基于局部聚类的轨迹数据流偏倚采样

王考杰[1,2] 郑雪峰[1] 宋一丁[2] 安丰亮[2]
(北京科技大学信息工程学院 北京 100083)[1] (总后勤部后勤科学研究所 北京 100071)[2]

摘 要 移动对象轨迹数据管理是移动计算领域的研究热点。通过采样技术构造数据流摘要是普遍采用的方法之一。传统的均匀采样往往容易丢失某些关键变化数据。利用轨迹数据流的局部连续性特征,提出一种基于滑动窗口的偏倚采样算法。算法将滑动窗口通过聚类划分成若干大小不一的基本窗口,并针对每个基本窗口给定一个采样率,对窗口内数据进行偏倚采样,从而形成数据流摘要。算法利用了轨迹数据的内在特征,因此具有较高的采样质量。最后,基于实际数据对算法进行了实验,结果证明了算法的有效性。

关键词 轨迹数据流,偏倚采样,局部聚类
中图法分类号 TP311 **文献标识码** A

Local Cluster Based Biased Sampling of Trajectory Stream

WANG Kao-jie[1,2] ZHENG Xue-feng[1] SONG Yi-ding[2] AN Feng-liang[2]
(School of Information Engineering, University of Science and Technology Beijing, Beijing 100083, China)[1]
(Logistics Scientific Institute, General Logistics Department, Beijing 100071, China)[2]

Abstract Managing trajectories of moving objects is a research focus in mobile computing. Building data synopses by sampling technologies is one of the widely used method. But traditional uniform sampling usually discard some significant points that reveal relative spatiotemporal changes. A novel biased sampling approach based on sliding window model was proposed utilizing the property of local continuity. Firstly, through local clustering, the sliding window was divided into various sized basic windows and sampling the data elements of a basic window using biased sampling rate, then forming trajectory stream synopses. This algorithm takes advantage of the intrinsic characteristics of trajectory stream and achieves superior approximation quality. The extensive experiments verified the effectiveness of our algorithm.

Keywords Trajectory stream, Biased sampling, Local cluster

1 引言

近年来,通讯技术与空间定位技术的迅猛发展带动了移动计算的快速普及,进而出现了大量基于位置的服务计算[1]。移动对象不断变化的空间位置数据,即运动轨迹,是一个具有时间特征的数据序列。这些数据具有明显的时变、无限的"流"特征,因此被称为数据流或流数据。基于数据流的计算是目前数据处理与分析挖掘领域的研究热点之一[2]。

流数据在动态环境中连续产生,具有巨大的数量、无限的流动和快速变化的行为。这样的数据不可能完全存储在数据库中,因此,选取部分数据临时存储在内存中,针对这些数据进行查询分析,获得近似但足够可信的结果显然是一种可行的选择。如何选择要保留的数据是一个首先要解决的问题。

在大多数情况下,从数据流中构造数据摘要是一种非常可行的方法。常用的数据摘要方法有:采样[3-6]、直方图[7]、小波[8,9]、草图[10]等。每种摘要结构都有其特点,至于选择何种摘要结构则依赖于所要解决的问题类型。

移动对象的轨迹可以看作是该对象所代表的点在一个连续的时间区间,其空间位置的不断变换而形成的时间序列数据。这样的数据具有明显的时空特征,即数据在时间上是连续的,在空间上也是连续的。

本文利用轨迹数据的这种自身特性,讨论了基于采样技术的轨迹数据流摘要构造问题,提出了一种基于滑动窗口的局部聚类偏倚采样算法,其通过维持滑动窗口内若干依据数据局部聚类特征而建的大小不一的偏倚样本,实现了轨迹数据流摘要的线性构造。其中,如何基于欧式距离对轨迹数据进行聚类以及如何基于聚类特征进行偏倚采样是本文的主要贡献。

2 背景知识

为方便起见,我们把本文中用到的符号列在表1中。

到稿日期:2010-06-21 返修日期:2010-11-10 本文受国家科技支撑计划重点项目(2006BAG01A07)资助。
王考杰(1972—),男,博士生,主要研究领域为数据流分析、数据挖掘,E-mail:ewagkj@tom.com;**郑雪峰**(1951—),男,硕士,教授,主要研究领域为数据流分析、网络安全;**宋一丁**(1959—),男,硕士,高级工程师,主要研究领域为数据仓库、数据集成、软件质量;**安丰亮**(1972—),男,硕士,工程师,主要研究领域为数据集成、软件质量。

