_j 表示尺度 j 上的小波系数数量,$n_j=2^{-j}n$,n 表示数据长度.

自相似过程的频谱具有性质[27]:$P(2^{-j}v_0)\propto(2^{-j}v_0)^{-\gamma}$,$\gamma\in(-1,3)$是频谱指数.当 $f(x)$是分数布朗运动时,$\gamma\in(1,3)$,且 $F=(5-\gamma)/2$;当 $f(x)$是分数高斯噪声时,$\gamma\in(-1,1)$,且 $F=(3-\gamma)/2$.根据自相似过程的频谱性质,可以推导出公式(2),并采用最小二乘法估算频谱指数γ的值,进而得出分形维数 F 的值:

$$\log_2(P(2^{-j}v_0))=\log_2\left(\frac{1}{n_j}\sum_k |d_{j,k}|^2\right)=\log_2(A(2^{-j}v_0)^{-\gamma})=\gamma\times j+B \tag{2}$$

其中,A,B 均为常数.

为了满足小波函数的消失矩 $R>\gamma/2$ 的条件,本文将采用消失矩为 2 的 DB2 小波[25]作为数据流小波变换的基小波.

(2) 时间序列数据流多粒度时,变分形维数定义为

$$F=\{F_{ij}|i,j=1,2,\ldots\} \tag{3}$$

其中,F_{ij} 表示多粒度时变分形维数,其中,i 表示分形维数的粒度层次序号,j 表示对应粒度层次下的数据流时变分形维数的序号.如图 1 所示,i 与 j 存在如下关系:

- 当 $i=k$ 时,j 的最大值为

$$j_{\max}=floor\left(\frac{n}{m^{k-1}}\right),k=1,2,...,\log_m^n;m,n=1,2,... \tag{4}$$

其中,m 表示层次粒度,n 表示时间序列数据流子窗口的数量,k 表示粒度层次序号,$floor()$向下取整函数.

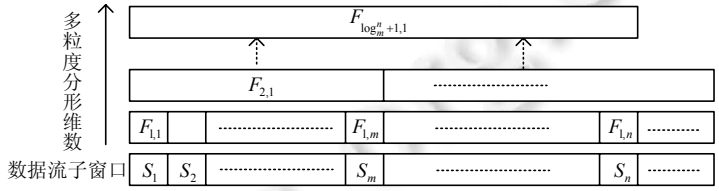


Fig.1 Illustration of multi-granularity and time-varying fractal dimension

图 1 多粒度时变分形维数示意图

通过多粒度时变分形维数,可以实现对时间序列数据流的分形维数进行多层次的实时监控,及时而准确地发现数据流的趋势变化信息.如:对 ICU 病房的 ECG(心电图)信号进行多粒度时变分形维数的监控,可以实时地从不同粒度观察 ECG 信号的变化模式,适时地给出针对性的措施;在股票时间序列的相似性研究中,可以综合考虑不同粒度下的时变分形维数,获得更加详细的数据变化规律,将其作为一个相似性度量的指标,能够更全面而准确地度量时间序列间的相似度.

2 数据流多粒度时变分形维数的计算

本文结合第 2.1 节提出的数据流分析的特点,通过改进小波变换方法,设计了可以在线计算一维时间序列型数据流 $S=\{s_1,s_2,\ldots,s_i,\ldots\}$在不同粒度下的时变分形维数的方法.主要分为两个步骤:首先对 S 进行 DB2 小波变换;其次,在内存中构建多粒度小波变换树.具体如下.

2.1 数据流小波变换

(1) 数据流窗口边界延拓

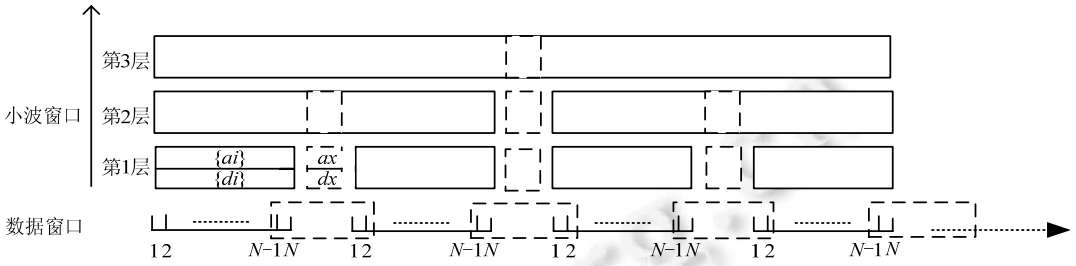
在传统的小波变换方法中,信号(数据集)的长度是有限的,会出现边界问题.公式(5)是 DB2 小波对 8 个数据元素信号的小波变换矩阵.通过矩阵的最后两行可以看出,DB2 小波变换的完成必须要再增加两个元素 s_9 和 s_{10}.

$$\begin{bmatrix} h_0 & h_1 & h_2 & h_3 & 0 & 0 & 0 & 0 & & \\ g_0 & g_1 & g_3 & g_4 & 0 & 0 & 0 & 0 & & \\ 0 & 0 & h_0 & h_1 & h_2 & h_3 & 0 & 0 & & \\ 0 & 0 & g_0 & g_1 & g_3 & g_4 & 0 & 0 & & \\ 0 & 0 & 0 & 0 & h_0 & h_1 & h_2 & h_3 & & \\ 0 & 0 & 0 & 0 & g_1 & g_2 & g_3 & g_4 & & \\ 0 & 0 & 0 & 0 & 0 & 0 & h_0 & h_1 & h_2 & h_3 \\ 0 & 0 & 0 & 0 & 0 & 0 & g_0 & g_1 & g_2 & g_3 \end{bmatrix} * \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \\ s_8 \end{bmatrix} \tag{5}$$

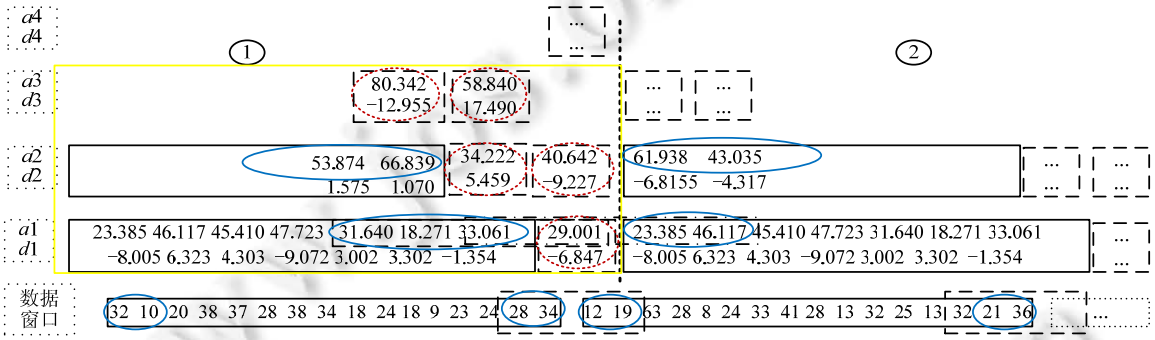
信号的边界延拓可以解决这一问题,常用的有 3 种方法:零延拓、对称延拓和周期延拓.零延拓是在信号的

两端补 0,以满足小波变换的需求;对称延拓是分别以信号的两个端点为中心,将信号对称地映射到两端;周期延拓是将信号看作一个周期,在信号的两端各增加 1 个周期以延长信号的长度.

由于数据流具有无限性的特点,本文将采用固定大小的滑动窗口对数据流进行划分,并设计一种针对数据流特点的边界延拓方式——融合延拓.融合延拓是在当前滑动窗口数据的两端补上前一窗口和后一窗口的数据,以满足相应小波变换所需的延拓数据.如图 2(a)所示,数据窗口是指在数据流上直接划分的窗口,小波窗口是指进行小波变换后保存近似系数和小波系数的窗口.



