

面向缺失数据的布鲁姆近似成员查询算法*

吴佳雯¹,王宇科²,裴书玉¹,谢 鲲¹,刘楚达³

(1.湖南大学 信息科学与工程学院,湖南 长沙 410082;

2.湖南大学 校园信息化建设与管理办公室,湖南 长沙 410082;

3.长沙航空职业技术学院,湖南 长沙 410082)

摘要:随着网络的发展,越来越多的场景需要在不完整数据下进行近似成员查询,传统成员查询的布鲁姆过滤器不能满足上述要求。提出面向缺失数据的布鲁姆近似查询算法,先对高维不完整数据的缺失部分进行预填充,通过PCA算法,将高维数据转换到低维数据,使用局部敏感哈希函数与标准哈希函数结合的方式将低维数据存储到布鲁姆过滤器中。使用两个真实数据集验证了所提算法的功能,所提面向缺失数据的布鲁姆近似查询算法,能有效地解决存在缺失数据的近似成员查询问题。

关键词:布鲁姆过滤器;近似成员查询;查询算法

中图分类号:TP393.0

文献标识码:A

DOI:10.16157/j.issn.0258-7998.212468

中文引用格式:吴佳雯,王宇科,裴书玉,等.面向缺失数据的布鲁姆近似成员查询算法[J].电子技术应用,2022,48(3):78-82,87.

英文引用格式:Wu Jiawen,Wang Yuke,Pei Shuyu,et al. Approximate membership query algorithm for incomplete data based on Bloom filter[J]. Application of Electronic Technique,2022,48(3):78-82,87.

Approximate membership query algorithm for incomplete data based on Bloom filter

Wu Jiawen¹,Wang Yuke²,Pei Shuyu¹,Xie Kun¹,Liu Chuda³

(1.College of Computer Science and Electronic Engineering,Hunan University,Changsha 410082,China;

2.Office of Information,Hunan University,Changsha 410082,China;

3.Changsha Aeronautical Vocational and Technical College,Changsha 410082,China)

Abstract: More and more scenarios require approximate membership queries for incomplete query data, but traditional Bloom filters for membership queries cannot meet these requirements. An approximate membership query algorithm for incomplete data based on Bloom filter is proposed. It first preprocesses the missing parts of the high-dimensional incomplete data, then converts the high-dimensional data to the low-dimensional data based on PCA technique, and the low-dimensional data is stored in a Bloom filter by combining local sensitive hash functions with standard hash functions. Extensive experiments are conducted using two publicly real-world network performance datasets, and it shows that the proposed algorithm efficiently solves the approximate membership query problem for data with incomplete data. It is also necessary to enrich the means of filling in the missing parts in the data pre-processing. The proposed solution can effectively solve the approximate membership query problem for data with missingness.

Key words: Bloom filter; approximate membership query; query algorithm

0 引言

标准的布鲁姆过滤器(Bloom Filter, BF)^[1]是一个空间效率很高的数据结构,它可以表示集合并支持集合的成员查询,快速判断查询元素是否在集合中。当给定一个查询元素 e 时,它被用来回答查询元素是否在这个集合。一个标准的布鲁姆构造一个长度为 m 的比特位数组,初始化为0。在插入阶段,它使用 k 个独立的哈希函数 $h_1(\cdot), \dots, h_k(\cdot)$ 来计算插入元素在数组中对应的 k 个

哈希位置 $h_1(e) \% m, \dots, h_k(e) \% m$,并将这 k 个哈希位置置位为“1”。在查询阶段,通过检查是否所有的 k 个哈希位置都置位为“1”,来判断元素是否在集合中。如果它们都置位为“1”,则认为查询元素 e 在集合 S 中;否则,则认为查询元素 e 不在集合 S 中。

现有标准布鲁姆过滤器通常用于常规的精确匹配的成员集合查询(Exact-matching Membership Query, EMQ),即:检查查询数据本身是否存储在布鲁姆过滤器,它是否是集合的一个成员。布鲁姆过滤器作为一种空间精简、查询高效的支持成员集合查询结构,一直被广泛用

* 基金项目:国家自然科学基金项目(61972144);湖南省科教联合基金项目(2019JJ70031)

于各种实际应用中^[2-3]。在网络领域应用中,布鲁姆过滤器可以用来存储防火墙海量的黑名单数据^[4],以及在网站中进行内容去重等^[5]。在大数据应用中,例如 HBase 中使用布鲁姆过滤器来减少代价高昂的 I/O 次数,提升数据库查询效率^[6]。

