请注意其他教师和这些幻灯片的使用者：如果你发现我们的这些材料对你自己的讲座有帮助，我们将非常高兴。请自由地逐字逐句地使用这些幻灯片，或根据自己的需要对其进行修改。如果你在自己的讲座中使用了这些幻灯片的大部分内容，请附上这条信息或我们网站的链接：http://www.mmds.org

Mining Data Streams (Part 2)

海量数据集的挖掘

Cou[redacted]ng D[redacted]E 许可证

Counting Distinct Elements

- **问题：**
  - 数据流由一个从大小为*N*的集合中选择的元素宇宙组成。
  - 保持到目前为止看到的不同元素的数量的计数

- **明显的方法：**
  保持到目前为止所看到的元素的集合
  - 就是说，保留一个迄今为止看到的所有

不同元素的哈希表

Applications

- **在一个网站被抓取的网页中发现了多少不同的词?**
  - 不寻常的低或高数字可能表明是人工网页（垃圾邮件？）
- **每个客户在一周内要求多少个不同的网页?**

- **上周我们售出了多少种不同的产品?**

Using Small Storage

- 真正的问题是：如果我们没有空间来维持迄今所看到的元素集，怎么办？
- 以无偏见的方式估计计数
- 接受计数可能有一点误差，但限制误差大的概率

Flajolet-Martin Approach

- 挑选一个哈希函数*h*，将每个 *N* 个元素至少要**对数$_2$ *N***位
- 对于每个流元素*a*，让*r(a)*为*h(a)*中尾部 **0**的数量。
  - **r(a)** = 从右数第一个1的位置
    - 例如，说*h(a)=12*，那么*12*在二进制中是*1100*，所以 *r(a)=2*
- 记录***R*=看到的最大*r(a)*。**
  - **R = max$_a$ r(a)**, 在迄今为止看到的所有项目*a* *中*。

- **估计不同元素的数量 = 2^R**

Why It Works: Intuition

- **非常非常粗略和启发式的直觉，为什么 Flajolet-Martin会成功：**
  - ***h(a)*以相等的概率将*a*洗成*N*个值中的任何一个。**
  - 那么***h(a)***是一个**$\log_2 N$**比特的序列、其中2^{-r}分数的所有作为都有***r***个零的尾巴
    - 约有50%的为*****0**
    - 约有25%的人对****00**有兴趣
    - 因此，如果我们看到***r*=2**的最长尾巴（即项目哈希结束***100**），那么我们可能已经看到了到目前为止，**大约有*4* 个**不同的项目
  - **因此，在我们看到一个长度为*r*的零后缀的**

项目之前，需要哈希大约^{2r}个项目。

Why It Works: More formally

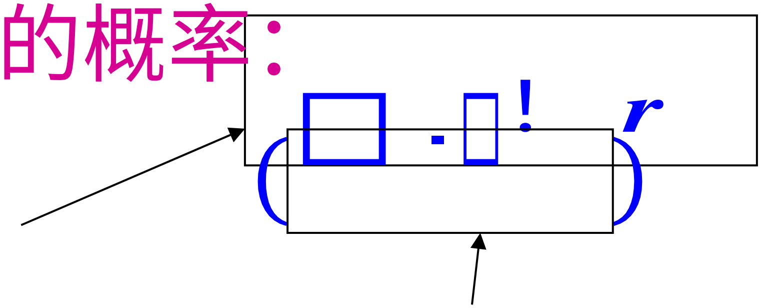
- 现在我们展示一下为什么弗拉乔莱-马丁会成功
- 从形式上看，我们将表明，找到r个零的尾巴的概率：
  - 如果$m \gg 2$，则转为1。r
  - 如果$m \ll$Nd_1D7D0↩，则转为0。r

其中?是到目前为止在流中看到的不同元素的

数量。

- **因此，^{2R}几乎总是在*m!***

- **一个给定的*h*(*a*)至少以*r*个零结束的概率是多少，是2^{-r}**
  - h(a)均匀地随机洗练元素
  - 一个随机数最后至少有*r*个零的概率是2^{-r}
- 那么，在*m*个元素中**没有**看到长度为*r*的尾巴的概率：