(a) 数据流小波变换融合延拓图



(b) 数据流小波变换融合延拓实例图

Fig.2 Data stream merging extended in wavelet transform and examples

图 2 数据流窗口融合延拓及实例图

在数据流窗口融合延拓过程中,每个数据窗口的数据在进行小波变换后只需保留窗口左、右两端各两个数据.数据窗口端点的两个数据有两个作用:一是作为数据边界的延拓,用来计算数据窗口边界的小波系数,如第 1 层小波窗口中的 a_x 和 d_x,即,前一个小波窗口的右边界小波系数;二是将两个数据窗口融合为更大时间粒度的数据窗口,在窗口融合的过程中,只需对边界数据进行小波变换,就可以得到整个大时间粒度窗口的小波变换系数,而不用重新对大时间粒度窗口的数据进行小波变换.

图 2(b)举例说明了数据流窗口融合延拓机制.DB2 小波的尺度函数系数和小波函数系数分别为 $h=\left(\frac{1+\sqrt{3}}{4\sqrt{2}},\frac{3+\sqrt{3}}{4\sqrt{2}},\frac{3-\sqrt{3}}{4\sqrt{2}},\frac{1-\sqrt{3}}{4\sqrt{2}}\right)$ 和 $g=\left(\frac{1-\sqrt{3}}{4\sqrt{2}},\frac{-3+\sqrt{3}}{4\sqrt{2}},\frac{3+\sqrt{3}}{4\sqrt{2}},\frac{-1-\sqrt{3}}{4\sqrt{2}}\right)$.第 k 和 k+1 个数据窗口的数据分别为 s_k=(32,10,20,38,37,28,34,18,9,23,24,28,34),s_{k+1}=(12,19,63,28,8,24,33,41,28,13,32,25,13,32,36).

首先,对第 k 个数据流窗口进行小波变换:不考虑边界延拓的问题,计算第 k 个数据窗口的 DB2 小波变换,即,计算 h 和 g 分别与第 k 个数据窗口的内积,$a[i]=h_0\times s_k[2i]+h_1\times s_k[2i+1]+h_2\times s_k[2i+2]+h_3\times s_k[2i+3]$,$d[i]=g_0\times s_k[2i]+g_1\times s_k[2i+1]+g_2\times s_k[2i+2]+g_3\times s_k[2i+3]$,如图 2(b)所示,$a1$=(23.385,46.117,45.410,47.723,31.640,18.271,33.061),$d1$=(−8.005,6.323,−9.072,3.002,3.302,−1.354).

其次,计算第 k 个数据流窗口的融合延拓边界:将第 k 个窗口的最后两个数据元素和第 k+1 个窗口的前两个

数据元素按顺序组合,计算其与 DB2 小波的尺度函数系数和小波函数系数的内积,即 $g\times(28,34,12,19)$和 $h\times(28,34,12,19)$,得到第 1 粒度层次的小波窗口的第 1 层小波变换系数的边界延拓系数$(a1x,d1x)=(29.001,-6.847)$,并与第 1 步计算的小波变换系数一起组成了第 k 个窗口的完整的第 1 层小波变换系数.

同上步骤,再计算第 1 粒度层次小波窗口的第 2 层小波变换系数的边界延拓系数,直到计算到这一粒度层次小波变换的顶层边界延拓系数.

(2) 数据流窗口融合

低粒度层次相邻小波窗口的融合可以形成更高粒度层次的小波窗口:根据数据流窗口融合延拓机制,只需要将相邻小波窗口的各层相邻的小波近似系数合并,再与 DB2 小波的尺度函数系数和小波函数系数作内积运算,作为更高粒度层次的小波窗口的系数,而同层的其他小波变换系数只要将对应层次的小波变换系数按序合并在一起即可.

窗口的融合并不是顺序进行的,只有小波窗口层次相同的相邻窗口才能融合.不同数据窗口小波变换的融合延拓对窗口融合的层次的影响是不同的,具体关系可以通过定理 1 得出.

定理 1. 假设当前数据窗口的序号为 $k=1,2,\ldots$,数据窗口的融合顺序与相关小波窗口的层次之间存在如下关系 $A(\oplus$,融合符号):

$$A=\begin{cases}k-2^{i-1}\oplus k-2^{i-1}+1, & k=2^n,k>2^{i-1},i=1,2,3,...,n\\ k-1\oplus k, & k\neq 2^n,k \bmod 2=0\end{cases} \tag{6}$$

证明:若 $k\neq 2^n$,且 k 为偶数,则相邻的两个数据窗口直接融合,得到第 2 粒度层次的第 $k/2$ 个小波窗口;若 $k=2^n$,则按照 $i=1,2,3,\ldots,n$ 的顺序对序号为 $k-2^{i-1}$ 和 $k-2^{i-1}+1$ 的数据窗口融合,并产生第 $i+1$ 粒度层次的第 $k/2^i$ 个小波窗口;若 k 为奇数,则数据窗口不进行融合操作. □

如图 2(a)所示:

- 首先,对当前的每个数据窗口进行 DB2 小波变换,生成第 1 粒度层次小波窗口;
- 其次,当 $k=2$ 时,计算第 1 个和第 2 个数据窗口的融合延拓,生成第 2 粒度层次的第 1 个小波窗口;
- 最后,当 $k=4$ 时,由于 $k>2^{i-1}$,所以 $i=1,2$,即,数据窗口的融合延拓顺序为:首先,在 $i=1$ 时,第 4 个和第 3 个数据窗口融合延拓产生第 2 粒度层次的第 2 个小波窗口;其次,由于第 1 个和第 2 个数据窗口与第 3 个和第 4 个数据窗口都已融合,所以在 $i=2$ 时,第 2 个和第 3 个数据窗口融合延拓产生第 3 粒度层次的第 1 个小波窗口.

2.2 多粒度小波变换树

针对数据流挖掘的特点,本文设计了一种基于内存的数据结构——多粒度小波变换树,可以实时计算数据流多粒度时变分形维数,有效降低动态环境下分形维数计算的复杂性.

定义 1(多粒度). 较小的粒度表示较小的时间跨度,大的时间粒度由小的时间粒度组成,由此构成了多粒度框架.假设 t 为基本时间单元,也是最小的时间粒度,则不同的时间粒度可用 $2^i\times k\times t(k=1,2,\ldots,i=0,1,\ldots)$表示,$i$ 表示粒度框架的层次,i 值越大,粒度越大,即时间跨度越大;$k\times t$ 表示基本粒度单元的时间跨度.

定义 2(多粒度小波变换树(multi-granularity wavelet transform tree,简称 MWTT)). MWTT 是一个二元组:$MWTT=(D,R)$,D 表示节点的集合,R 表示节点间的关系.每个节点由 3 个域组成:编号域、数据域和指针域.

- 编号域保存节点对应的小波窗口层次和序号;
- 数据域保存小波谱和窗口融合延拓等所需的相关信息;
- 指针域保存了节点间的关系,包含两个指针:一是指向同层次的相邻节点,二是指向融合延拓后生成的下一粒度的相关节点.

如图 3 所示,多粒度小波变换树的每一层节点都对应于图 2(a)所示的相应层次的小波窗口.树的叶子节点在树的最上层,保存的是最小粒度小波窗口的相关信息.同层相邻节点的合并可以生成更大粒度层次的小波窗口,具体的节点合并与小波窗口的层次关系可由定理 1 推导得出.

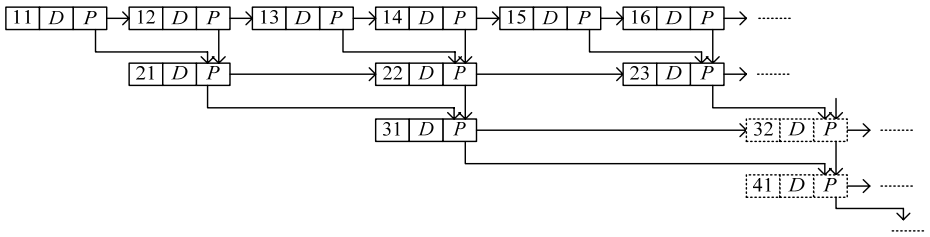


Fig.3 Multi-Granularity wavelet transform tree

图 3 多粒度小波变换树

如图 2(b)所示,多粒度小波变换树节点的构建和合并如下操作.