表 1　本文中用到的符号

符号	描述
N	数据流长度
S	轨迹数据流
S_i	轨迹数据流元素
ω	滑动窗口大小
τ	欧式距离门限值
σ	偏倚采样率
$C=\{s_i,s_{i+1},\cdots,s_j\}$	t_i 至 t_j 时间内的数据构成的聚类
Pivot	聚类 C 的中心点
Radius	聚类 C 的半径
B_i	基本窗口
$\hat{B}_i$	基本窗口 B_i 构成的采样摘要窗口

2.1　轨迹数据流

移动对象可看作是欧式空间的一个点，其移动位置形式化描述为一个四元组：

$$s_i:\langle id,t_i,x_i,y_i\rangle, i\in 1,\cdots,N,\cdots$$

式中，t_i 是对象在(x_i,y_i)位置时的时间戳，id 为四元组标识。

对象的轨迹数据流定义为由四元组 s_i 构成的无边界时间序列。

2.2　滑动窗口

滑动窗口是数据流处理模型之一。在这种处理模型中，并不在内存中保留所有的数据项，而只保留给定时间区间内到达的最近 N 个数据项。

例如，在图 1 中，给定窗口大小 ω，只有最近的 ω 个数据项被保留下来，用于数据分析与挖掘。当新的数据项到达时，滑动窗口中最旧的数据项因过期而自动被丢弃。

目前已有许多基于滑动窗口模型进行数据流处理的相关研究[11]。

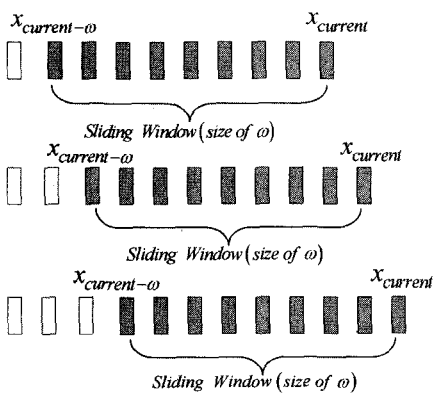

图 1　滑动窗口模型

2.3　距离

$dis(q,c)$是以基本窗口 q，c 为输入进行计算返回的一个非负数值，表示从 q 到 c 之间的距离。为不失一般性，本文中采用欧式距离作为距离测量方案。

给定基本窗口 X,Y，则：

$$dis(X,Y)=\sqrt{\sum_{i=1}^{\omega_b}(X_i-Y_i)^2}$$

3　基于局部聚类的数据流偏倚采样

本文提出一种基于滑动窗口的轨迹数据流摘要构造方法。算法的本质是基于“偏倚采样”的思想，将数据流聚类划分成若干基本窗口，针对每个基本窗口内数据的聚类特征，确定一个偏倚采样系数，对数据进行偏倚采样。将所有基本窗口内样本进行合并，便形成滑动窗口内的数据流摘要。

算法主要由两部分构成，即数据流局部聚类算法，以及基于局部聚类窗口的偏倚采样算法。

3.1　蓄水池采样

采样算法中，蓄水池采样[3]由于其简单、容易实现而得到广泛的应用，同时由于其单遍运算的特征，非常适合数据流摘要构造。

其算法描述如下所示。

算法 1　ReservoirSample(N,k)

输入：N，数据集或无限的数据流；k，蓄水池大小

输出：以 k/N 概率均匀采样形成的样本

1. 初始化 $Buffer$ 数据结构，用于缓存蓄水池数据流项
2. While(有新的数据流项)
3. 　初始化该数据流项 $s_i=\langle id,t_i,x_i,y_i\rangle$
4. 　计数器 $countOfItem$ 加 1
5. 　If $countOfItem<k$
　　//蓄水池未满
6. 　　将 $e_i=\langle x_i,w_i,t_i\rangle$缓存至 $Buffer$
7. 　Else
　　//蓄水池已满
8. 　　生成随机小数 r
9. 　　if $r<\frac{k}{countofItem}$
10. 　　计算 $p=k*r$
11. 　　以当前数据流元素替代 $Buffer[p]$
　　//以 r 概率替换蓄水池元素

3.2　偏倚采样

蓄水池采样本质上是一种均匀采样，均匀采样存在的最大问题是容易丢失某些关键数据。对密度变化较大的数据进行均匀采样，可能会丢失某些稀疏但重要的数据，同时又可能将大量不太重要的噪声数据包含进来。

针对均匀采样的问题，文献[12,13]提出了基于数据分布密度的偏倚采样算法，其通过定义每项数据的采样概率，使样本能够反映原始数据的分布密度特征，但只适用于静态数据。针对数据流的时变特性，文献[4]采用一个与时间密切关联的偏倚函数，实现了非均匀采样。算法考虑了“最近数据最重要”这一数据流固有特征，因此能够保证最近的数据采样质量最高，但同时也造成在一个固定时间段内(如滑动窗口)采样质量不一的问题。

