计算机科学
COMPUTER SCIENCE

IntervalSketch:面向数据流的间隔项近似统计方法

陈昕杨, 陈翰泽, 周嘉晟, 黄家卿, 余佳硕, 朱龙隆, 张栋

引用本文

陈昕杨, 陈翰泽, 周嘉晟, 黄家卿, 余佳硕, 朱龙隆, 张栋. IntervalSketch:面向数据流的间隔项近似统计方法[J]. 计算机科学, 2024, 51(4): 4-10.

CHEN Xinyang, CHEN Hanze, ZHOU Jiasheng, HUANG Jiaqing, YU Jiashuo, ZHU Longlong, ZHANG Dong. IntervalSketch:Approximate Statistical Method for Interval Items in Data Stream[J]. Computer Science, 2024, 51(4): 4-10.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.231000226

IntervalSketch:面向数据流的间隔项近似统计方法

陈昕杨[1,2]　陈翰泽[1]　周嘉晟[1]　黄家卿[1]　余佳硕[1]　朱龙隆[1,2]　张　栋[2,3]
1 福州大学计算机与大数据学院　福州 350108
2 泉城省实验室　济南 250100
3 福州大学至诚学院　福州 350002
(chenxinyang1223@gmail.com)

摘　要　流式数据库在数据库中的占比逐渐增加,在流式数据库的数据流中提取所需信息是一项重要任务。文中研究了数据流的间隔项,并将其应用到了网络场景中。其中间隔项指在数据流中以固定时间间隔到达的元素对,这是第一项在数据流中定义和统计间隔项的工作。为了高效统计间隔项的 top-K,提出了 IntervalSketch。IntervalSketch 首先基于模拟退火对数据流分块以加快统计速度,其次利用 Sketch 进行间隔项的存储,最后通过特征分组存储策略降低 Sketch 存储间隔项的空间开销,提升了统计间隔项的精度。IntervalSketch 在两个真实数据集上进行了大量对比实验,实验结果表明,在同样内存的情况下,IntervalSketch 明显优于基线方案,其中处理时间为基线方案的 1/3～1/2,平均绝对误差、平均相对误差约为基线方案的 1/3。

关键词:Sketch;数据库;数据挖掘
中图分类号　TP311.13

IntervalSketch:Approximate Statistical Method for Interval Items in Data Stream

CHEN Xinyang[1,2],CHEN Hanze[1],ZHOU Jiasheng[1],HUANG Jiaqing[1],YU Jiashuo[1],ZHU Longlong[1,2] and ZHANG Dong[2,3]
1 College of Computer Science and Big Data,Fuzhou University,Fuzhou 350108,China
2 Quan Cheng Laboratory,Jinan 250100,China
3 Zhicheng College,Fuzhou University,Fuzhou 350002,China

Abstract　The proportion of streaming databases is gradually increasing,and extracting the required information in the data streams of streaming databases is an important task. In this paper,we study interval items which refer to pairs of elements arriving with a fixed interval,and apply them to network scenarios. It is the first work to define and count interval items in data streams. To efficiently count the top-K interval items,IntervalSketch is proposed. IntervalSketch firstly chunks the data stream based on simulated annealing to accelerate the statistical speed,secondly,it uses Sketch to store the interval items,and lastly reduces the memory of storing the interval items in Sketch through the feature grouping storage strategy,which enhances the accuracy of counting the interval items. Extensive comparative experiments are carried out on two real datasets. Experimental results show that IntervalSketch significantly outperforms the baseline solution with the same memory,and the processing time is 1/3～1/2 of the baseline solution,the average absolute error and the average relative error are 1/3 of the baseline solution.

Keywords　Sketch,Database,Data mining

1　介绍

1.1　背景与动机

数据流是一个按照时间递增顺序排列的无穷序列。与传统数据库相比,由于数据流具有无穷性,完全保存数据流过于浪费存储空间,同时数据流处理要求可以从大量的数据中快速地一次提取出所需的信息。然而,我们很难以线性速度处理高速数据流并报告数据流中蕴含的间隔项。为了在较短的时间内处理数据流,并获得近似结果,概率数据结构[1-3]凭借其误差小、内存小、速度快的特点而被广为接受。同时定义和发现新的模式一直是重要的研究热点,如周期项[4-5]、简单项[6]、稳定流[7]、突发流[8]、低频持续流[9]等。

本文定义了一种模式,称为数据流模型的间隔项。所谓间隔项,指以固定间隔到达的元素对,我们着重研究间隔项的频繁项问题与间隔项的查询问题。与此同时,将数据流场景推广到网络场景中,通过间隔项进行数据分析和网络测量。

到稿日期:2023-10-31　返修日期:2024-01-23
基金项目:国家重点研发计划专项(2023YFB2904000,2023YFB2904005);泉城省实验室(QCLZD202304);山东省实验室项目(SYS202201)
This work was supported by the National Key R&D Program of China (2023YFB2904000, 2023YFB2904005), Quan Cheng Laboratory (QCLZD202304) and Research Project of Provincial Laboratory of Shandong,China(SYS202201).
通信作者:张栋(zhangdong@fzu.edu.cn)

