链滴

BloomFilter的错误率保证

作者：corner87

原文链接：https://ld246.com/article/1439912385227

来源网站：链滴

自从知道BloomFilter这个索引方法，我就迫不及待的开始使用了。但是一直都只是明白个原理用起来也觉得不亦乐乎。
这周末把《数学之美》这本书看了一遍，很多收获。很多方法自己都曾经接触过、了解过。经过者一深入浅出的介绍，让自己对这些方法又多了一些理解。就像这个BLoomFilter一样，这次看了以，尤其是作者顺手还给出了错误率的估算，我还看懂了，就感到很惊喜。以前看论文的时候，我都是接忽略这些理论的证明的，只看原理。
我感觉我现在也能随手写出BloomFilter的错误率估算了。来一个。
当给定条件：
m：过滤器的bit长度
k：哈希函数的个数
n：插入过滤器的元素个数
我们知道一个元素经过k个哈希函数后，会导致bit串中k个位置被置为1。于是，对比特串中的第i位置来说，其被一个哈希函数置为1的概率为：1/m，则其依然为0的概率为1-1/m。对一个元素来讲如果经过k个哈希函数都没有把第i个位置的值置为1的概率就应该是(1-1/m)^k，对第二个元素还是没能把这个位置的值置为1的概率就为(1-1/m)^2k，依次类推，n个元素过后，i位置仍旧为0的概率是(11/m)^nk。反过来，在经过n次增加修改比特串后，某个比特位为1的概率是1-(1-1/m)^nk。也就是，这是建立好查询索引后，每个为1的比特位都不是100%的可信的，因为哈希会有冲突的，于是，对个查询元素，经过k个哈希函数后查得k个比特位都为1的概率是(1-(1-1/m)^nk)^k。这个值是很小的特别是在当我们可以预估要索引的元素大概有多少数量时，我们可以留出充分多的bit位（1024个比位，也就才1k的大小），这个充分对内存来说是很easy的，和多设几个哈希函数，就能把错误率限制一个很小的范围内。
这个在海量数据查询时，查询效率是O(K)的，一个常数级别的、一般都是个位数的复杂度。内存销也小。特别适合海量数据查询和容许一点点错误率的使用环境。我在做词典检索时就用的这个，还人用于爬虫中url的重复性检测，等等。不好的是，不支持删除操作（但可以重新建一个BloomFilter记录删除的元素）。