为了解决这种矛盾，提出了基于局部聚类的偏倚采样方法。

3.3　基于局部聚类的偏倚采样

轨迹数据流有鲜明的自身特征，即其在局部是连续的，B. Yingyi 在文献[14]中依据该特性提出局部聚类的概念，用以检测轨迹数据流异常。本文借用其思想，将局部聚类应用于采样前的预处理，通过聚类将数据流元素按欧式距离聚类划分为长度不等的基本窗口。窗口大小有其现实语义，即窗口越大，表示大量类似的数据元素聚集在该窗口；窗口越小，则说明该部分数据变化较大。根据窗口大小，确定该窗口内的数据采样率。对于大窗口，需降低采样率，以节省存储空间，而对于小窗口，则适当增加采样率，以保留某些揭示轨迹重要变化的数据。

定义 1(局部聚类)　给定距离门限值 τ，以及时间约束 m_b，序列 $C=\{s_i,s_{i+1},\cdots,s_j\}$，$j-i+1\leqslant m_b$ 是数据流 S 的一个数据片断，如果 $\exists s_k\in C,\forall s_x\in C$，有 $dis(B_k,B_x)<\tau$，其中

B_k，B_x 指 C 中以 k，x 为结尾的数据片断，则 C 就是一个局部聚类。满足条件 $\forall s_i \in C, dis(B_x, B_i) < \tau$ 的数据点 s_x 定义为 C 的中心点，$\mathrm{Max}_{s \in C}(B_x, B_s)$ 定义为 C 的半径，时间戳早(晚)于 x 的数据序列定义为 s_x 的左(右)窗口。

为了实现采样率的自动调整，我们定义了一个参数偏倚率 σ，用来表征采样率与窗口的关系。

$$\sigma = \frac{\alpha}{B_i^e} \tag{1}$$

式中，B_i 是第 i 个窗口的大小，α 是一个与样本大小有关的变量。

假设滑动窗口内需要保留 M 个样本，g 为当前已有基本窗口数量，则 α 的取值可以由以下步骤计算得出：

$$M = \sum_{i=1}^{g} B_i \sigma = \sum_{i=1}^{g} B_i f(\frac{\alpha}{B_i^e}) \Rightarrow \alpha = \frac{M}{\sum_{i=1}^{g} B_i^{1-e}} \tag{2}$$

在算法执行开始，M 作为初始变量由用户指定，g 则随算法运行不断变化。

算法详细描述如下所示。

算法 2 CluReservoir(b, ω, τ, M)

输入：b，聚类窗口大小；ω，滑动窗口大小；τ，距离门限值；M，滑动窗口内需要保留的样本总量

输出：数据流摘要

```
1. 初始化 Buffer 数据结构，用于缓存当前聚类窗口未满时的数据流项
2. 初始化 SampleWindow 数据结构，用于存储滑动窗口内的所有采样摘要窗口
   初始化聚类中心点 Pivot=null
   初始化聚类半径 Radius=0
3. While(有新的数据流项)
4.     初始化该数据流项 s_i=<id,t_i,x_i,y_i>
5.     计数器 countOfItem 加 1
           //记录数据流项数量
6.     If CountOfSampleWindow <ω
           // 滑动窗口未满
7.       If countOfItem <b
           //聚类窗口未满
8.         将 s_i=<id,t_i,x_i,y_i> 缓存至 Buffer
9.       Else
             //当前聚类窗口 B_i 已满
10.        If Pivot=null
             //如果聚类中心点未发现
11.          计算 B_i 与 B_{i+1} 的距离 dis(B_i,B_{i+1})
12.          If dis(B_i,B_{i+1})>τ
13.            Pivot=newPivot(t_i)
                 //发现并初始化聚类中心点
14.          If dis(B_i,B_{i+1})<τ
                   并且 dis(B_i,B_{i+1})<Radius
15.            Radius=dis(B_i,B_{i+1})
               //保存聚类半径
16.        Else//发现聚类中心点
17.          计算 B_i 与 B_{i+1} 的距离 dis(B_i,B_{i+1})
18.          If dis(B_i,B_{i+1})<τ
                   并且 dis(B_i,B_{i+1})>Radius
19.            Radius=dis(B_i,B_{i+1})
20.          Else
21.            B_i=Buffer
               //当前基本窗口 B_i 已经构造完成
22.          根据式(2)计算 α
23.          根据式(1)、B_i 和 α 计算 σ
24.          根据 B_i 和 σ 计算 k
25.          Call ReservoirSample(B_i,k)
               //对聚类窗口内数据进行蓄水池采样，形成采样摘要窗口 B_i^
26.          计数器 countOfItem 减 1
27.      采样摘要窗口 S_d^(B_i) 缓存至 SampleWindow
28.      计数器 CountOfSampleWindow 加 1
29.      清空 Buffer
30.    Else
           //滑动窗口已满
31.      丢弃起始时间戳最久的数据
32.      计数器 CountOfSampleWindow 减 1
33.      更新 SampleWindow
```