然而,在新兴的网络应用中,传统的布鲁姆过滤器所提供的精确匹配已不足以应对各种各样的网络应用需求。这些应用要求现有算法可以对数据进行近似成员查询(Approximate Membership Query, AMQ)^[7]。与精确匹配不同的是,如果 AMQ 返回肯定的结果,那么这里存在两种情况:第一是查询元素本身存在于 BF 中,第二是查询元素本身在某个范围内与 BF 中存储的一条或多条元素近似。AMQ 是给定距离度量下,考察查询数据与集合 S 中任意成员的接近程度。这个距离的度量可以是汉明距离(Hamming distance)^[8]、欧氏距离(Euclidean distance)^[9]等。AMQ 常常应用在各种领域,如信息中心网络(Information-Centric Networking, ICN)^[10-12]利用 AMQ 来进行 IP 路由选择,在图像检索领域使用 AMQ 进行相似图像检索等。

近些年来针对 AMQ 的布鲁姆过滤器研究得到了学术界越来越多的关注^[7,13-15]。Kirsch 等人最先将局部敏感哈希函数(Locality-Sensitive Hashing, LSH)^[15]和 BF 结合,提出距离敏感的布鲁姆近似查询算法(Distance-Sensitive Bloom Filters, DSBF)^[9]。在 DSBF 基础上,Qian Jiangbo 等人提出了一种汉明多粒度局部敏感布鲁姆过滤器(Hamming Metric Multi-Granularity Locality-Sensitive Bloom Filter, HLBF)^[7],HLBF 支持不同粒度汉明距离的近似成员查找。

然而这些支持近似成员查询的布鲁姆过滤器结构都有一个共同的局限性。存储和查询元素必须是完整的数据,无法处理请求查询含有缺失数据的情况。现实生活中的数据集合并不是完整无缺的,它们元素通常为高维空间的元素,高维数据往往还伴随着维度冗余的情况,且存在数据缺失。

为了解决这个问题,本文提出一种面向缺失数据的布鲁姆近似查询算法。它先对缺失数据进行预填充,再通过 PCA(Principal Component Analysis, PCA)算法^[16]将高维数据转换到低维数据,使用低维数据进行布鲁姆的近似查询。在近似查询中,采用局部敏感哈希函数与标准哈希函数结合的方式来插入、查询低维数据。本文使用局部敏感哈希函数中 AND 和 OR 操作相结合的方法进一步降低算法的假阳性率与假阴性率,提高了算法的性能。本文在真实数据集上对所提算法进行了验证,充分证明所提算法有效地解决了存在缺失数据的情况下近似成员的查询问题。

1 方案概述

本文设计一个新颖的布鲁姆结构,用来实现面向高

维缺失数据的近似成员查询。近似成员查询指的是,当查询元素与现有集合中的某些元素在某个范围内相似时,该查询元素就被视为集合的一个近似成员。本文给出近似成员查询的基本定义如下:

定义 1(近似成员查询)给定一个距离参数 R 、元素集合 S ,查询元素 q 被视为近似于集合 S ,当且仅当: $\exists p \in S, \|q, p\| \leq R$,其中, $\|*\|$ 表示 q 与 p 之间的距离,距离的度量方式可以是汉明距离、欧氏距离等。当 $R=0$ 时,该查询就是一个精确的成员查询,因此精确查询可视为是一个特殊的近似查询。

设计的面向缺失数据的布鲁姆近似查询算法的方案如图 1 所示,主要分为 3 个步骤:

(1)填充高维的缺失数据。缺失的原始数据无法直接在布鲁姆过滤器中存储/查询,需要对缺失的数据进行填充;

(2)基于 PCA 技术,将高维填充数据降维表示到低维空间,去除不相关属性,提取出数据的重要特征;

(3)在插入阶段,利用 LSH 哈希函数族为降维后数据生成 L 个独立的组,每一组看作是原数据的哈希值表示。并将每一组作为一个整体输入,再通过 b 个传统随机哈希函数进行哈希运算,对映射在 BF 上的哈希位置置位为“1”,完成低维数据在布鲁姆过滤器的存储。