为了更好地理解间隔项，本文通过下面的例子进行举例说明。

例 1(消费模式) 在网络购物过程中，预测人的购买行为是一个经典的问题。一个人的购买、收藏、加入购物车等行为都会在系统数据库中留下记录，大多记录体现为流数据的形式。如一个人在购买 A 后，过一段时间后会购买 B，这便是间隔项的体现。现存预测方法如使用人工智能模型预测[10]无法跟上高速数据流。如果能及时发现消费数据中频繁的间隔项，那么在某个物品的购买量增加后，就可以预测在未来某个时间段某样物品的购买数量也会增加，商家就能及时调整商品的备货量。

例 2(传感器数据) 随着运动采集技术的发展和逐渐成熟，通过传感器从运动物体上采集数据是影响深远的技术之一，其中应用较多的是追踪动物行为[11]。其中追踪的信息包括动物的地理位置、行为、海拔等。在行为的追踪上，对于一只受追踪的动物，它的间隔频繁项如果预示它会在进食 10 h 后排便，而当进食和排便之间的时间关系出现混乱时，我们就有理由怀疑该动物的身体处于紊乱状态。

同时，值得一提的是，间隔项是一种具有普适性的流量模式，可以包含众多测量任务，适用于许多带有时间间隔的流量模式。在网络场景中间隔项可以用来优化带有时间间隔的流量模式，如检测周期项(当网络中流与流相同时)、突发流(当在时间间隔内发生突增突降时)、客户端对服务器的请求与服务器对客户端的及时响应、对延迟敏感的小流量进行流量分析[12]等。

1.2 解决方案

尽管之前会有一些关于时间序列的挖掘工作，但是其定义的场景与我们不同。与此同时，现有方案存在数据流处理效率低与空间利用率低的问题，本文将在 2.1 节和 2.2 节加以阐述。现有数据挖掘工作无法提供一种应用于数据流模型中、处理数据流效率高的挖掘算法。考虑到时间与空间的优化，由于间隔项具有时间跨度，可以通过将数据流模型拆分成多个测量周期进行测量。同时，间隔项是对元素关系的挖掘，因此元素关系相比元素的数量级是非线性级别的，即使采用 Sketch 进行关系的存储，处理哈希冲突的效率也较低。解决上述问题引申出两个挑战：1)如何将数据流拆分成多个测量周期以准确捕捉 top-K 的间隔项；2)如何在关系数据量大的情况下优化 Sketch 的空间开销。

针对上述问题，本文提出了一种数据挖掘方法，即 IntervalSketch。IntervalSketch 通过基于模拟退火的周期探索切分数据流，通过分组储存策略优化存储时引发的哈希冲突，本文第 4 章将会具体阐述这两部分的设计。

2 相关工作

本章简要介绍了一些与 IntervalSketch 相关的工作。虽然相关工作中存在与本文方法相似的问题和解决方案，但本研究是第一个专注于挖掘并存储间隔项的工作。

2.1 Sketch

Sketch 是一种利用哈希函数将大样本空间映射到小样本空间的数据结构，利用多行聚合的结构修正哈希函数带来的冲突，在网络、数据库、指令集[13]等场景中都有应用。CM-Sketch[1]通过多个哈希函数将数据元素映射到二维数组中，并增加相应位置的计数，最后通过返回多个计数器的最小值来估计元素频率。Mv-Sketch[3]则是对 CM-Sketch 的改进，在二维数组的每个桶中使用了主票选算法(将在 4.2 节中描述)。在数据流模型的挖掘中，PeriodicSketch[4]对周期项进行了定义，它使用 Sketch 对元素的最新时间戳进行记录。在下一次该项到达时，与储存在桶中的时间戳相减后将其插入哈希表中。由于哈希表中存在哈希冲突，为了筛选可能成为 top-K 的周期项，PeriodicSketch 使用 Guaranteed Soft Uniform(GSU)替换策略。而本文提出的间隔项是对周期项的一种扩展。具体来说，周期项仅限于挖掘一个动作的模式，而间隔项则是挖掘两个动作(可以是相同动作)的规律，实现了测量范围的扩展。HyperCalm Sketch[5]则是对 PeriodicSketch 的优化。它主要分为两部分：1)提出了一种基于时间感知的 Bloom 过滤器，用于检测批次的开始；2)提出了一种增强的 top-K 算法，用于报告 top-K 周期性批次。

2.2 数据挖掘

数据挖掘领域中确实有着一些挖掘时间序列的算法，但遗憾的是它们的应用场景与本文方法有着很大的不同，而且需要解决的问题也与本文方法有着很大的不同。接下来将详细介绍相关算法。

在关联规则的挖掘[14]中有着相似的定义，用于发现数据集中的元素之间的关联性和依赖性。这些规则通常用于市场分析、销售预测、产品推荐等应用，以揭示不同元素之间的关系，帮助企业做出更好的决策。最经典的例子就是买尿布的爸爸总会顺手地买啤酒，将两者摆在一起，销量就会上升。在数据流模型中挖掘关联规则，则是对关联规则加入了时间概念。

