**Count-Min Sketch**

在分析流数据时，有一类问题是计算流中各个item出现的次数。当不同item的数量比较小时，可以直接在内存中存储各个item对应的次数（可以用HashMap）。但是当不同item数量很大时，这样的存储将耗费巨大的内存空间。Count-Min是采用概率近似的方法计算各个Item对应的次数。具体如下图所示：

h1

hd

w

i

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | +c |  |  |  |  |  |  |
|  |  |  |  | +c |  |  |  |  |  |
|  |  |  |  | +c |  |  |  |  |  |
|  |  |  |  |  |  |  | +c |  |  |

设N为各个item对应的计数总和。对各item进行hash计算，其hash值可以平均分布于w个桶中。那么某个桶可以平均对应的冲突计数为N/w（hash值冲突导致）。

选择d个不同的hash函数（均要求其计算的hash值可以平均分布于w个桶中）。每个item分别用这d个hash计算其桶位，然后在对应的桶位上增加统计量（+c）。

概率上讲（设+c=+1），对于某个item计算得出的d个桶位，桶位冲突计数要大于2N/w的概率为1/2（**Markov inequality**），桶位冲突计数要小于2N/w(概率为1/2)。某个item对应的hash桶位冲突越多，其对应数值越大，冲突越少，其对应数值越小。**item对应hash桶位中最小的值最接近item对应的统计量（大于等于真实值）**。

而item对应的所有桶位冲突计数均大于2N/w的概率是1/2d。所以存在冲突计数小于2N/w的概率是p = 1-1/2d。当d=4时，p=15/16；d=8时，p=255/256。

如果要求最大错误计数为总计数的1/1000，即2/w < 1/1000，那么要求w = 2000；1/2d = 0.001，那么d = log21000 ≈ 10。如果采用4字节计数，那么存储空间是w×d×4=2000×10×4≈80KB

<http://web.stanford.edu/class/cs369g/lectures.html>