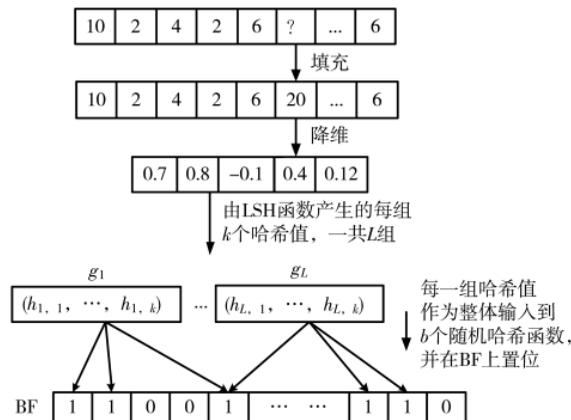


图 1 面向缺失数据的布鲁姆近似成员查询方案

2 详细设计

2.1 缺失数据预填充与 PCA 降维

由于原数据带有缺失,而使用 PCA 降维^[16]的数据需要维度对齐,因此需处理数据的缺失处。

具体地,首先计算出原始数据每个维度无缺失处的均值,记录下各个维度的均值,并使用维度均值补全缺失处的值,以上即为对数据的初步处理操作。

接着,将处理后的数据组织成一个 $n \times d$ 的矩阵 X , X 中每行代表一条数值向量,每列代表一个维度。并使用标准的 PCA 降维技术:矩阵去中心化 $X - \text{协方差矩阵计算} - \text{选取投影矩阵 } P - \text{降维计算 } \bar{X} \times P$,至此,完成了缺

失数据的预处理和降维操作。

2.2 插入算法

在 PCA 降维后,使用降维后的数据作为下一阶段的输入数据进行插入操作,具体地:

(1)利用 p-stable LSH^[9]得到 L 个哈希值的集合,哈希函数为 $h_{(a,e)} = \lfloor (a \cdot v + e) / w \rfloor$,其中 v 为输入数据, a 为与 v 维度相同的一个随机向量, a 中的每个分量均从 p-stable 分布中获取, w 为桶宽, e 是一个大于等于 0 且不大于 w 的随机数。本文在高斯分布中选取 a 的值,通过变换 a 和 e 的取值得到一个大小为 k 个的哈希值集合:

$$g_i = \{h_1, h_2, \dots, h_k\}, 1 \leq i \leq L \quad (1)$$

其中, $h_j (1 \leq j \leq k)$ 表示利用 p-stable LSH 得到的第 j 个哈希值,一个 g_i 组包含 k 个哈希值, $g_i (1 \leq i \leq L)$ 表示第 i 组哈希值集合,重复上述生成一个组的步骤,一共生成 L 个组。

(2)将每个 g_i 组作为输入,使用 b 个独立的随机哈希函数将其映射到布鲁姆过滤器的 b 个位置。具体地,一个标准的随机哈希函数将一个 g_i 组映射为一个值,然后将该值模上位数组大小得到一个数值,这个数值即为插入的位置。此步骤可表示为:

$$\text{index}_l = H_l(g_i) \bmod m, 1 \leq l \leq k \quad (2)$$

其中, $H_l(*)$ 表示第 l 个标准的随机哈希函数, m 表示位数组大小, index_l 表示使用第 l 个哈希函数定位到的位置,然后将位数组中第 index_l 位的值置为 1。重复上述第二阶段,使得 L 组 g_i 都被映射到布鲁姆过滤器位数组中,这样就完成了一次插入操作。综上所述,对于每一条输入数据,位数组中共有 $L \times b$ 个位置被置为“1”。

2.3 查询算法

布鲁姆过滤器的查询操作一般发生在插入操作完成之后。带有缺失的查询数据在使用近似查询布鲁姆过滤器进行近似查询之前,同样需要对数据进行预填充和 PCA 降维处理。使用降维后的数据作为输入数据在布鲁姆中查询,具体地:

(1)与插入算法的第一步相同,使用 p-stable LSH 函数得到 L 组哈希值集合 $\{g_1, \dots, g_L\}$,其中每一组 g_i 包含 k 个哈希值,即: $g_i = \{h_1, h_2, \dots, h_k\}$ 。相似查询中,相近的值可能会落于相邻的桶中,因此与插入操作不同是,除了每一组 g_i ,还需要扩展每一组哈希值集合 g_i 为 $2k$ 组哈希值集合,作为下一步随机哈希函数的输入。

(2)将每一组 g_i 和由它扩展的 $2k$ 组哈希值集合 $g_j^i (1 \leq j \leq 2k)$,一共作 $2k+1$ 组哈希函数集合,作为随机哈希函数输入。计算每一组哈希函数映射到布鲁姆的 b 个位置,判断 b 个位置是否全部被置为“1”。