4 实验分析

我们实现了局部聚类偏倚采样算法和常规蓄水池采样算法，并基于合成数据和实际数据对算法进行了实验。所有算法均以 Java 6 实现，运行环境为酷睿双核 1.6G，内存 1G。

4.1 试验指标

试验重点关注两个指标：采样质量、运算时间。

采样质量：

为了评估采样质量，使用线形插值方法从最终样本数据重构数据流，然后计算重构数据 $\hat{S}$ 和原始数据 S 在每一个时刻取值的均方差 $L_2(\hat{S}, S)$，作为质量评估指标。

$$L_2(\hat{S}, S) = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \| \hat{e}_i - e_i \|_2^2}$$

式中，

$$\| \hat{e}_i - e_i \|_2 = \sqrt{\sum_{a=1}^{d} (\hat{e}_i^a - e_i^a)^2}$$

式中，d 为轨迹数据的维度。

算法运算时间：

算法由于需要进行基于内存的数据删除操作，因此运算时间与采样率有直接关系。

4.2 试验数据

使用一组合成数据和一组真实数据对算法进行测试。合成数据使用 MATLAB 生成 Random Walk 数据集，包含 1000 组二维轨迹数据。

真实数据使用一组真实的 GPS 导航轨迹数据。该数据集记录了某人在 12 天内开车从家到办公室的的实际路线，包含 30000 组二维轨迹数据。

4.3 试验结果

图 2 至图 5 展示了算法的实验结果。

图 2 显示的是在合成数据下，采样质量与采样率之间的对比关系，总体来说，随着采样率的提高，样本重构误差缩小。从图中可以看出，当采样率在 5%～15% 区间时，CluReservoir 的重构误差只是 ReservoirSample 的 20%。随着采样率的提高，两种算法的误差缩小，但 CluReservoir 算法仍然明显优于 ReservoirSample 算法。

图 3 所示为两种算法在实际数据下的试验结果，反映了和图 4 类似的情况。

(下转第 185 页)

as Salience in Text Summarization[J]. Journal of Artificial Intelligence Research, 2004, 22: 457-479

[3] Passonneau R J. Applying the Pyramid Method in the 2006 Document Understanding Conference[C]// Proceedings of the 2006 Document Understanding Conference. Brooklyn, NY, 2006

[4] Salton G, McGill M J. Introduction to Modern Information Retrieval [M]. New York: McGraw-Hill Book Company, 1983: 400

[5] 李阜. 基于滑窗取词的单文档自动摘要技术研究[D]. 长沙: 国防科技大学, 2009

[6] Brin S. The Anatomy of a Large-scale Hypertextual Web Search Engine[J]. Computer Networks and ISDN Systems, 1998, 30: 1-7

[7] Kleinberg J M. Authoritative Sources in a Hyperlinked Environment[J]. Journal of the ACM, 1998, 46(5): 604-632

[8] Mani I. Summarization Evaluation: An Overview[C]// Proc. of the NTCIR Workshop Meeting on Evaluation of Chinese and Japanese Text Retrieval and Text Summarization. Tokyo: National Institute of Informatics. 2001

[9] Radev D, Teufel S, Saggion H, et al. Evaluation Challenges in Large Scale Multi-Document Summarization[C]// ACL 2003. Sapporo, Japan, 2003: 7-12

[10] Saggion H, Teufel S, Radev D, et al. Meta-evaluation of Summarization In a Cross-lingual Environment Using Content-based Metrics[C]// Proc. of the 19th COLING. TaiPei, 2002

[11] Lin C Y, Hovy E. Automatic Evaluation of Summaries Using N-gram Co-occurrence Statistics[C]// Proc. of Language Technology Conference. Edmonton, Canada, 2003

(上接第 137 页)