(1) 多粒度小波变换树节点的构建

首先,不考虑边界延拓的问题,计算第 1 个数据窗口和第 2 个数据窗口的 DB2 小波变换,如图得到实线框中的小波变换系数.同时,在数据域中保存以下信息:数据流窗口的首尾各两个数据以及第 1 层小波变换窗口的右端 3 个近似系数和左端 2 个近似系数、其他层小波窗口的首尾各两个近似系数,如图 2 所示,在数据窗口和小波变换窗口上的实线圈圈中的数据;然后,在数据域更新相关窗口的小波谱、相应的编号域和指针域的相关信息,即可构建相应的节点 1.

同理可构建节点 2.

(2) 多粒度小波变换树节点的合并

节点 1 和节点 2 的合并,需要利用数据域中保存的相关信息计算边界延拓系数,如图 2(b)所示,在小波变换窗口上的虚线圈中的数据.同时,计算新节点数据域需要保存的相关信息及其小波谱,并更新节点的编号和建立相关节点间的指针,从而实现了低粒度层次节点向高粒度层次节点的合并.

多粒度小波变换树的每个节点的数据域都保存了对应粒度下的时间序列数据流的 DB2 变换的小波谱,通过公式(2)可以计算出每个节点对应的分形维数,并且通过多粒度小波变换树的合并,可以计算更大粒度层次的节点对应的分形维数,从而可以获得时间序列数据流的多粒度时变分形维数.按照小波变换的定义,本文的变换方法并未对数据进行完全分解,但是窗口内数据的小波能量得到了完全分解.因此,计算的小波谱可以合理地反映数据的分形维数,相关实验证实了此方法的有效性,如后文实验 3.1 中的图 4 所示.

多粒度小波变换树的另一特点是动态增长性,树的规模随着数据流的实时到达而不断增大.随着树的节点和层次的不断增加,会对计算和存储带来较大的负担.假设最小粒度为 n(数据流窗口的大小),采用浮点型数据保存多粒度小波变换树节点内的信息,小波分解的层次为 $\log_2^n-2$,则依据以上分析,每个数据窗口保存 4 个数据,即,占用 8×4 个字节;第 1 层小波窗口保存 5 个近似系数,即,占用 8×5 个字节,其他各层小波窗口保存 4 个近似系数,即,占用 8×4 个字节,因此,小波窗口共占用 $8\times5+(\log_2^n-2)\times8\times4$ 个字节;小波谱则占用 $(\log_2^n-2)\times8$ 个字节.因此,一个最小粒度的多粒度小波变换树的节点共占用 $8\times4+8\times5+(\log_2^n-2)\times8\times4+(\log_2^n-2)\times8$,即 $40\times\log_2^n-8$ 个字节,记为 A.假设第 1 层共有 m(m 是 2 的整数倍)个节点,粒度层次每增加一层,数据域中就需保存相应的近似系数,并更新小波谱,因此,每高一粒度层次的节点需要多占用 40 个字节,则 $\log_2^m$ 层多粒度小波变换树总共需要占用 $(2\times m-1)\times A+(80\times m-\log_2^m-80)$ 个字节.因此,可根据内存情况动态调整节点数量,删除时间久远的一些小粒度节点,尽可能地保存当前节点.节点的删除可以提升算法的空间效率,但是直接删除会造成时间久远数据流的相关信息的丢失,即,无法计算数据在较小粒度下的分形维数.空间效率的提升与删除节点的层级是密切相关的:假设删除节点 j 的层级为 l,即,将节点 j 下的所有子节点都删除,用节点 j 表示其所有子节点,则节点 j 的子节点相对应的 $2^k\times n(k=0,\ldots,l-2)$范围内数据的 $l-1$ 个粒度下的分形维数都将统一由节点 j 所对应的 $2^{l-1}\times n$ 范围内数据的一个粒度下的分形维数表示.即,节点 j 对应数据的分形维数的表示精度降低了 2^{l-1} 倍,而相应的内存占用减少了 $\sum_{k=0}^{l-2}2^{l-(k+1)}\times(A+k\times40)$ 字节,可以有效地提升算法的空间效率.

另一种方法是固定内存中的小波变换树的节点数量,当多粒度小波变换树的节点数量超过上限时,将时间久远的数据对应的节点保存在硬盘上.将保存在硬盘上的节点分为两类:一类是已融合的节点,一类是未融合的节点.根据定理 1,可以搜索相应的未融合的节点,提取到内存中参与运算.但该方法在数据窗口较小、更新频繁时比较耗时,因此,本文更倾向于第 1 种牺牲精度换效率的直接删除的方法.

2.3 多粒度时变分形维数计算方法

多粒度时变分形维数计算方法分为 3 个步骤:首先,构建数据流上的滑动窗口,并计算相应子窗口数据的小波变换;然后,在内存中构建多粒度小波变换树,并对其进行实时更新和维护;最后,计算不同粒度下的局部分形维数.

算法 1 详细描述了基于数据流融合延拓的小波变换算法,并可在 $O(N)$时间内得到相应的小波变换系数.

算法 1. 数据流融合延拓的小波变换算法(transform).

Input:S_n,S_{n+1},$h=(h_1,h_2,h_3,h_4)$,$g=(g_1,g_2,g_3,g_4)$ /*第 n 个和第 n+1 个数据流窗口和 DB2 小波的滤波器*/;

Output:SP_n,R_n /*第 n 个数据流窗口的小波谱和融合系数*/.

1. k=1;
2. $S_1=S_n$, $S_2=S_{n+1}$;
3. $DB2Tran(S_1,S_2)$\{
   i=1;
   $Htmp_1=Htmp_2=Gtmp_1=Gtmp_2=NULL$;
   For (j=1; j<| S_1|−2; $j=j$+2)\{
      $Htmp_1[i]=(S_1[j],S_1[j+1],S_1[j+2],S_1[j+3])\cdot h$;
      $Gtmp_1[i]=(S_1[j],S_1[j+1],S_1[j+2],S_1[j+3])\cdot g$;
      $Htmp_2[i]=(S_2[j],S_2[j+1],S_2[j+2],S_2[j+3])\cdot h$;
      $Gtmp2[i]=(S_2[j],S_2[j+1],S_2[j+2],S_2[j+3])\cdot g$;
      i++;
   \}
4. $Htmp_1[i]=(S_1[|S_1|-1],S_1[|S_1|],S_2[1],S_2[2])\cdot h$;
   $Gtmp_1[i]=(S_1[|S_1|-1],S_1[|S_1|],S_2[1],S_2[2])\cdot g$;
5. for m=1 to $|Gtmp|$
      $SP_n[k]=SP_n[k]+(Gtmp_1[m]^2/|Gtmp_1|)$;
6. $S_1=Htmp_1$;
   $S_2=Htmp_2$;
   k++;
7. if $k<\log_2(N)-1$
      $DB2Tran(S_1,S_2)$;
   else
      $R_n[]=Htmp_1$;
   \}

算法 1 中:

- 第 3 行计算窗口内数据与 DB2 小波滤波器的内积,并将计算得到的近似系数和小波系数分别保存在临时数组中;
- 第 4 行计算了窗口的边界系数,由第 n 个窗口的最后两个数据元素和第 n+1 个窗口的前两个数据元素组成的向量分别和 DB2 小波滤波器的内积计算出,并保存在相应的临时数组的最后一位;
- 第 5 行计算了第 k 层小波变换的小波谱;

- 在第 6 行中,为了进行第 k+1 层小波变换,将两个窗口的第 k 层小波变换的近似系数赋给相关参数,并在第 7 行中判断小波变换的层次条件:若未达到变换的最高层,继续递归计算小波变换;否则,将最高层的近似系数保存在 R 数组中,作为计算两个窗口融合的参数输入算法 2.