如图 2 所示,将每一组 g_i 扩展成 $2k$ 组哈希值集合 g_j^i 的方式是:对于组中 $\{h_1, h_2, \dots, h_k\}$ 每一个哈希值,将其值减 1 而其他值保持不变得到一组哈希值集合,然后将

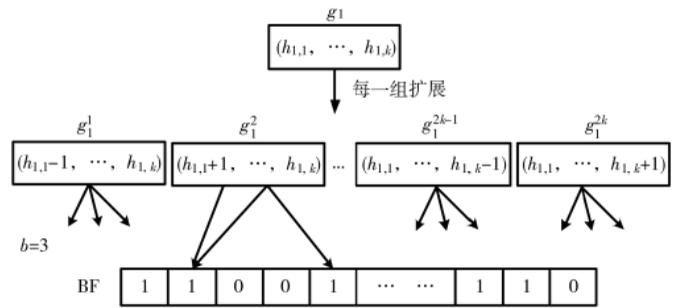


图 2 近似成员查询操作

其值加 1 而其他值保持不变得到另一组哈希值集合,这样原来的一组哈希值集合 g_i 就扩展成了 $2k$ 组哈希值集合 $g_j^i, 1 \leq j \leq 2k$,即:

$$\begin{cases} g_1^1 = \{h_1+1, h_2, \dots, h_k\} \\ g_1^2 = \{h_1-1, h_2, \dots, h_k\} \\ \vdots \\ g_k^{2k-1} = \{h_1, h_2, \dots, h_k+1\} \\ g_k^{2k} = \{h_1, h_2, \dots, h_k-1\} \end{cases} \quad (3)$$

对于每组哈希值集合 g_i 和它对扩展的 $2k$ 组哈希值集合,若任意一组在 BF 的 b 个索引位置都被置为 1,则查询元素存在。否则,查询数据与给定集合并不相似。

2.4 假阳性分析

假阳性率出现在原数据不与插入到布鲁姆过滤器中的集合近似,但是查询结果返回 true 的时候,也就是说算法误认为在插入集合中存在与查询数据相近的数据。因此,假阳性误判率由两个部分组成:局部敏感哈希产生的假阳性误判率、标准哈希函数产生的假阳性误判率,因此假阳性率可表示为:

$$P_{fp} = P_{LSH} + (1 - P_{LSH}) FPR_{SBF} \quad (4)$$

其中, P_{LSH} 表示由局部敏感哈希产生的假阳性误判率, FPR_{SBF} 表示由带有 b 个哈希函数的标准布鲁姆过滤器产生的误判率。

$$P_{LSH} = 1 - \prod_{i=1}^n (1 - x_i^k - k \cdot x_i^{k-1} (1 - x_i) \sum_{\partial=-1, +1} e^{-C \cdot x_i(\partial)^2})^L \quad (5)$$

其中, L 表示组数, k 表示每组包含的哈希值个数, n 表示插入到布鲁姆过滤器中数据的个数, C 是常量, $x(\partial)$ 可表示为:

$$x(\partial) = \begin{cases} (a \cdot x + e) - \lfloor \frac{a \cdot x + e}{w} \rfloor \times w & \partial = -1 \\ w - (a \cdot x + e) + \lfloor \frac{a \cdot x + e}{w} \rfloor \times w & \partial = +1 \end{cases} \quad (6)$$

其中, a 是和原数据维度相同的随机向量, w 表示桶宽, e 为 $[0, w]$ 范围内的随机数。

又由[7]可知,在标准的 BF,其假阳性误判率为:

$$FPR_{SBF} = 1 - (1 - (1 - (1/m)^{nb})^b)^{(2k+1)L} \quad (7)$$

其中, m 表示位数组大小, b 表示使用的标准哈希函数

个数, k 表示每组包含的哈希值个数。

2.5 假阴性分析

算法的假阴性发生在局部敏感哈希函数产生假阴性并且在查询布鲁姆过滤器位数组时 b 个位置不都为 1。因此根据假阳性率的分析可得, 算法的假阴性率为:

$$P_{in} = 1 - P_{LSH} - (1 - P_{LSH})FPR_{SBF} \quad (8)$$

其中, $P_{LSH} = 1 - \prod_{i=1}^n (1 - x_i^k - k \cdot x_i^{k-1} (1 - x_i) \sum_{\theta=-1, +1} e^{-C \cdot x_i(\theta)^2})^L$,

$FPR_{SBF} = 1 - (1 - (1 - (1 - (1/m))^{nb})^{(2k+1)^L})$, L 表示组数, k 表示每组包含的哈希值个数, n 表示插入到布鲁姆过滤器中数据的个数, m 表示位数组大小, b 表示使用的标准哈希函数个数, x 为查询数据。