Why It Works: More formally

- **请注意：** $(1-2^{-r})^m = (1-2^{-r})^{2^r(m2^{-r})} \approx e^{-m2^{-r}}$
- **没有找到长度为*r*的尾巴的概率是：**
  - 如果***m***<<2^r，则概率趋向于**1**。
    - $(1-2^{-r})^m \approx e^{-m2^{-r}} = 1$ 当**m/2r→0**时
    - 所以，找到长度为***r*的**尾巴的概率趋向于**0**
  - 如果***m***>>2^r，则概率趋向于**0**。
    - $(1-2^{-r})^m \approx e^{-m2^{-r}} = 0$ 当**m/2^r→∞**时

Why It Works: More formally

- **因此，2^k几乎总是在*m*!**

Why It Doesn't Work

- E[2^R]实际上是无限的
  - 当$R \rightarrow R+1$时，概率减半，但价值翻倍
- 解决方法包括使用许多哈希函数h_i，并得到许多R的样本$_i$
- 样品R_i，如何组合?
  - 平均值? 如果有一个非常大的数值▯R▯怎么办?
  - 中位数? 所有估计值都是2的幂
  - 解决方案:
    - 将你的样品分成几个小组
    - 取各组的中位数
    - 然后取中位数的平均数

（2）Computing M机会

Generalization: Moments

- 假设一个流有从N个值的集合A中选择的元素
- 让m_i是值i在流中出现的次数
- 第k个*时刻*是

$$\sum_{i \in A} (m_i)^k$$

Special Cases

$$\sum_{i \in A} (m_i)^k$$

- **0thmoment =** 不同元素的数量
  - 刚刚考虑的问题
- **第1个时刻=**计数的数量元素=流的长度
  - 易于计算
- **第2个时刻=** ***惊喜数S*** **=** 衡量分配不均的程度

Example: Surprise Number

- **长度为100的流**
- **11个不同的值**
- 项目计数：**10, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9**
  惊喜S = 910
- 项目数：**90, 1, 1, 1, 1, 1, 1, 1 ,1, 1, 1**
  惊喜S = 8,110

[Alon, Matias, and Szegedy]

AMS Method

- AMS方法适用于所有时刻
- 提供一个无偏见的估计
- 我们将只专注于第二时刻S
- 我们挑选并跟踪许多变量X：
  - 对于每个变量X，我们存储*X.el*和*X.val*
    - *X.el*对应的是项目*i*
    - *X.val*对应于项目*i*的**计数**。
  - 注意这需要在主内存中进行计数，所以X的数量是有限的。
- 我们的目标是计算$S = \sum_i m_i^2$ ⍰

One Random Variable (X)

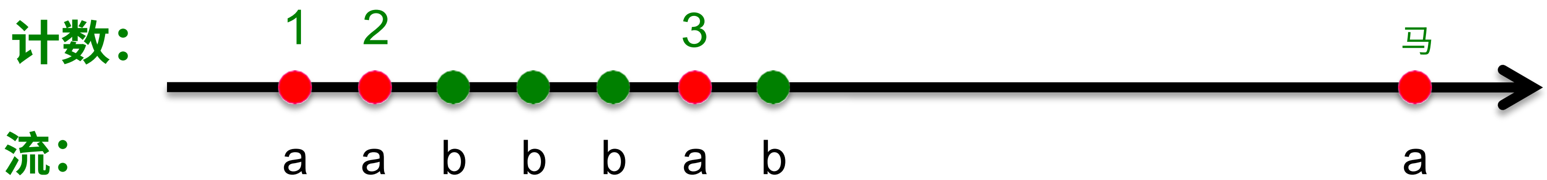
- **如何设置*X.val*和*X.el*?**
  - 假设流的长度为***n***（我们稍后放宽这一点）。
  - 挑选一些随机的时间***t***（***t*<*n***）来开始，这样任何时间都有同样的可能性
  - ***我们设定X.el=i***，在时间***t***，流有项目***i***。
  - 然后，我们保持计数***c***（***X.val* = *c***），*即*从选定的时间***t***开始，流中的数量。
- **那么第二时刻的估计（$\sum_i m^2$）是：**

$$S = f(X) = n\ (2 \cdot c - 1)$$

  - 注意，我们将跟踪多个X，(X_1 , X_2 , … X_k)，我们

的最终估计将是 $S = 1/k \sum^{\square} f(X_{\square})$

Expectation Analysis



- 第2个时刻是$S = \sum_i m_i^2$
- c_t ...时间t的项目出现的次数

  从时间t开始($c_{1a} = m_a$, $c_{2a} = m_a - 1$, $c_{3b} = m_b$)
- $E[f(X)] = \frac{1}{n}\sum_{t=1}^{n} n(2c_t - 1)$