算法 2 说明了窗口之间的融合以及相关信息的计算,主要分为 4 个步骤:首先,在第 1 行计算窗口融合后的小波谱信息;之后,将每个窗口的小波变换的顶层近似系数合并,并对其进行小波变换,如第 2 行和第 3 行所列;然后,在第 4 行计算新生成的层次上的小波谱;最后,将融合窗口的顶层小波变换近似系数保存在数组 MR 中,作为下一次窗口融合的系数.算法 2 的时间复杂度主要由第 1 行控制,其他行的复杂度都为 $O(1)$.第 1 行的时间复杂度与数据窗口的小波变换层次 k 有关,而 $k=\log_2 N$,N 为数据流窗口包含的数据元素个数,因此,算法 2 的时间复杂度为 $O(\log_2 N)$.

算法 2. 数据流融合延拓的窗口融合算法(merge).

Input:SP_n,SP_{n+1} /*第 n 个和第 n+1 个窗口的小波谱*/;

Output:MSP,MR /*融合窗口的小波谱和融合系数*/.

1. for k=1 to $|SP_n|$
   $MSP[k]=(SP_n[k]+SP_{n+1}[k])/2$;
2. $R=R_n\cup R_{n+1}$;
   i=1;
3. for (j=1; j<|R|−2; j=j+2){
   $Htmp[i]=(R[j],R[j+1],R[j+2],R[j+3])\cdot h$;
   $Gtmp[i]=(R[j],R[j+1],R[j+2],R[j+3])\cdot g$;
   i++;
   }
4. for m=1 to $|Gtmp|$
   $MSP[k+1]=MSP[k+1]+(Gmtp[m]^2/|Gmtp|)$;
5. $MR=Htmp$;

算法 3 描述了多粒度小波变换树的构建和维护:第 1 行调用算法 1 对数据流窗口数据进行基于融合延拓的小波变换;第 2 行将第 1 行得到的第 1 粒度层次的小波窗口插入到树的第 1 层;第 3 行根据定理 1 决定窗口的融合顺序,生成不同层次的小波窗口,并将相关信息保存到树的相应位置的节点中.

算法 3. 多粒度小波变换树构建及维护算法.

Input:S_n,S_{n+1} /*第 n 个和第 n+1 个数据流窗口*/;

Output:多粒度小波变换树.

for n=1 to +∞{
1. $Transform(S_n,S_{n+1},h,g)$;
2. SP_n and R_n insert into MWTT's first level;
3. if (n mod 2)==0{
   for i=1 to $\log_2(n)$){
   Merge(SP and R of the $(n-2^{i-1})$th and $(n-2^{i-1}+1)$th windows);
   SP and R insert into $(n/2^i)$th node of MWTT's the (i+1)th level;
   }
   }
}

定理 2. 算法 3 的时间复杂度为 $O(nN)$,空间复杂度为 $O(n\log N)$.

证明:算法 3 的时间复杂度主要由两部分组成:首先是第 1 行对数据流窗口的小波变换,复杂度为 $O(nN)$,n

和 N 分别表示窗口的数量和大小;其次是第 3 行多粒度小波变换树的融合操作,随着窗口的不断融合,数据流划分的粒度会不断变大,窗口融合的复杂度也随之增加为 $O(\log(2^{i-1}\times N))$,i=1,2,...,$\log_2 n$,则第 3 行的时间复杂度为 $O(\log_2 n\times\log_2 N)$.因此,算法 3 总的时间复杂度为 $O(nN+\log_2(n+N))$,即 $O(nN)$.算法 3 的空间复杂度:第 i 层的每个节点的空间复杂度为 $O(\log_2(2^{i-1}\times N)+O(1))$,$O(\log_2(2^{i-1}\times N))$表示第 i 层的小波谱数据的复杂度,$O(1)$表示节点的融合系数的复杂度.n 个数据流窗口最多可以构建$\lceil\log_2 n\rceil$(小于等于 $\log_2 n$ 的整数)层节点,因此,总的空间复杂度为 $\sum_{i=1}^{\lceil\log_2 n\rceil}\frac{n}{2^{i-1}}[\log_2(2^{i-1}\times N)+O(1)]$,化简后约为 $O((n-1/2)\times\log_2 4N)$,即 $O(n\log N)$.

该定理表明:算法 3 的时间和空间复杂度与数据流窗口的数量和大小有着密切的关系,与数据流的大小呈线性关系,完全满足数据流挖掘的需求.算法 3 中第 3 行的主要工作是根据定理 1 对多粒度小波变换树进行更新操作,其复杂度为 $O(\log_2(n+N))$,其中,N 表示窗口的规模,n 表示窗口的数量.随着数据流的不断到达,窗口的数量 n 会不断变大,更新操作的总时间会不断增加.为了避免频繁的更新操作可能产生的计算力不足的情况,可以通过设置较大的窗口规模 N,从而减小窗口数量 n,减少多粒度小波变换树的更新次数.但是,窗口规模 N 的变大使得时间序列数据流挖掘粒度变大,可能会导致挖掘精度的下降,不能发现小粒度下的分形维数的变化规律,因此,数据流的挖掘必须要在挖掘效率和挖掘精度之间找到平衡.

最后,通过遍历小波变换树的各节点,对各节点内的小波谱数据进行分析,可以得到不同粒度下各时间段的分形维数,即多粒度时变分形维数.

3 实 验

下面通过在真实和模拟的数据流上的实验来验证本文方法的有效性.本文所有实验结果均在 MS Windows XP 系统和 P4 2.0G CPU,3G RAM 环境下运行得到.

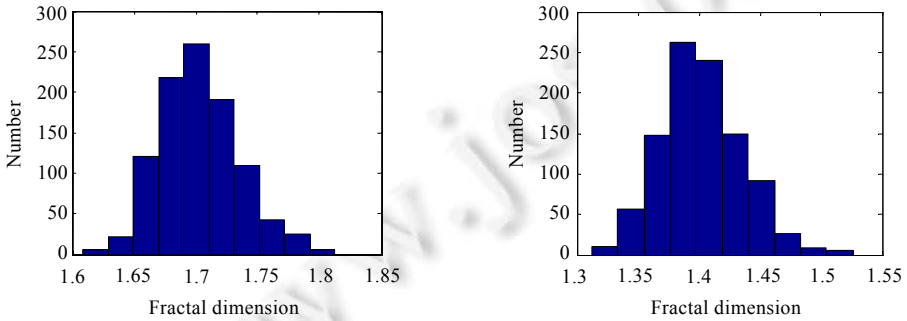
3.1 模拟数据流上相关实验

模拟数据流采用 Abry 和 Sellan 提出的分形布朗运动信号快速生成方法 wfbm[26,28]模拟具有不同分形维数的数据流.

(1) 精确度相关实验

实验数据由 wfbm 模拟 1 000 组长度为 10 000 的分形布朗运动数据序列,每组数据序列的分形维数都相同.

图 4 显示了利用本文算法估算的最大粒度的分形维数精度分布情况,图 4(a)和图 4(b)分别显示的是分形维数均为 1.7 和 1.4 的各 1 000 组长度为 10 000 的分形布朗运动数据序列的分形维数估算值的分布情况.分形维数为 1.7 的 1 000 组数据序列的分形维数计算的均值和方差为 1.706 4,0.001,分形维数为 1.4 的 1 000 组数据序列的分形维数计算的均值和方差为 1.402 1,0.001.