3 实验评估

3.1 数据集介绍

本文实验所使用的数据均来自真实数据集。

(1) CoverType 数据集

CoverType 数据集^[17]属于一个分类数据集, 该数据集有 581 012 条不含缺失的数据, 每条数据包含 54 个维度, 数据集线性不可分。具体来说, 这 54 个属性包括 10 个定量变量、4 个二元荒野地区和 40 个二元土壤类型变量。

(2) HandWrite 数据集

手写字识别数据集^[17]是一个分类数据集, 共有 5 620 个不重复且无缺失数据, 每条数据都有 64+1 个维度, 其中前 64 个维度都是 0~16 之间的整数, 最后一个维度是数据的标签。

3.2 评价指标

假阳性误判率(False Positive Rate, FPR), 如果一个查询数据不近似于插入到布鲁姆过滤器中的数据集合, 但是查询返回 true, 这就是一个假阳性错误。

假阴性误判率(False Negative Rate, FNR), 如果一个查询数据近似于插入到布鲁姆过滤器中的数据集合, 但是查询返回 false, 这就是一个假阴性错误。

3.3 实验结果及分析

为了检验近似查询布鲁姆过滤器的参数与查询速率以及误判率之间的关系, 本节进行了参数实验, 并固定布鲁姆过滤器位数组大小为 $m=2^{24}$ bit, 使用的标准随机哈希函数的个数为 $b=3$ 。插入到布鲁姆过滤器中的数据量为 7 413 条, 查询数据为 3 177 条, 总缺失率为 0.005。

图 3 是在数据集 CoverType 下, 固定条件(数据降到 10 维、 $w=12$ 、 k 变化时 $L=3$ 、 L 变化时 $k=7$) 下, L 与 k 的值不断增加时, 平均每条数据近似查询所需的时间曲线图。由图可知, 当二者取值不断增加时, 平均查询时间也随之增加, 其中查询时间随 k 呈指数增加, 随 L 呈线性增加。这是由于 k 增加意味着每个 g_i 组包含的哈希值增多, 哈希计算所需时间增多, 并且一个 g_i 组除了自身外, 还需检查 $2k$ 个扩展组, 因此平均查询时间随 k 呈指数

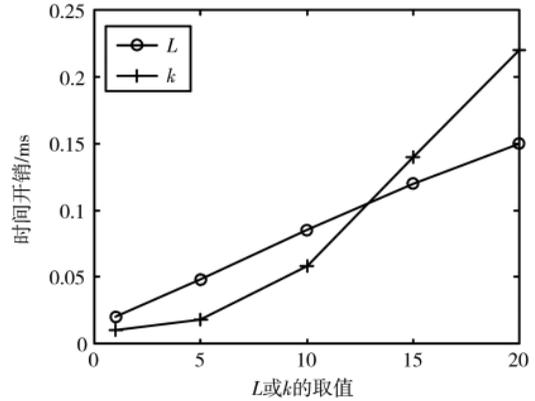


图 3 平均查询时间

增加, 随 L 呈线性增加。

图 4 为在 CoverType 和 HandWrite 数据集下, 假阳性误判率、假阴性误判率的随 L 变化曲线图。由图可看出在两个数据集中, 当 L 增加时, 假阳性率均随之增加。

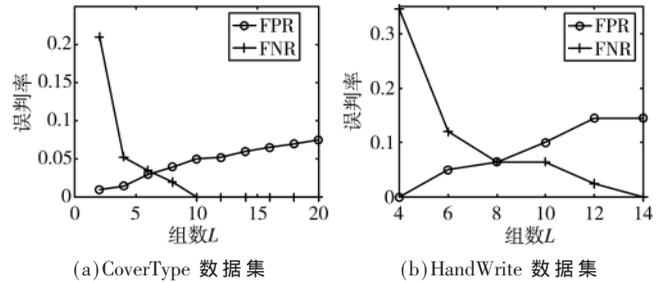


图 4 两个数据集下不同 L 取值的误判率

图 5 为在 CoverType 和 HandWrite 数据集下, 假阳性误判率、假阴性误判率随每组哈希函数个数 k 变化的曲线图。当 k 增加时, 每个组的哈希值越多, 在进行映射时, 要将组内哈希值作为一个整体插入到位数组中, 因此哈希值越多, 组与组之间差异越大, 碰撞的概率越小, 假阳性率随之降低, 假阴性率随之上升。