Ticom[15]解决了在交织、噪声和不完全序列数据中挖掘项的周期性的问题，其时间复杂度达到 $O(n^2)$。在有时间戳概念的数据挖掘中，大多专注于挖掘子序列的频繁项，如通过分析序列中数据元素的时间间隔及其顺序来定义和发现新颖的有趣序列模式[16]。同时也提出了间隔的概念[17]，但其并未被应用到数据流模型。

最后在数据挖掘领域也有着许多算法，如 Apriori[18]，FP-tree[19]等。但遗憾的是，以上现有工作都无法适用于本文的场景，无法在数据高速到来时，使用较短的更新时间进行处理。同时这些算法难以做到一次处理后仍可以保留重要信息，而且其存在较大的空间开销或者不合适的存储策略。

3 问题定义

本章给出了一些相关的概念与问题的具体定义。

3.1 数据流模型

给定一个数据流 $S=\{e_1,e_2,e_3,\cdots,e_n\}$，到来时间 $T=\{t_1,t_2,t_3,\cdots,t_n\}$，$t_i$ 为 e_i 的时间戳。

1) https://tianchi.aliyun.com/dataset/649

3.2 间隔项

给定两个项e_i和e_i($i<j$),那么间隔就为t_j-t_i,间隔项表达为$\langle e_i, e_j, t_j-t_i\rangle$。

3.3 Top-K 问题

给定一个数据流,要解决的问题是找出 K 个出现频率最高的间隔项。需要注意的是,$\langle x, y\rangle$可能会有不同的间隔值,因此可能需要多次报告它们。

3.4 例子

给定一串数据流 $e=\{a,e,c,a,d,c\}$,时间戳为 $\{1,2,3,4,5,6\}$,其中 a 与 c 的时间戳间隔相差 2,一共出现 2 次,如果我们寻找最大的间隔项,则$\langle a,c,2\rangle$为我们所寻找的最频繁间隔项。

4 IntervalSketch

本章将展示本文的解决方案与两个关键技术:基于模拟退火的周期探索和特征分组存储策略。本文中的常用符号及其意义如表 1 所列。如图 1 所示,IntervalSketch 主要分为两部分:基于模拟退火的周期探索与特征分组存储策略。第一部分使用基于模拟退火的周期探索策略划分数据流;第二部分利用间隔项的特征进行筛选,再插入不同 Sketch,减少哈希冲突以优化存储空间。最后,为了找到 top-K 间隔项,我们只需遍历 IntervalSketch,找出频率最大的 K 个间隔项进行报告。

表 1 本文中的常见符号

Table 1 Commpn symbols in this paper

符号	意义
S	数据流
E_t	当前探索的范围上限
E_{t+1}	E_t 通过扰动生成的新范围上限
K	扰动次数
T	每次周期扰动大小
$f(\cdot)$	对该测量周期的评分
L_0	初始测量的时间范围
L_i	周期时间范围
L	统计的总时间范围
Num	该测量周期挖掘 top-K 信息量
TOT	当次周期测量时桶映射总数
n	统计的总项数
∂	模拟退火配置参数(大于 0)
Num_{Interval}	IntervalSketch 的分间隔数
$F_d(\cdot)$	IntervalSketch 第 d 行哈希函数
d	Sketch 行数
w	Sketch 列数
W_i	第 i 个可选择插入的 Sketch