(a) 分形维数为 1.7 的数据流分形维数估计精度分布图 (b) 分形维数为 1.4 的数据流分形维数估计精度分布图

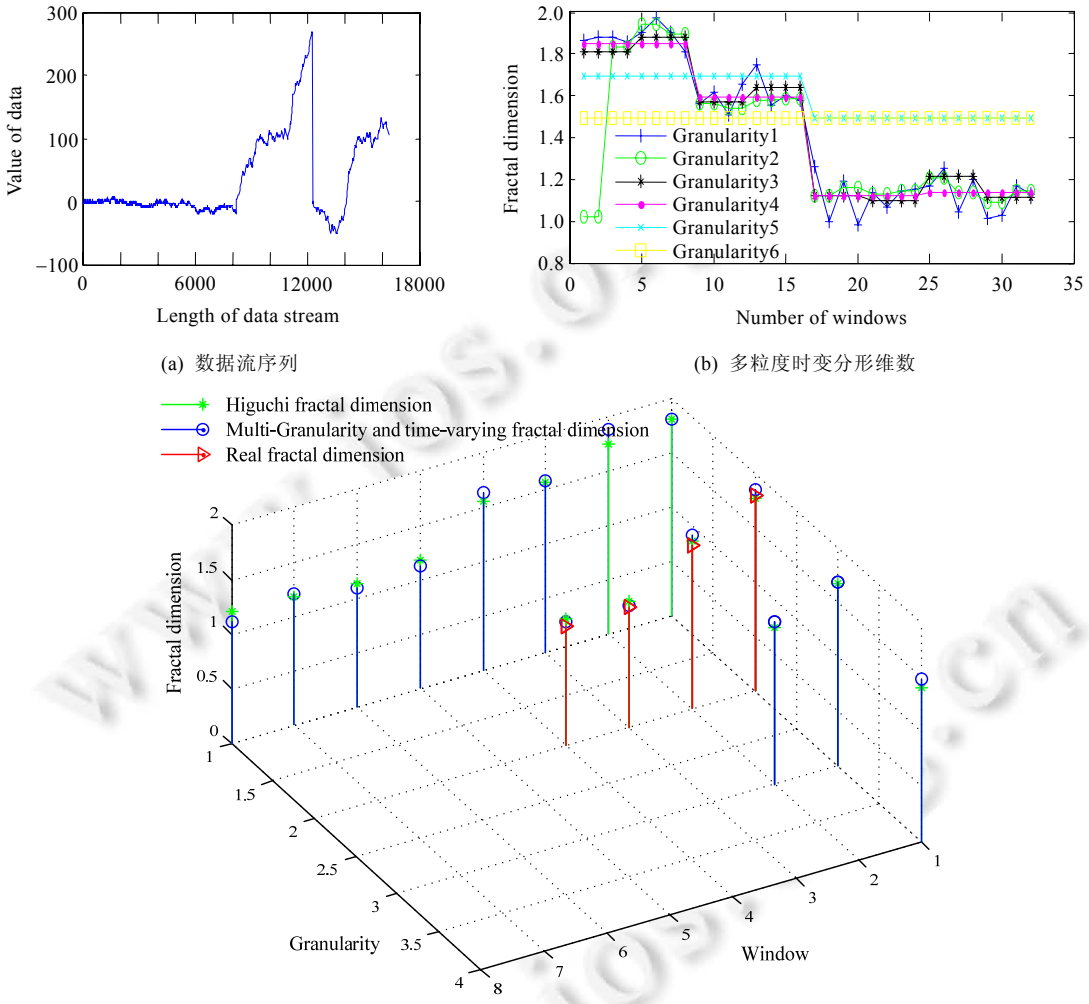
Fig.4 Accuracy histogram of fractal dimension on data streams

图 4 数据流分形维数估计精度分布图

本文方法计算出的模拟数据的分形维数与其真实的分形维数之间的误差较小,如图 4 所示,分形维数的计算值大致呈正态分布,且均值非常接近真实分形维数值,分形维数的计算值分布的方差很小.因此,本文方法计算出的分形维数能够以较高的置信度近似模拟数据的真实分形维数,其精度基本上可以满足数据流分形分析和挖掘的需求.

(2) 数据流多粒度时变分形维数监测相关实验

图 5(a)显示了长度为 16 384 的模拟数据流序列 S 的分布情况.S 由 4 个相同长度(4 096)的子序列组成,它们的分形维数分别为 1.8,1.5,1.1 和 1.1.



(a) 数据流序列 (b) 多粒度时变分形维数

(c) 多粒度时变分形维数与 Higuchi 分形维数比较图

Fig.5 Monitoring multi-granularity and time-varying fractal dimension on data stream

图 5 数据流多粒度时变分形维数监测

图 5(b)显示了 S 在不同粒度下的分形维数变化情况.S 被划分为 32 个数据窗口,其中,最低层粒度包含一个数据窗口,其大小为 512,每上一层粒度的大小都是下一层的 2 倍.如图所示.

- 在粒度 1~粒度 4 上,算法都能够清晰地区分出 4 个子序列,其中,在粒度 1 下,各子序列被划分为更小的子序列(即最小的粒度),其上的分形维数的变化较为剧烈,但总体上与真实的分形维数变化相一致;

- 但在粒度 5 和粒度 6 中,则无法识别出所有的子序列.粒度 5 和粒度 6 将不同的子序列划分到了同一个粒度下,只有一个当前粒度下的全局分形维数反映粒度的分形特征.

小粒度下的分形维数反映了更多的细节信息,但规律性较差;大粒度下的分形维数反映了整体特征,但缺乏对数据内部的深入洞悉.因此,从多粒度分析数据流的分形维数变化情况,能够从不同的尺度深刻地挖掘数据背后隐藏的知识,抓住系统变化的本质.

图 5(c)显示了多粒度时变分形维数与 Higuchi 方法计算的分形维数之间的比较情况.Higuchi 方法是针对时间序列型数据设计的具有较高精度的分形维数计算方法,但其只能计算静态时间序列数据.为了与本文方法进行比较,将 Higuchi 方法分别应用到不同粒度对应窗口中的时间序列上,获得各粒度上的分形维数.如图所示.

- 在不同粒度上的各窗口内,Higuchi 分形维数与多粒度时变分形维数都非常接近;
- 在第 2 粒度上,时间序列 S 被分为 4 个窗口,每个窗口内的时间序列都有一个理论上的真实分形维数,如图所示,Higuchi 方法和本文方法计算出的分形维数与真实分形维数的误差较小.

因此,本文提出的基于小波变换的多粒度时变分形维数在具有较高的计算精度的同时,可以从不同粒度实时监控时间序列数据流的分形维数的变化情况.

(3) 数据流多粒度时变分形维数计算方法性能相关实验

图 6 显示了本文方法随着数据流规模的不断扩大,在 5 个不同粒度下的计算性能变化情况.

- 图 6(a)显示了不同粒度下,小波变换树的内存占用量的变化情况.

在大粒度下,小波变换树只保存了少量的信息,因此内存占用量较小,且随着数据流规模的不断扩大,其增量并不显著;在小粒度下,小波变换树保存了大量细节信息,因此内存占用量较大,随着数据流规模的不断扩大,其增量也较为显著.图中最左上方曲线显示了原始数据占用内存的情况,可以看出,本文设计的小波变换树在不同粒度下都可以有效地提取数据流的概要信息.

- 图 6(b)显示了数据流多粒度时变分形维数计算时间的变化情况.

计算的累计时间与数据流规模的扩大呈线性关系.

以上实验结果与上一节中的算法复杂度的理论分析相一致,因此,本文方法具有良好的性能,可以高效地计算数据流多粒度时变分形维数.