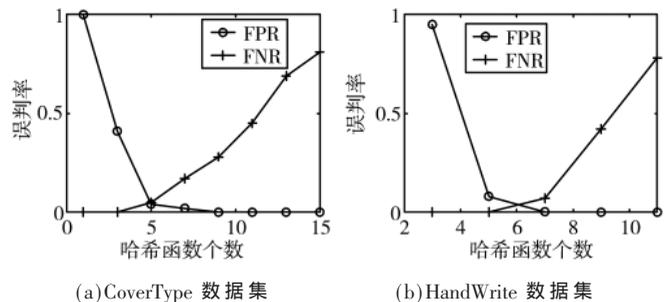


图 5 两个数据集下不同 k 取值的误判率

图 6 为在两个数据集中, 算法假阳性误判率以及假阴性误判率随 p-stable LSH 哈希函数中桶宽 w 的变化曲线图, 从图中可看出, 算法假阳性率随着 w 的增加而增大, 假阴性率随着 w 的增加而减小。这是由于 w 越大

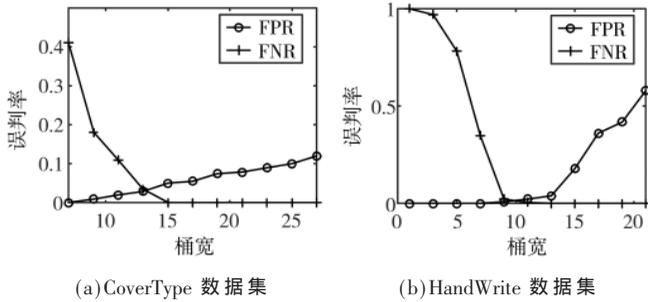


图 6 不同 w 下的误判率

意味着桶宽越大,因此近似范围越大,落入同一桶的数据越多,哈希冲突的概率越大,假阳性率越大,同时相近的值越可能落入同一个桶,因此假阴性率越小。

图 7 为在数据集 CoverType 中,本节方案在不同的近似距离下误判率变化曲线图,实验时本方案将数据维度从 54 维降到 8 维,使用 $L=4$ 个组,每组使用 $k=6$ 个局部敏感哈希函数,同时设置桶宽 $w=12$ 。图中显示随着近似性度量的欧氏距离增加,假阴性误判率渐渐增加,这是因为当距离增加,认为距离较远的点也近似于插入集合,但是原有的判定条件以及参数均不变,所以对于假阴性误判率,其分母真阳性数增加,但是原有的查询数目不变,而假阴性数目=真阳性数-查询出 true 的数目,所以当真阳性数 tp 增加时,假阴性误判率增加。同理,对于假阳性误判率,当真阴性数减小时,假阳性误判率减少。

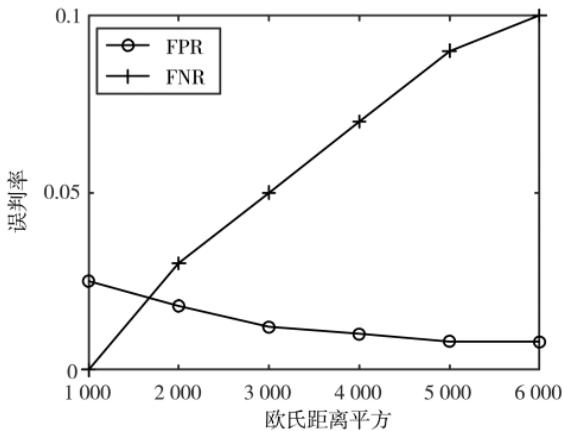


图 7 不同欧式距离下的误判率

表 1 中,在两个数据集下本方案将查询数据降维到不同维度然后进行近似查询,线性查找(Linear Search)在查询数据具有不同维度时进行暴力查询。由表可看出,本方案所需的平均查找时间比线性查询明显少得多。所谓线性查找就是遍历插入集合,计算查询数据与每一个插入数据的距离并判断是否小于指定值,若是则认为近似,因此线性查找所需时间与插入集合的数目有关。但是本文方案基于布鲁姆过滤器,所需查找时间仅仅与

表 1 不同维度下算法的平均查询时间

维度	本文所提算法		线性搜索
	CoverType 数据集	HandWrite 数据集	
50	0.056	0.087	1.27
40	0.05	0.077	1.03
30	0.046	0.072	0.8
20	0.043	0.068	0.53
10	0.041	0.053	0.28