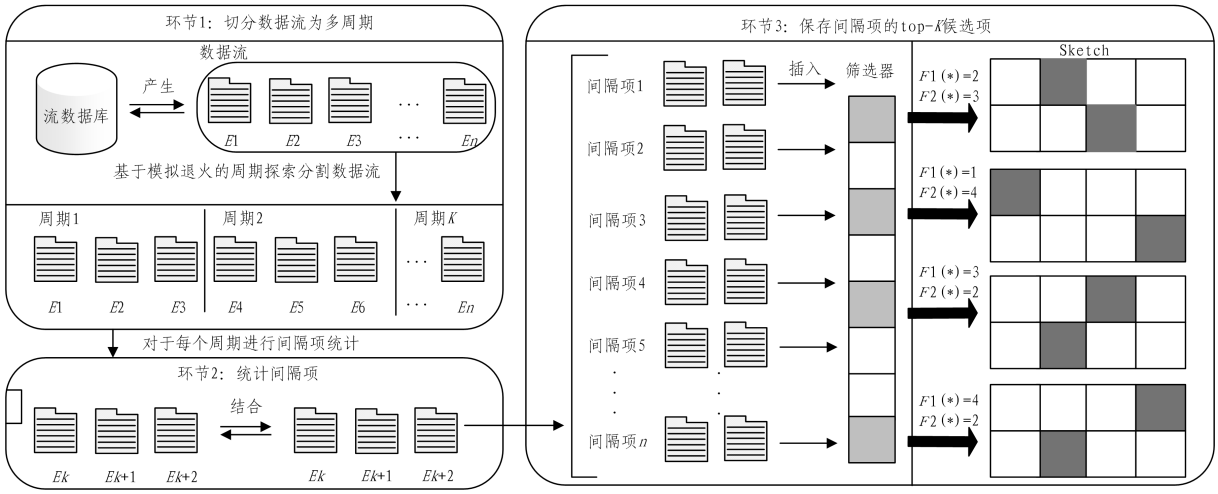

图 1 整体工作流程概述

Fig. 1 Overview of overall workflow

4.1 基于模拟退火的周期探索

在周期探索中,为了评价当前时间范围是否合理,本文引入了评分函数,评分函数与当前挖掘信息量和该周期测量的时间范围有关。

$f=-\alpha*L_i+\beta*Num$(α:β默认为 1:1)

而挖掘信息量 Num 是一个与周期测量的时间范围有关的变量,具体如下:

$$Num=a_n*L^n+a_{n-1}*L^{n-1}+\cdots+a_2*L^2+a_1*L+a_0$$

$$=\sum_{i=1}^{n}a_i*L^i$$

所以 $f=-\alpha*L_i+\beta*\sum_{i=1}^{n}a_i*L^i$

因此,评分函数是一个关于时间范围的单变量多峰函数。为了找到评分函数关于单次测量的时间范围L_i的全局最优值,我们用模拟退火算法寻找全局最优值。模拟退火算法可以在挖掘足够的间隔项 top-K 信息的同时,限制测量周期过度扩展。

4.1.1 测量周期中 f 的近似计算

$$f=-\alpha*\frac{L_i}{L}+\beta*Num=-\alpha*\frac{L_i}{L}+\beta*\sum\frac{TOT}{n*n}$$

由于周期测量的时间范围与 top-K 的挖掘信息量不在相同的尺度上,因此需要进行归一化。其中,L_i代表该测量周期的时间范围,Num 代表该测量周期内挖掘到的 top-K 挖掘信息量。随着时间推移,top-K 排名将会越来越稳定,在测量周期内记录间隔项映射入桶时,Num 加上该桶的计数除以 n 的平方,代表该间隔项对于 top-K 的贡献度。

4.1.2 Metropolis 准则

在模拟退火的启发式探索过程中,最优状态由初始状态迭代产生,然而启发式探索过程中产生的中间状态虽是局部最优状态,却容易被判定为全局最优状态。为了避免陷入局部最优状态,引入了 Metropolis 准则。Metropolis 准则主要分为两点:1)对于迭代过程中评分低的时间范围,有概率选择接受;2)随着模拟退火的进行,对于评分低的

时间范围接受概率逐渐降低。

更新状态的概率如下：

$$P=\begin{cases}1, & f(E_t)\leqslant f(E_{t+1})\\ e^{\frac{-\partial * K}{f(E_t)-f(E_{t+1})}}, & f(E_t)>f(E_{t+1})\end{cases}$$

基于模拟退火的周期探索的步骤如算法 1 所示。

算法 1 基于模拟退火的周期探索

输入：(L_0，T，S，f)

1. / * 初始化 * /
2. $E_t \leftarrow L_0$
3. While !S. empty()do
4. $E_{t+1} \leftarrow E_t + T$
5. K+=1
6. $\Delta E = f(E_{t+1}) - f(E_t)$
7. if E>0 do
8. $E_t \leftarrow E_{t+1}$
9. else do
10. R←Random()
11. $P = e^{\frac{-\partial * K}{f(E_t)-f(E_{t+1})}}$
12. if R≤P do
13. $E_t \leftarrow E_{t+1}$

4.1.3 实时处理应用

在处理数据流时，先根据算法 1 划定好测量周期的时间范围。当项实时到来时，判定是否超出给定的时间范围。如果未超过，便进行测量周期内的间隔项生成并插入 IntervalSketch，如果超过则准备进行新一轮数据流切分。

4.2 特征分组存储策略

本文采用紧凑的数据结构 Mv-Sketch 来解决空间爆炸的问题。在哈希冲突问题上，1）使用主票选算法来保留可能成为 top-K 的间隔项；2）如果简单地使用一个较大的 Sketch 存储间隔项$\langle x,y,interval\rangle$，与先按 interval 进行 Sketch 的选择后再存储$\langle x,y\rangle$相比，在内存不变的情况下，后者的哈希冲突率会低于前者。因此如图 1 中的环节 2 所示，对于一个间隔项$\langle x,y,interval\rangle$，现采用间隔将其索引到不同的 Sketch 中，再采用$\langle x,y\rangle$进行哈希计算。