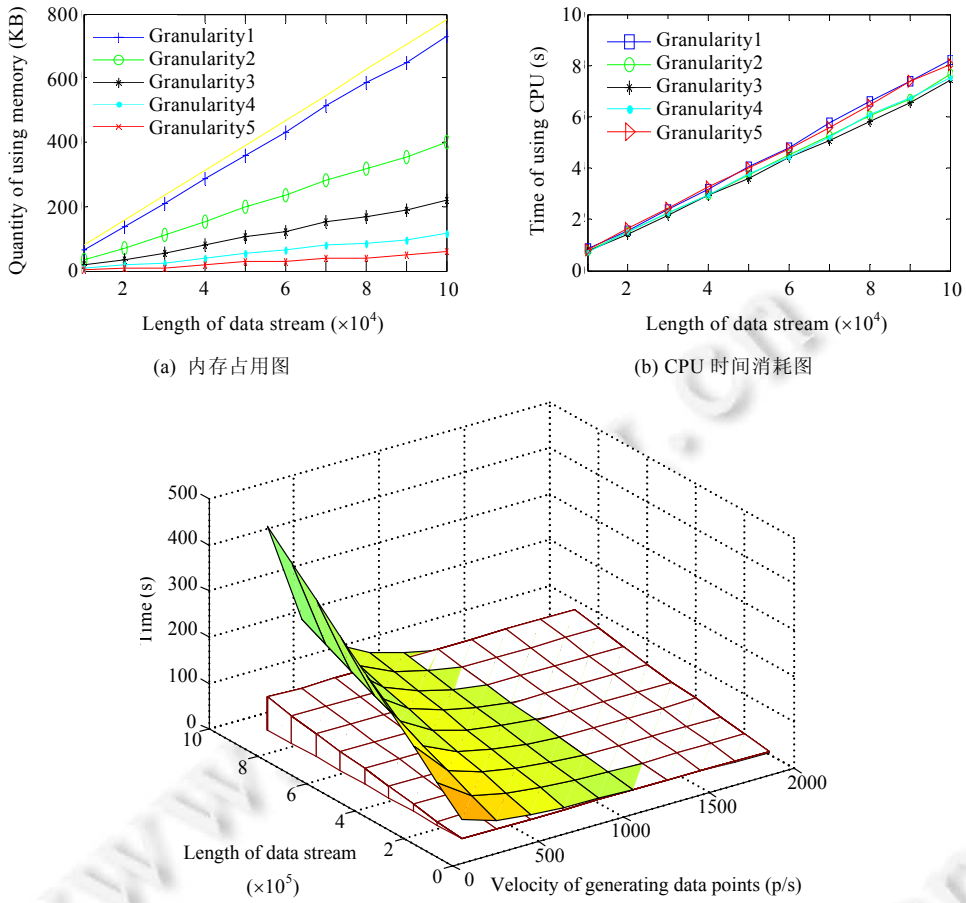
图 6(b)中一个有趣的现象是:粒度 5(最大粒度)和粒度 1(最小粒度)的曲线十分相近,粒度 2 和粒度 4 的曲线十分相近,粒度 3 所用的累计时间最少.造成这种现象的原因,可以通过定理 2 分析得出:多粒度小波变换树的构建和维护的时间复杂度为 $O(nN+\log_2(n+N))$,n 表示节点数量,N 表示窗口大小.相同数据流规模下:当粒度较大时,数据流窗口内的数据相对较多,即 N 较大和 n 较小,因此数据流的小波变换的时间延长,但是小波变换树的节点的更新次数减少;而粒度较小时,数据流窗口内的数据相对较少,即 N 较小和 n 较大,因此数据流的小波变换的时间缩短,但是小波变换树的节点的更新次数增加.因此,会出现较大粒度和较小粒度的计算时间相近的现象.

- 图 6(c)通过不断扩大数据流的规模和提高数据流的产生速度,测试了本文算法的适应性.

图 6(c)中,深色的曲面是由数据流的(产生速度、规模、消耗时间)三维属性所构成的,图中底部的浅色锥形体表示本文算法计算相应速度和规模的数据流所耗用的时间.通过对模拟数据的计算时间的统计与模拟数据产生速度和规模的统计之间的比较,可由图 6(c)得出以下结论:

- 本文算法在数据流产生速度为每秒 1 200 个数据点时,可以很好地处理不同规模的数据流;
- 但在速度超过每秒 1 200 个数据点后,本文算法的计算时间将超过数据流的产生时间,出现计算力不足的情况.

为此,本文在后续的研究中将考虑采用一些抽样方法以控制数据流的产生速度.



(a) 内存占用图 (b) CPU 时间消耗图

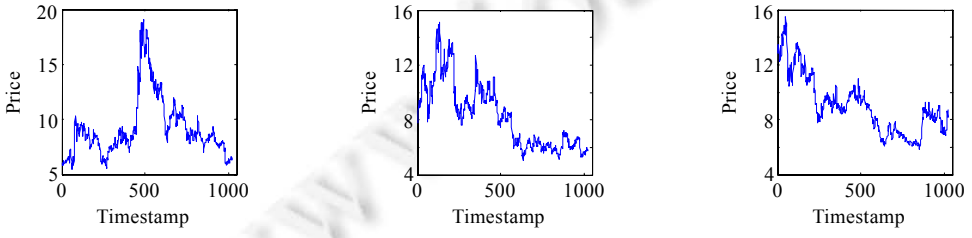
(c) 数据流多粒度时变分形维数计算方法的速率-规模适应性图

Fig.6 Performance of measuring multi-granularity and time-varying fractal dimension on data stream

图 6 数据流多粒度时变分形维数计算方法性能图

3.2 真实数据流上相关实验

本文将使用上海证券交易所的 3 支股票收盘价的真实数据来模拟数据流.3 支股票分别是氯碱化工(600618)、嘉宝集团(600622)和浦东金桥(600639).实验使用了从 2009 年 5 月 25 日~2013 年 8 月 21 日的 1 024 个交易日的收盘价数据,如图 7 所示.

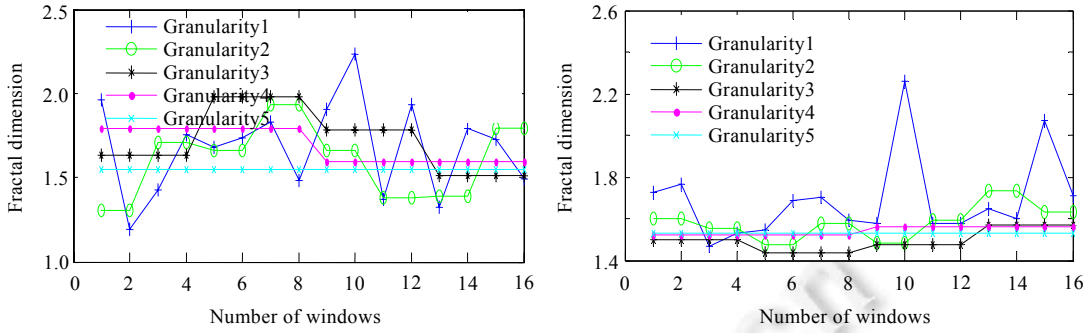


(a) 股票 600618 收盘价序列 (b) 股票 600622 收盘价序列 (c) 股票 600639 收盘价序列

Fig.7 Close price sequence of 600618, 600622 and 600639

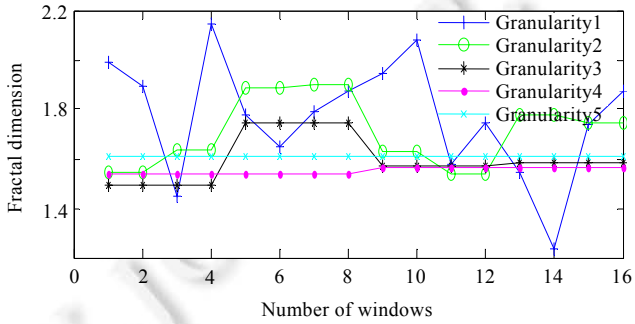
图 7 股票 600618,600622 和 600639 收盘价序列

图 8 显示了相关数据流序列的多粒度时变分形维数变化情况.