图 4 描述了基于 Random Walk 数据集，每处理一个数据流元素，两种算法的运算时间。从图中可以看出，运算时间随采样率的提高而增长，由于 CluReservoir 算法在 ReservoirSample 的基础上进行了聚类运算，因此运算时间相对较长，但仍然表现出时间线性的特征，说明算法适用于数据流处理场景。图 5 显示的基于大规模实际数据的实验结果也验证了这一点。

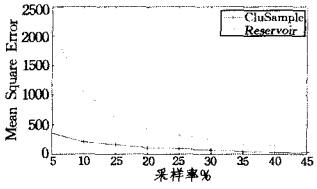

图 2 采样质量与采样率关系（合成数据，$\omega=256$）

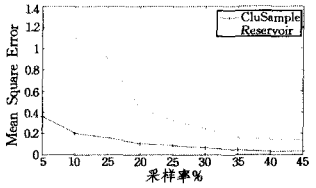

图 3 采样质量与采样率关系（真实数据，$\omega=10240$）

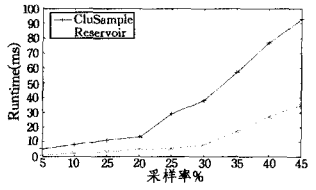

图 4 处理时间与采样率关系（真实数据，$\omega=256$）

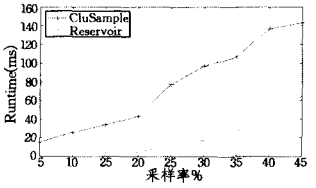

图 5 处理时间与采样率关系（真实数据，$\omega=10240$）

结束语 本文提出一种基于局部聚类的轨迹数据流摘要构造算法。该算法基于“偏倚采样”的思想，利用轨迹数据流局部连续性的特征，首先将指定时间范围内的数据流聚类划分成若干大小不一的数据窗口，并针对数据窗口大小定义偏倚系数，实现对窗口内数据的偏倚采样，从而构造数据流在滑动窗口内的摘要。由于算法充分考虑了数据的分布特征，对重要轨迹数据会自动提高采样率，因此，在平均采样率较低的情况下，其采样质量明显优于传统的蓄水池均匀采样。

参 考 文 献

[1] Sergio I, Eduardo M, Arantza I. Location-dependent query processing: Where we are and where we are heading[J]. ACM Comput. Surv., 2010, 42(3): 1-73

[2] Brian B, Shivnath B, Mayur D, et al. Models and issues in data stream systems[C]// Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems. Madison, Wisconsin, ACM, 2002

[3] Jeffrey S V. Random sampling with a reservoir[J]. ACM Trans. Math. Softw., 1985, 11(1): 37-57

[4] Charu C A. On biased reservoir sampling in the presence of stream evolution[C]// Proceedings of the 32nd international conference on very large data bases. Seoul, Korea, VLDB Endowment, 2006

[5] Kun-Ta C, Hung-Leng C, Ming-Syan C. Feature-preserved sampling over streaming data[J]. ACM Trans. Knowl. Discov. Data, 2009, 2(4): 1-45

[6] 张春阳，周继恩，钱权，等. 抽样在数据挖掘中的应用研究[J]. 计算机科学，2004，31(2)：126-128

[7] Arvind A, Singh M G. Approximate counts and quantiles over sliding windows[C]// Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems. Paris, France, ACM, 2004

[8] Panagiotis K, Nikos M. One-pass wavelet synopses for maximum-error metrics[C]// Proceedings of the 31st international conference on Very large data bases. Trondheim, Norway, VLDB Endowment, 2005

[9] Dimitris S, Antonios D, Timos S. Hierarchically compressed wavelet synopses[J]. The VLDB Journal, 2009, 18(1): 203-231

[10] Edith C, Haim K. Summarizing data using bottom-k sketches[C]// Proceedings of the twenty-sixth annual ACM symposium on principles of distributed computing. Portland, Oregon, USA, ACM, 2007

[11] Vladimir B, Rafail O, Carlo Z. Optimal sampling from sliding windows[C]// Proceedings of the twenty-eighth ACM SIGMOD-SIGACT-SIGART symposium on principles of database systems. Providence, Rhode Island, USA, ACM, 2009

[12] Christopher R P, Christos F. Density biased sampling: an improved method for data mining and clustering[C]// Proceedings of the 2000 ACM SIGMOD international conference on Management of data. Dallas, Texas, United States, ACM, 2000

[13] 余波，朱东华，刘嵩，等. 密度偏差抽样技术在聚类算法中的应用研究[J]. 计算机科学，2009，36(2)：207-209

[14] Yingyi B, Lei C, Wai-Chee F A, et al. Efficient anomaly monitoring over moving object trajectory streams[C]// Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining. Paris, France, ACM, 2009