L 、 k 、 b 的值有关,通常这些参数较小且固定,算法是常数级查询时间,所以所需时间代价较小。从表中亦可看出,随着数据维度的降低,两者所需的平均查询时间均明显下降,基于降维的近似查找布鲁姆过滤器算法又通过降低查询数据维度进一步加快了查询速度。

4 结论

本文提出了一种面向缺失数据的布鲁姆近似成员查询算法,可以判断查询的缺失数据是否与存储在布鲁姆中的数据相似。本文使用两个真实数据集对所提算法的有效性进行验证,在假阳性率、假阴性率和查询时间上,优于当前基于线性搜索的近似查询算法。

参考文献

- [1] BLOOM B H.Space/time trade-offs in hash coding with allowable errors[J].Communications of the ACM,1970,13(7): 422-426.
- [2] LUO L, GUO D, MA R T B, et al.Optimizing bloom filter: challenges, solutions, and comparisons[J].IEEE Communications Surveys & Tutorials, 2018, 21(2): 1912-1949.
- [3] TARKOMA S, ROTHENBERG C E, LAGERSPETZ E.Theory and practice of Bloom filters for distributed systems[J].IEEE Communications Surveys & Tutorials, 2011, 14(1): 131-155.
- [4] GERA VAND S, AHMADI M.Bloom filter applications in network security: a state-of-the-art survey[J].Computer Networks, 2013, 57(18): 4047-4064.
- [5] Cen Zhiwang, Xu Jungang, Sun Jian.A multi-layer bloom filter for duplicated URL detection[C]//Proc.of 2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE).Chengdu, China, 2010.
- [6] BHUSHAN M, BANERJEA S, YADAV S K.Bloom filter based optimization on HBase with MapReduce[C]//Proc.of 2014 International Conference on Data Mining and Intelligent Computing(ICDMIC).Delhi, India, 2014: 1-5.
- [7] Qian Jiangbo, Huang Zhipeng, Zhu Qiang, et al.Hamming metric multi-granularity locality-sensitive bloom filter[J].IEEE/ACM Transactions on Networking, 2018, 26(4): 1660-1673.
- [8] CRAINICEANU A, LEMIRE D.Bloofi: multidimensional bloom filters[J].Information Systems, 2015, 54: 311-324.

(下转第 87 页)

3 结论

本次研究通过采用 Hi3559+FPGA 架构,设计完成了多路 HD-SDI 高清视频传输系统的开发和验证,该系统能够在 Hi3559 上实现 8 路 HD-SDI 视频图像的采集、传输、处理以及全景显示。经过开发和测试验证,该系统能够满足 HD-SDI 视频传输的协议标准和速率要求,能够支持长距离的传输,工作稳定可靠。对于想要利用 Hi3559 实现多路高清视频的采集、远距离传输以及智能算法等方向具有较高的推广和参考应用价值。

参考文献

- [1] 唐伟.面向嵌入式图像处理系统的 SOPC 芯片通信接口设计与实现[D].南京:东南大学,2018:51-52.
- [2] 刘旭阳.2013 年电影电视工程学会(SMPTE)电影技术标准发展综述[J].现代电影技术,2014(2):40-43.
- [3] 姜忠兵.基于 SMPTE274M 标准的高清视频发生器设计与实现[J].电子设计工程,2012,20(17):150-152.
- [4] 钱宏文,王毅,刘会.一种基于 FPGA 的多路 HD-SDI 测试系统[J].电子技术应用,2020,46(8):121-125.
- [5] 张泽孜,谭大伦,王冬海.Hi3559A 在视频结构化平台中的应用[J].无线互联科技,2019(12):143-144.
- [6] 薛一哲,吴云,王昭.基于 DSP 串行通信接口开发的研究[J].航空计算技术,2018,18(1):95-97.
- [7] 白金成.基于 FPGA 的多路 Cameralink 数字图像光纤传输系统[J].液晶与显示,2019,34(8):787-792.
- [8] 彭勃,何宾.FPGA 在视频拼接中的应用与实现[J].计算机工程与设计,2013,34(5):1635-1639.
- [9] 付强,杨文豪,倪文龙.基于 FPGA 的 HD-SDI 转 HDMI 接

(上接第 82 页)