(a) 股票 600618 的多粒度时变分形维数

(b) 股票 600622 的多粒度时变分形维数



(c) 股票 600639 的多粒度时变分形维数

Fig.8 Multi-Granularity and time-varying fractal dimension of 600618, 600622 and 600639

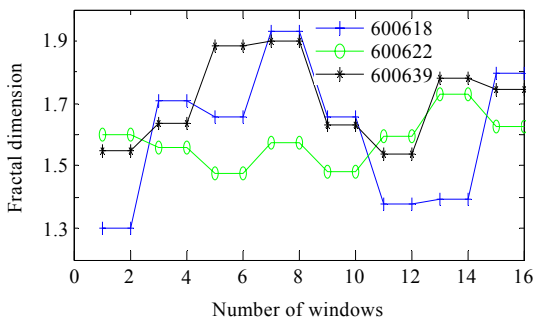
图 8 股票 600618,600622 和 600639 的多粒度时变分形维数

在图 8 所示的最小的粒度 1 上,分形维数在某些数据窗口上出现了估计值大于 2 的情况(由于本文研究的是一维时间序列数据流,因此分形维数 D 的取值范围是 $1<D<2$).针对这种情况,本文通过分析不同粒度上的小波谱,并从中找出了原因:最小粒度的取值太小,以至于数据流窗口内包含的数据不具有明显的分形特征,从相关窗口中计算出的小波谱不具有尺度不变性的特点,因此,关于小波谱和对应尺度间的双对数曲线的斜率拟合存在较大的误差,从而使得计算出的分形维数出现较大的误差.

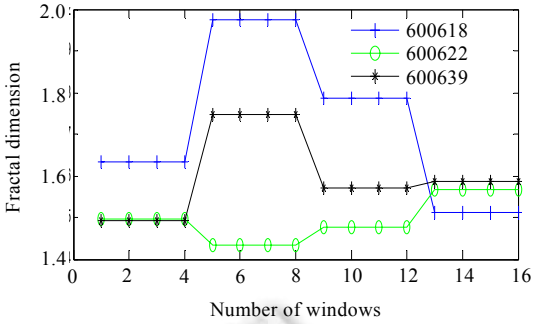
图 9 显示了不同数据流序列的时变分形维数在相同粒度下的时变特征.

- 图 9(a)显示:在粒度 2 上,3 支股票分形维数的变化曲线都具有较明显的双峰形状,但是变化的幅度差异较大,其中,氯碱化工和浦东金桥的分形维数在变化趋势和数值上更相似.
- 图 9(b)显示:在粒度 3 上,数据流序列被划分为 4 个部分,氯碱化工的分形维数是(1.591,1.859,1.714,1.536),嘉宝集团的分形维数是(1.547,1.529,1.548,1.563),浦东金桥的分形维数是(1.471,1.807,1.641,1.529),氯碱化工和浦东金桥的分形维数的时变特征在粒度 3 上非常相近;
- 而在粒度 4 上,数据流序列被划分为两部分,氯碱化工的分形维数是(1.732,1.604),嘉宝集团的分形维数是(1.575,1.590),浦东金桥的分形维数是(1.562,1.577),因此,嘉宝集团和浦东金桥不仅在分形维数值上更相近,而且分形维数的时变特征也一致(都是右半部分的分形维数比左半部分的大 0.015),如图 9(c)所示.

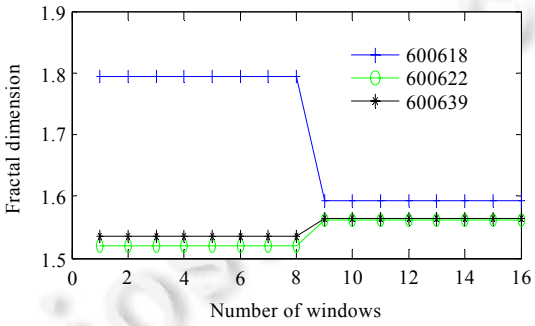
在粒度 1 上,分形维数的变化幅度较大,而且没有明显的规律可循,但反映了数据流上分形维数最细致的变化.氯碱化工、嘉宝集团和浦东金桥的整体(粒度 5)分形维数分别为 1.541,1.559 和 1.621.在粒度 5 上,氯碱化工和嘉宝集团更相近.



(a) 股票 600618,600622 和 600639 在粒度 2 上的分形维数



(b) 股票 600618,600622 和 600639 在粒度 3 上的分形维数



(c) 股票 600618,600622 和 600639 在粒度 4 上的分形维数

Fig.9 Compare the fractal dimensions of 600618, 600622 and 600639 on granularity-2,3 and 4

图 9 股票 600618,600622 和 600639 在粒度 2~粒度 4 上的时变分形维数比较

图 8 和图 9 显示了数据流在不同的粒度上可以展现出不同的特征及演化规律,因此,多粒度时变分形维数是一个多层次、多角度深入刻画数据流特性的方法,既可以计算大粒度上的分形维数,宏观地分析数据流的演化规律;又可以计算小粒度上的局部分形维数,微观地探寻数据流中隐藏的模式.

以上实验验证了本文提出的多粒度时变分形维数计算方法的有效性,其具有以下特点:

① 实时在线计算数据流分形维数,快速、准确地定位数据流分形维数发生变化的粒度和时间;

② 实时监测不同粒度下数据流分形维数的变化情况,多层次分析数据流演化规律;

③ 算法具有线性复杂度,满足数据流挖掘的单遍扫描数据的特点;

④ 通过访问内存中的多粒度小波变换树,可以计算任意粒度下的任意时段的分形维数,并可以根据内存的使用情况动态调整多粒度小波变换树的大小.

4 结束语

本文讨论了数据流多粒度时变分形维数的概念,提出了一种基于小波变换技术的数据流多粒度时变分形维数计算方法,介绍了数据流窗口的小波变换边界融合延拓方法与多粒度小波变换树的构建和动态维护方法.同时,通过实验验证了本文方法的有效性.我们下一步的工作将从以下几方面进行:首先,进一步完善本文方法,如粒度大小的选择、小波谱尺度范围的选择,并结合实际应用,考虑更加合理的数据流偏斜窗口划分方法等;其次,与其他数据流挖掘方法相结合,深入研究基于该方法的数据流趋势检测、相似性计算等数据流挖掘任务;三是将本文方法应用到真实的数据流分析领域,解决实际问题.

致谢 在此,作者衷心感谢审稿人的批评和指导.

References:

[1] Faloutsos C. Data mining using fractals and power laws. In: Dong GZ, Lin XM, Wang W, Yang Y, Jeffrey XY, eds. Proc. of the 9th Joint Int'l Conf. on Asia-Pacific Web Conf. (APWeb) and the 8th Int'l Conf. on Web-Age Information Management (WAIM). Springer-Verlag, 2007. [doi: 10.1007/978-3-540-72524-4_1]

[2] Barbara D, Chen P. Tracking clusters in evolving data sets. In: Russell I, Kolen JF, eds. Proc. of the 14th Int'l Florida Artificial Intelligence Research Society Conf. Key West: AAAI Press, 2001. 239−243.

[3] Qin SK, Qian WN, Zhou AY. Fractal-Based algorithms for burst detection over data streams. Ruan Jian Xue Bao/Journal of Software, 2006,17(9):1969−1979 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/17/1969.htm

[4] Folino G, Pizzuti C, Spezzano G. An adaptive distributed ensemble approach to mine concept-drifting data streams. In: O'Conner L, ed. Proc. of the 19th IEEE Int'l Conf. on Tools with Artificial Intelligence (ICTAI 2007). Patras: IEEE Computer Society, 2007. 183−188. [doi: 10.1109/ICTAI.2007.51]

[5] Ni ZW, Ni LP, Liu HT, Jia RY. Dynamic Data Mining. Beijing: Science Press, 2010 (in Chinese).