插入（主票选算法）的步骤如下。首先利用间隔项中的特征 interval 进行 Sketch 选择。然后在该 Sketch 中的每个具有哈希函数$F_1(\cdot),F_2(\cdot),\cdots,F_d(\cdot)$的数组中选择槽位$W_{\text{interval}}[F_1(e)\%w],W_{\text{interval}}[F_2(e)\%w],\cdots,W_{\text{interval}}[F_d(e)\%w]$。最后，查看槽位中存储的候选者是否与本文的相同，如果相同，则将$W_{\text{interval}}[F_i(e)\%w]$增加 1；如果不同就将$W_{\text{interval}}[F_i(e)\%w]$减 1，为零后则更新存储键，并将$W_{\text{interval}}[F_i(e)\%w]$置为 1。

对于间隔项$\langle x,y,interval\rangle$，首先利用间隔项中的特征 interval 进行 Sketch 的选择，然后选择$W_{\text{interval}}[F_i(e)\%w]$中的最小值报告为间隔项的频率。当需要得知 top-K 时，仅仅需要遍历 Sketch 即可。特征分组存储策略如算法 2 所示。

算法 2 特征分组存储策略

输入：(x，y，interval，W)

输出：Ans

1. / * 初始化 * /
2. interval = interval % $\text{Num}_{\text{Interval}}$
3. Ans=inf
4. / * 更新 * /
5. for i←1 to d_W do
6. pos=$F_i[\langle x,y\rangle]$
7. $W_{\text{interval}}[\text{pos}]$. update()//更新主票选
8. / * 报告 * /
9. for i←1 to d_W do
10. Ans=min(Ans，$W_{\text{interval}}[F[\langle x,y\rangle]]$)
11. return Ans

5 数学分析

本章给出了 IntervalSketch 的总体时间、空间复杂度，以及单次切分统计的时间开销。

5.1 时间、空间复杂度分析

假设之前每次周期测量的项分别是$n_1,n_2,n_3,\cdots,n_k$，更新到目前为止的时间复杂度为：

$$O\left(\sum_{i=1}^{k}\frac{n_i^2}{2}\right)=O\left(\frac{\sum_{i=1}^{k}n_i^2}{2}\right)$$

依据平方展开关系：

$$O\left(\frac{\sum_{i=1}^{k}n_i^2}{2}\right)<O\left(\frac{n^2}{2}\right)<O(n^2)$$

空间复杂度为：

$$O(Num_{\text{Interval}} * d * w)$$

5.2 分组存储策略优化

本节将介绍分组存储策略优化哈希冲突的理论推导。当 k 个间隔项同时用一个桶进行计数时，称其为冲突，该 k 个间隔项的计数都会受到影响，因此将 k 定义为衡量桶中冲突的评价指标 J。

假设两两不相等的间隔项序列 $E=\{e_1,e_2,e_3,\cdots,e_k\}$，其中间隔项之间具有不同的间隔特征，使用 m 个桶（$m\leqslant k$）进行计数。将每个间隔项映射桶号记为 $B[*]$，桶中数量记为 $N[*]$。

当 k 个间隔项哈希映射入 m 个桶时，由哈希散列的均匀性可知：

$$\Pr(B[*]=i)=\frac{1}{m}(1\leqslant i\leqslant m)$$

由此可得出评价指标的数学期望为：

$$J_1=E(N(i))(1\leqslant i\leqslant m)=m\left(\frac{1}{m}*k\right)=k$$

采用分组存储策略后，假设将 k 个间隔项依据特征分为 y 组（$y\leqslant m$），则必有 y 个间隔项不存在冲突，则不妨先假设 k 个间隔项中有 y 个放置于不同的哈希测量桶中，剩余（$k-y$）个间隔项进行均匀哈希散列，则：

$$J_2=E(N(i))(1\leqslant i\leqslant m)$$
$$=y\left[\left(\frac{k-y}{m}+1\right)\right]+(m-y)\left(\frac{k-y}{m}\right)$$

当分组存储策略有效时，即：

$$J_2=E(N(i))(1\leqslant i\leqslant m)$$
$$=y\left(\frac{k-y}{m}+1\right)+(m-y)\left(\frac{k-y}{m}\right)$$

$$\leqslant J_1 = m\left(\frac{1}{m} * k\right) = k$$

解得 $0 < y \leqslant \min(m, k)$

仅当 $y=1$ 时取得 $J_2 = J_1$。若满足如下表达式：

$0 < y \leqslant \min(m, k)$

则对数据进行预处理特征提取后再进行分组数据插入，可以有效减少哈希冲突的发生。

6 实验结果

本章展示了 IntervalSketch 的实验结果。首先，6.1 节描述了实验设置；6.2 节描述了几个重要参数对 IntervalSketch 的重要影响；6.3 节在两个数据集上进行了与基线方案的对比。

6.1 实验设置

6.1.1 数据集

本文使用了两个真实数据集进行实验，即 CAIDA 数据集[20]和淘宝用户行为数据集[21]。在 CAIDA 数据集上，将每 5um 划定为一个时间段。将数据集切分为 2 000 数据量为一组的数据，进行多组实验后取平均值。由于间隔项具有衡量范围广的特点，因此验证 IntervalSketch 答案时选择每次报告频率在前 30 的间隔项。这里应该注意的是，由于我们检查的是 IntervalSketch 所报告的 top-K 是否正确，我们的假正类(FP)与假负类(FN)数量相等，PR 与 RR 是相等的，因此将 PR 与 RR 放在一起讨论。