$= \frac{1}{n}\sum_i n\,(1 + 3 + 5 + \cdots + 2m_i - 1)$

m_i ... 流中i项的总计数（我们假设流的长度为n）。

按看到的数 值对时间进行 分组

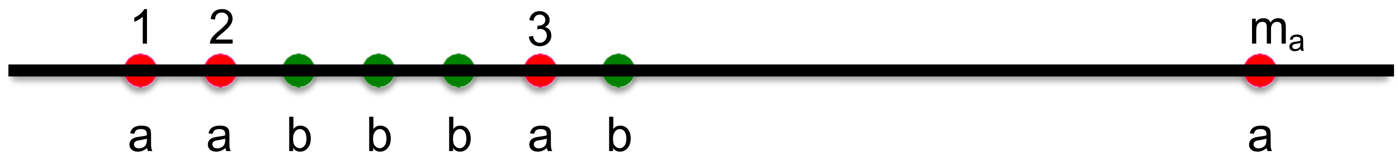
看到最后一个*i*的时间t (**c_t =1)**

看到倒数第二个*i*的时间 ***t*** (c_t =2)

看到第一个*i*的时间***t*** (***c*** **= m**

) ***ti***

计：



流：

- $E[f(X)] = \frac{1}{n}\sum_i n\,(1 + 3 + 5 + \cdots + 2m_i - 1)$
  - 小小的侧面计算：$(1 + 3 + 5 + \cdots + 2m_i - 1) = \sum_{i=1}^{m_i} (2i - 1) = 2\frac{m_i(m_i+1)}{2} - m_i = (m_i)^2$
- 那么 $E[f(X)] =$

  $=$ $\sum_i$
- 所以，$E[f(X)] = (\square_{\square})$

Expectation Analysis

n 2

- 我们有第二个时刻（在期待中）！

Higher-Order Moments

- **对于估计第k个时刻，我们基本上使用相同的算法，但改变估计：**
  - 对于k=2，我们使用 $n\ (2\cdot c-1)$
  - 对于k=3，我们使用： $n\ (3\cdot c^2 - 3c + 1)$ (其中c=X.val)
- **为什么?**
  - **对于k=2：** 记得我们有 （$1 + 3 + 5 + \cdots + 2m_{\square} - 1$）
    我们表明条款**2c-1**（对于c=1,...,m）相加为m^2
    - $\sum_{?=1}^{m} 2c - 1 = \sum_{?=1}^{m} c^2 - \sum_{?=1}(c-1)^2 = ?^2$
    - **所以：** $2c - 1 = c^2 - (c-1)^2$
  - **对于k=3：** $c^3 - (c-1)^3 = 3c^2 - 3c + 1$ $(\square - 1)$
- **一般来说**

- 估计值=□(□^{k} -

□)

Combining Samples

- **在实践中：**
  - 计算$f(X) = n(2c - 1)$为尽可能多的变量**X**，你可以在内存中装下。
  - 将它们分组平均化
  - 取平均数的中位数
- **问题：溪流永远不会结束**
  - 我们假设有一个数字***n***，即流中的位置数
  - 但真正的水流是永远流下去的，所以***n***是一个变量 - 到目前为止看到的输入数量

Streams Never End: Fixups

- (1) 变量X有*n*作为因子--单独保留*n*；只是在X中保留计数
- (2) 假设我们只能存储*k*个计数。随着时间的推移，我们必须抛出一些x：
  - **目标：** 每个起始时间*t*的选择概率为*k/n*
  - **解决方案：（固定大小的采样！）。**
    - 选择*k*个变量的前*k*次
    - 当第*n*个元素到达时（*n>k*），以*k/n*的概率选择它
    - 如果你选择了它，就把之前存储的一个变量X扔

出去，概率相同