[6] Ni LP, Ni ZW, Wu H, Ye HY. Feature selection method based on fractal dimension and ant colony optimization algorithm. Pattern Recognition and Artificial Intelligence, 2009,22(2):293−298 (in Chinese with English abstract). [doi: 10.3969/j.issn.1003-6059.2009.02.019]

[7] Hippenstiel R, El-Kishky H, Radev P. On time-series analysis and signal classification—Part I: Fractal dimensions. In: Matthews MB, ed. Proc. of the 38th Asilomar Conf. on Signals, Systems and Computers. IEEE Computer Society, 2004. 2121−2125. [doi: 10.1109/ACSSC.2004.1399541]

[8] Rangayyan RM, Nguyen TM. Pattern classification of breast masses via fractal analysis of their contours. In: Lemke HU, Inamura K, Doi K, Vannier MW, eds. Proc. of the Int'l Congress Series, Vol.1281. Berlin: Elsevier Science, 2005. 1041−1046. [doi: 10.1016/j.ics.2005.03.329]

[9] Barbara D, Chen P. Fractal mining-self similarity-based clustering and its applications. In: Maimon O, Rokach L, eds. Data Mining and Knowledge Discovery Handbook. Springer-Verlag, 2010. 573−589. [doi: 10.1007/978-0-387-09823-4_28]

[10] Chen Q. Application on mining association rules with attributes reduction based on fractal dimension. In: Zhu RB, Ma Y, eds. Proc. of the Information Engineering and Applications. London: Springer-Verlag, 2012. 1288−1296. [doi: 10.1007/978-1-4471-2386-6_171]

[11] Sadikin M, Wasito I. Fractal dimension as a data dimensionality reduction method for anomaly detection in time series. In: Proc. of the 7th Int'l Conf. on Information & Communication Technology and Systems (ICTS 2013), Vol.105. Bali, 2013. 110. http://icts.if.its.ac.id/openaccess/2013/files/PP_19_PAPER_7.pdf

[12] Ni ZW, Xiao HW, Wu ZJ, Xue YJ. Attribute selection method based on improved discrete glowworm swarm optimization and fractal dimension. Pattern Recognition and Artificial Intelligence, 2013,26(12):1169−1178 (in Chinese with English abstract). [doi: 10.3969/j.issn.1003-6059.2013.12.011]

[13] Ni LP, Ni ZW, Gao YZ. Stock trend prediction based on fractal feature selection and support vector machine. Expert Systems with Applications, 2011,38(5):5569−5576. [doi: 10.1016/j.eswa.2010.10.079]

[14] Tasoulis SK, Maglogiannis I, Plagianakos VP. Fractal analysis and fuzzy c-means clustering for quantification of fibrotic microscopy images. Artificial Intelligence Review, 2014,4(3):313−329. [doi: 10.1007/s10462-013-9408-9]

[15] Popescu AL, Popescu D, Ionescu RT, Angelescu N, Cojocaru R. Efficient fractal method for texture classification. In: Proc. of the 2nd IEEE Int'l Conf. on Systems and Computer Science (ICSCS 2013). Villeneuve d'Ascq: IEEE, 2013. 44−49. [doi: 10.1109/IcConSCS.2013.6632021]

[16] Matsubara Y, Li L, Papalexakis E, Lo D, Sakurai Y, Faloutsos C. F-Trail: Finding patterns in taxi trajectories. In: Pei J, Tseng VS, Cao LB, *et al*., eds. Proc. of the Advances in Knowledge Discovery and Data Mining. Berlin, Heidelberg: Springer-Verlag, 2013. 86−98. [doi: 10.1007/978-3-642-37453-1_8]

[17] Holden T, Ye JM. Human Y-chromosome gene classification using fractal dimension & Shannon entropy. arXiv preprint arXiv: 1404.2540. 2014. http://arxiv.org/abs/1404.2540

[18] Yan GH, Li ZH, Dang JW. Finding natural cluster hierarchies based on MultiFractal. Ruan Jian Xue Bao/Journal of Software, 2008, 19(6):1283−1300 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/19/1283.htm [doi: 10.3724/SP.J.1001.2008.01283]

[19] Nunes SA, Romani LAS, Avila AMH, Coltri PP, Traina AJM, de Sousa EPM. Finding spatio-temporal patterns in multidimensional data streams. Journal of Information and Data Management, 2013,4(3):327−340.

[20] Chakrabarti D, Faloutsos C. F4: Large-Scale automated forecasting using fractals. In: Proc. of the 11th Int'l Conf. on Information and Knowledge Management. McLean: ACM Press, 2002. 2−9. [doi: 10.1145/584792.584797]

[21] Korn F, Muthukrishnan S, Wu Y. Fractal modeling of IP network traffic at streaming speeds. In: Liu L, Reuter A, Whang KY, Zhang JJ, eds. Proc. of the ICDE 2006. Atlanta: IEEE, 2006. 155. [doi: 10.1109/ICDE.2006.73]

[22] Nunes SA, Romani LAS, Avila AMH, Traina Jr. C, de Sousa EPM, Traina AJM. Fractal-Based analysis to identify trend changes in multiple climate time series. Journal of Information and Data Management, 2011,2(1):51−57.

[23] Nunes SA, Romani LAS, Avila AMH, Colti PP, Traina Jr. C, Cordeiro RLF, de Sousa EPM, Traina AJM. Analysis of large scale climate data: How well climate change models and data from real sensor networks agree. In: Proc. of the 22nd Int'l Conf. on World Wide Web Companion, Int'l World Wide Web Conf. on Steering Committee. Rio de Janeiro: ACM Press, 2013. 517−526.

[24] de Sousa EPM, Traina AJM, Traina Jr. C, Faloutsos C. Evaluating the intrinsic dimension of evolving data streams. In: Haddad H, ed. Proc. of the 2006 ACM Symp. on Applied Computing. Dijon: ACM Press, 2006. 643−648. [doi: 10.1145/1141277.1141426]

[25] Daubechies I. Ten lectures on wavelets. In: Proc. of the CBMS-NSF Regional Conf. on Series in Applied Mathematics. SIAM, 1992. http://www.openisbn.com/preview/0898712742/

[26] Abry P, Sellan F. The wavelet-based synthesis for fractional Brownian motion proposed by F. Sellan and Y. Meyer: Remarks and fast implementation. Applied and Computational Harmonic Analysis, 1996,3(4):377−383. [doi: 10.1006/acha.1996.0030]

[27] Ye ZX, Cao YJ. Application of Hurst exponent in analysis of stock market's efficiency. Systems Engineering, 2001,19(3):21−24 (in Chinese with English abstract). [doi: 10.3969/j.issn.1001-4098.2001.03.005]

[28] Bardet JM, Lang G, Oppenheim G, Philippe A, Stoev S, Taqqu MS. Semi-Parametric estimation of the long-range dependence parameter: A survey. In: Doukhan P, Oppenheim G, Taqqu MS, eds. Proc. of the Theory and Applications of Long-Range Dependence. Boston: Birkhauser Boston Inc., 2003. 557−577.

附中文参考文献:

[3] 秦首科,钱卫宁,周傲英.基于分形技术的数据流突变检测算法.软件学报,2006,17(9):1969−1979. http://www.jos.org.cn/1000-9825/17/1969.htm

[5] 倪志伟,倪丽萍,刘慧婷,贾瑞玉.动态数据挖掘.北京:科学出版社,2010.

[6] 倪丽萍,倪志伟,吴昊,叶红云.基于分形维数和蚁群算法的属性选择方法.模式识别与人工智能,2009,22(2):293−298. [doi: 10.3969/j.issn.1003-6059.2009.02.019]

[12] 倪志伟,肖宏旺,伍章俊,薛永坚.基于改进离散型萤火虫群优化算法和分形维数的属性选择方法.模式识别与人工智能,2013,26(12):1169−1178. [doi: 10.3969/j.issn.1003-6059.2013.12.011]

[18] 闫光辉,李战怀,党建武.基于多重分形的聚类层次优化算法.软件学报,2008,19(6):1283−1300. http://www.jos.org.cn/1000-9825/19/1283.htm [doi: 10.3724/SP.J.1001.2008.01283]

[27] 叶中行,曹奕剑.Hurst 指数在股票市场有效性分析中的应用.系统工程,2001,19(3):21−24. [doi: 10.3969/j.issn.1001-4098.2001.03.005]

倪志伟(1963－),男,安徽桐城人,博士,教授,博士生导师,主要研究领域为数据挖掘,智能管理,决策支持系统.

胡汤磊(1984－),男,博士生,主要研究领域为数据挖掘,智能管理.

王超(1983－),男,博士生,主要研究领域为数据挖掘,智能管理.

倪丽萍(1981－),女,博士,副教授,主要研究领域为分形数据挖掘,商务智能.