6.1.2 度量指标

1) Accuracy(准确率)：指正确分类的间隔项在总体中的比例。

2) Precision Rate(精确率，PR)：指正确报告的间隔项的数量与报告的间隔项的数量的比值。

3) Recall Rate(召回率，RR)：指正确报告的间隔项个数与正确的间隔项个数的比值。

4) Average Absolute Error(AAE)：

$$AAE = \frac{1}{|Q|}\sum_{e_i \in Q} |f_i - \hat{f}_i|$$

Average Relative Error(ARE)：

$$ARE = \frac{1}{|Q|}\sum_{e_i \in Q} \frac{|f_i - \hat{f}_i|}{f_i}$$

其中，f_i 代表间隔项 e_i 的真实频率，$\hat{f}_i$ 为间隔项 e_i 的估计频率，Q 为查询集合。

5) Processing Time(处理时间，PT)：指算法处理数据流的时间。

6.1.3 基线解决方案

利用 Sketch 进行存储是一项较为常见且有效的策略，因此对数据流进行循环遍历后，再将间隔项插入 Sketch 中。最后通过 Sketch 的遍历查询 top-K 的间隔项。

6.2 参数影响

本节测试了几个关键参数(模拟退火周期探索初值、模拟退火每次跳跃步长)对 IntervalSketch 的 PR/RR 和 PT 的影响。

PT(Processing Time)如表 2 所列。本实验表明，随着模拟退火初值的增长，处理时间将会增加，但是随之而来的是精度的提升，因此考虑模拟退火周期初值时，需要与 PR/RR 紧密结合。

表 2 PT 随 L_0 的变化

Table 2 PT varies with L_0

L_0	30	40	50	60	70	80
PT	0.167076	0.221507	0.271149	0.346316	0.390016	0.484137

PR/RR 随模拟退火周期初值的变化如图 2 所示。本实验表明，L_0 最佳值为 50～70。PR/RR 随着初值增长而增长，最终在 70 后会趋近于一个稳定值。

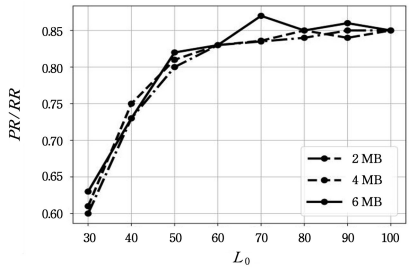

图 2 模拟退火周期初值对 PR/RR 的影响

Fig. 2 Effect of L_0 on PR/RR

PT 随 T 变化如表 3 所列。本实验表明，随着模拟退火跳跃步长的增长，处理时间将会增加，但是随之而来的是精度的提升。

表 3 PT 随 T 的变化

Table 3 PT varies with T

T	5	10	15	20
PT	0.3115	0.3409	0.3617	0.3765

PR/RR 随模拟退火跳跃步长的变化如表 4 所列。本实验表明，PR/RR 随着跳跃步长的增长而增长，最终在 15 后会趋近于一个稳定值。

表 4 模拟退火跳跃步长对 PR/RR 的影响

Table 4 Effect of T on PR/RR

T	5	10	15	20
PR/RR	0.731	0.767	0.831	0.832

PR/RR 随 Num_Interval 的变化如图 3 所示。本实验表明，Num_Interval 的最佳值设置在 10～13 之间。随着 Num_Interval 的增长，PR/RR 先增加后减少，这是因为流量通常为倾斜的，过高的 Num_Interval 会导致分类后的 Sketch 难以处理单个 Mv-Sketch 哈希冲突，而过低的 Num_Interval 无法有效分类间隔项，哈希冲突较为明显。

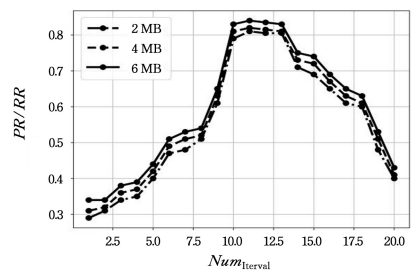

图 3 内存一定时 Num_Interval 对 PR/RR 的影响

Fig. 3 Effect of Num_Interval on PR/RR with given momery

6.3 对比实验

6.3.1 实验 1

本节将 IntervalSketch 的性能与基线解决方案进行比较。其中 IntervalSketch 与基线解决方案在内存相等的前提下进行比较。每次报告频率在前 30 的间隔项进行正确性分析。

Accuracy 如表 5 所列。当实验表明，在内存相等时，IntervalSketch 的 Accuracy 与基线方案相差不足 0.5%。

PT 如表 5 所列。本实验表明，IntervalSketch 平均是基线方案的 2/5。

AAE/*ARE* 如表 5 所列。本实验表明，IntervalSketch 平均是基线方案的 1/3。

表 5 *Accuracy*，*PT*，*AAE*，*ARE*

Table 5 *Accuracy*，*PT*，*AAE*，*ARE*

Algorithm	*Accuracy*	*PT*	*AAE*	*ARE*
Baseline	0.9953	0.4957	11.2667	0.0968
IntervalSketch	0.9910	0.1967	3.3667	0.0342