- [9] KIRSCH A, MITZENMACHER M. Distance-sensitive bloom filters[C]//2006 Proceedings of the Eighth Workshop on Algorithm Engineering and Experiments(ALENEX). Society for Industrial and Applied Mathematics, 2006: 41-50.
- [10] XYLOMENOS G, VERVERIDIS C N, SIRIS V A, et al. A survey of information-centric networking research[J]. IEEE Communications Surveys & Tutorials, 2013, 16(2): 1024-1049.
- [11] AMADEO M, CAMPOLO C, QUEVEDO J, et al. Information-centric networking for the Internet of Things: challenges and opportunities[J]. IEEE Network, 2016, 30(2): 92-100.
- [12] FANG C, YAO H, WANG Z, et al. A survey of mobile information-centric networking: research issues and challenges[J]. IEEE Communications Surveys & Tutorials, 2018, 20(3): 2353-2371.
- [13] HUA Y, XIAO B, VEERAVALLI B, et al. Locality-sensitive Bloom filter for approximate membership query[J]. IEEE Transactions on Computers, 2011, 61(6): 817-830.
- [14] HUA Y, LIU X. Locality-sensitive Bloom filter for approximate

口设计与实现[J].电子与封装,2018,18(2):37-39.

- [10] 杨晓萍,胡玉,张凯.基于 FPGA 的视频拼接技术研究[J].吉林大学学报,2016,34(6):709-715.
- [11] 吴俊霖.基于 FPGA 的 HD-SDI 信号传输与处理[D].成都:西南交通大学,2011:11-14.
- [12] 张永昊,宋华军,吴玉兴.基于 FPGA+DSP 的 SDI 高清图像处理系统[J].国外电子测量技术,2019,38(09):28-32.
- [13] 孙德晟.FPGA 的 SD/HD/3G-SDI 的图像环路测试装置设计[J].单片机与嵌入式系统应用,2019(11):69-73.
- [14] 徐大鹏,孙海江.HD-SDI 视频嵌入式图像采集设计[J].计算机测量与控制,2015,23(9):3214.
- [15] 邹亦婷.基于多处理器协同的智能工业相机图像处理系统的设计与实现[D].成都:电子科技大学,2019:8-9.
- [16] 李艳婷,王水鱼.视频图像中动态目标检测系统的 FPGA 硬件设计[J].计算机系统应用,2017,26(3):119-125.
- [17] 马森.嵌入式图像处理平台设计及实现[D].成都:电子科技大学,2018:12-13.

(收稿日期:2021-03-01)

作者简介:

王毅(1991-),男,硕士,工程师,主要研究方向:视频图像处理和 FPGA 应用。

张禹(1993-),男,本科,工程师,主要研究方向:DSP 和 FPGA 应用等方面的数字硬件开发。

钱宏文(1975-),男,本科,研究员级高级工程师,主要研究方向:集成电路应用和微系统领域设计。



扫码下载电子文档

membership query[M]//Searchable Storage in Cloud Computing. Springer, Singapore, 2019: 99-127.

- [15] DATAR M, IMMORLICA N, INDYK P, et al. Locality-sensitive hashing scheme based on p-stable distributions[C]// Proceedings of the Twentieth Annual Symposium on Computational Geometry, 2004: 253-262.
- [16] WOLD S, ESBENSEN K, GELADI P. Principal component analysis[J]. Chemometrics and Intelligent Laboratory Systems, 1987, 2(1-3): 37-52.
- [17] BLAKE C, KEOGH E, MERZ C. UCI repository of machine learning databases[Z]. 1999.

(收稿日期:2021-12-13)

作者简介:

吴佳雯(1996-),女,硕士研究生,主要研究方向:布鲁姆查询、近似查询。

王宇科(1978-),通信作者,男,硕士研究生,高级工程师,主要研究方向:网络安全、网络测量与分析、数据挖掘, E-mail: wyk@hnu.edu.cn。

裴书玉(1993-),女,硕士研究生,主要研究方向:布鲁姆查询、概率型数据结构。



扫码下载电子文档

版权声明

经作者授权，本论文版权和信息网络传播权归属于《电子技术应用》杂志，凡未经本刊书面同意任何机构、组织和个人不得擅自复印、汇编、翻译和进行信息网络传播。未经本刊书面同意，禁止一切互联网论文资源平台非法上传、收录本论文。

截至目前，本论文已经授权被中国期刊全文数据库（CNKI）、万方数据知识服务平台、中文科技期刊数据库（维普网）、DOAJ、美国《乌利希期刊指南》、JST 日本科技技术振兴机构数据库等数据库全文收录。

对于违反上述禁止行为并违法使用本论文的机构、组织和个人，本刊将采取一切必要法律行动来维护正当权益。

特此声明！

《电子技术应用》编辑部

中国电子信息产业集团有限公司第六研究所