PR/*RR* 如图 4、图 5 所示。本实验表明，IntervalSketch 优于基线解决方案，IntervalSketch 的 *PR*/*RR* 值最好情况下比基线方案高出 30%，平均比基线方案高出 8.3% 左右。

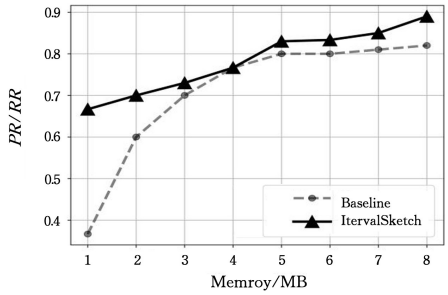

图 4 CAIDA

Fig. 4 CAIDA

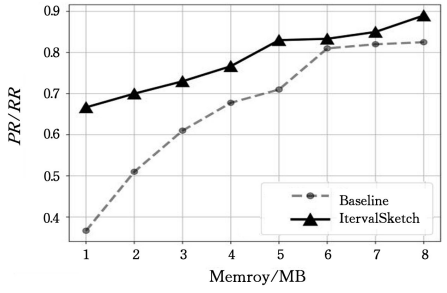

图 5 淘宝用户行为数据集

Fig. 5 Taobao user behaviour dataset

6.3.2 实验 2

本节研究了在使用内存一定时，处理数据量(Data)增大后对 *PT*，*Accuracy*，*AAE*，*ARE*，*PR*/*RR* 的影响。

PT 如图 6 所示。本实验表明，随着数据量的增大，IntervalSketch 的 *PT* 将呈现为较为均匀的线性级别增长，基线解决方案的 *PT* 将呈平方级别增长，在数据量范围为 2 000～5 000 时，IntervalSketch 约为基线方案的 2/5～3/11。指标还会随着数据量的增大而增大。

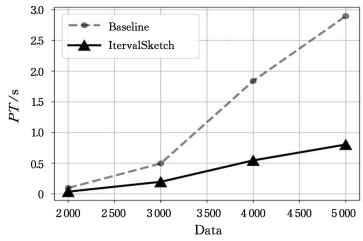

图 6 数据量增大对 *PT* 的影响

Fig. 6 Effect of data increasement on *PT*

Accuracy 如表 6 所列。实验表明，在内存相等时，随着数据量的增大，IntervalSketch 的 *Accuracy* 与基线方案相差不足 0.6%。

表 6 *Accuracy*

Table 6 *Accuracy*

Algorithm	*Accuracy*
Baseline	0.9982
IntervalSketch	0.9923

PR/*RR* 如表 7 所列。本实验表明，在内存相等时，随着数据量的增大，IntervalSketch 的 PR/RR 下降的程度比 Baseline 低，有着更强的稳定性。

表 7 *PR*/*RR*

Table 7 *PR*/*RR*

Algorithm	*Num*			
	2 000	3 000	4 000	5 000
Baseline	0.6153	0.6026	0.5802	0.5761
IntervalSketch	0.7153	0.6834	0.6734	0.6734

AAE 如图 7 所示。本实验表明，IntervalSketch 的 *AAE* 明显低于基线方案。而且随着数据量增长，IntervalSketch 的 *AAE* 增长速率明显低于基线方案，在 2 000～5 000 的数据范围内，IntervalSketch 的 *AAE* 是基线方案的 3.7～4.73 倍。

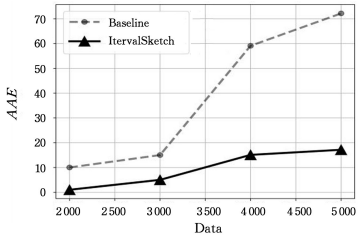

图 7 数据量增大对 *AAE* 的影响

Fig. 7 Effect of data increasement on *AAE*

ARE 如图 8 所示。本实验表明，IntervalSketch 的 *ARE* 明显低于基线方案。在 2 000～5 000 的数据范围内，IntervalSketch 的 *AAE* 是基线方案的 1.73～4.1 倍。

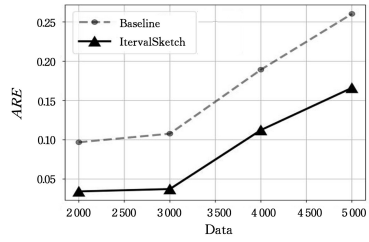

图 8 数据量增大对 *ARE* 的影响

Fig. 8 Effect of data increasement on *ARE*

结束语 IntervalSketch 研究了数据流模型中的间隔项，并将其应用到了网络场景中。这是第一项在数据流中定义和统计间隔项的工作。为了高效找到这些间隔项和统计间隔项的频繁项，提出了 IntervalSketch。为了获取 top-K 排名，使用了两种技术：基于模拟退火的测量周期探索和特征分组存储策略。最后基于两个数据集进行了大量实验，实验结果表明，在同样内存的情况下，IntervalSketch 明显优于基线方案，其中处理时间约为基线方案的 1/3～1/2，AAE 和 ARE 约为基线方案的 1/3，这两个指标均随着处理数据量的增大而大幅增大，同时其他指标相比基线方案也有 8.3%～30%的优化。当然本文工作还有许多不足的地方，如在间隔项存储时存在大量的哈希冲突，主票选算法表现并不够优秀，我们计划在未来将重心放在大量哈希冲突的解决与主票选算法的优化上。

参 考 文 献

[1] CORMODE G, MUTHUKRISHNAN S. An improved data stream summary: The count-min sketch and its applications [C]// LATIN 2004: Theoretical Informatics: 6th Latin American Symposium. Berlin Heidelberg: Springer, 2004: 29-38.

[2] YANG T, JIANG J, LIU P, et al. Elastic sketch: Adaptive and fast network-wide measurements[C]// Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication. 2018: 561-575.

[3] TANG L, HUANG Q, LEE P P C. Mv-sketch: A fast and compact invertible sketch for heavy flow detection in network data streams[C]// IEEE Conference on Computer Communications (IEEE 2019). 2019: 2026-2034.

[4] FAN Z, ZHANG Y, YANG T, et al. Periodicsketch: Finding periodic items in data streams[C]// 2022 IEEE 38th International Conference on Data Engineering(ICDE). IEEE, 2022: 96-109.

[5] LIU Z, KONG C, YANG K, et al. Hypercalm sketch: One-pass mining periodic batches in data streams[C]// 2023 IEEE 39th International Conference on Data Engineering (ICDE). IEEE. 2023.

[6] FAN Z, GUO J, LI X, et al. Finding Simplex Items in Data Streams[C]// 2023 IEEE 39th International Conference on Data Engineering(ICDE). IEEE, 2023: 1953-1966.

[7] LI X, FAN Z, LI H, et al. SteadySketch: Finding Steady Flows in Data Streams[C]// 2023 IEEE/ACM 31st International Symposium on Quality of Service(IWQoS). IEEE, 2023: 1-9.

[8] ZHONG Z, YAN S, LI Z, et al. Burstsketch: Finding bursts in data streams[C]// Proceedings of the 2021 International Conference on Management of Data. 2021: 2375-2383.

[9] FAN Z, HU Z, WU Y, et al. Pisketch: finding persistent and infrequent flows[C]// Proceedings of the ACM SIGCOMM Workshop on Formal Foundations and Security of Programmable Network Infrastructures. 2022: 8-14.

[10] WANG J, ZHANG Y. Opportunity model for e-commerce recommendation: right product; right time[C]// Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2013: 303-312.

[11] JAIN V R, BAGREE R, KUMAR A, et al. wildCENSE: GPS based animal tracking system[C]// 2008 International Conference on Intelligent Sensors Networks and Information Processing. IEEE, 2008: 617-622.

[12] ROYH A, ZENG J, BAGGAG P, et al. C. Snoeren"Inside the social network's (datacenter) network[C]// Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication. 2015: 23-137.

[13] TAN L, LYANG F, YI J K. Optimisation of Sketch algorithm based on AVX instruction set[J]. Computer Science, 2021, 48(11A): 585-587.

[14] AGRAWALR, SRIKANT R. Fast algorithms for mining association rules[C]// Proceedings 20th International Conference. Very Large Data Bases VLDB, 1994, 1215: 487-499.

[15] YUAN Q, SHANG J, CAO X, et al. Detecting multiple periods and periodic patterns in event time sequences[C]// Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. 2017: 617-626.

[16] CHANG J H, PAR K N H. A novel weighting technique for mining sequence data streams[C]// IT Convergence and Security 2012. Springer Netherlands, 2013: 929-936..

[17] CHEN Y L, CHIANG M C, KOM T. Discovering time-interval sequential patterns in sequence databases[J]. Expert Systems with Applications, 2003, 25(3): 343-354.

[18] AGRAWAL R, IMIELIŃSKI T, SWAMI A. Mining association rules between sets of items in large databases[C]// Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data. 1993: 207-216.

[19] HAN J, PEI J, YIN Y. Mining frequent patterns without candidate generation[J]. ACM Sigmod Record, 2000, 29(2): 1-12.

[20] Anonymized Internet Traces 2016[OL]. https://catalog.caida.org/dataset/passive_2016_pcap.

[21] Taobao user behavior data set[OL]. ttps://tianchi.aliyun.com/dataset/649.

CHEN Xinyang, born in 2002, undergraduate. His main research interests include sketch and data mining.

ZHANG Dong, born in 1981, professor, Ph.D supervisor, is a member of CCF (No. 29654M). His main research interests include software defined networking, network virtualization and Internet QoS.

（责任编辑：喻藜）