CSEP 521: Applied Algorithms Lecture 12 - Stream Algorithms: Frequency Estimates

Richard Anderson

February 11, 2021

Announcements

•

Frequency Moments

- Compute the sum of powers of frequency of elements
- Higher moments put more emphasis on most frequent items

F_0, F_1

- $F_0 = \Sigma f_i^0$
  - We will define $0^0 = 0$ here
  - $f_i^0 = 1$ if $f_i > 0$
  - Hence, F_0 is just the number of items. See lecture 11
- $F_1 = \Sigma f_i^1 = N$

$F_2 = \Sigma f_i^2$

- A, B, B, A, C, D, A, D, B, A, A, D
- $f_A = 5$
- $f_B = 3$
- $f_C = 1$
- $f_D = 3$
- $F_2 = 5^2 + 3^2 + 1^2 + 3^2 = 25 + 9 + 1 + 9 = 44$

Variance

- $Var(X) = E[(X-\mu)^2] = E[X^2] - E[X]^2$
- Measure of the skew of the distribution

Join estimation

- F_2 gives the number of pairs in the self join of R and R
- Also applies to the number of pairs in a join of R and S

Basic Algorithm – Tug-of-war algorithm

- Choose a random hash function h: U → {-1, 1}

```
Y = 0;
foreach x in stream
    Y += h(x);
return Y²;
```

$$Y = \sum_{j=1}^{M} f_j h(j)$$

Hash function assumption
Pairwise independence

- Assumptions on the hash function
  - Prob[h(x) = -1] = 0.5
  - Prob[h(x) = 1] = 0.5
- Hash values are pairwise independent
  - Prob[h(x) = h(y)] = 1/m
  - Knowing the value of h(x) tells you nothing about the value of h(y)
  - Independent random variables

Tug-of-war algorithm is a good estimator of F_2

- Result – Expected value of Y^2 is F_2

Analysis $Y = \sum_{j=1}^{M} f_j h(j)$

$$\begin{aligned}
\mathbf{E}(Y^2) &= \mathbf{E}((\sum_{j=1}^{M} f_j\ h(j))^2) \\
&= \mathbf{E}(\sum_{i=1}^{M}\sum_{j=1}^{M} f_i\ h(i)\ f_j\ h(j)) \\
&= \mathbf{E}(\sum_{j=1}^{M} f_j^2\ h(j)^2 + \sum_{i\neq j} f_i\ h(i)\ f_j\ h(j)) \\
&= \sum_{j=1}^{M} f_j^2 + \sum_{i\neq j} f_i\ f_j\ \mathbf{E}(h(i)h(j))
\end{aligned}$$

If X and Y are independent Random Variables, E(XY) = E(X)E(Y)

$$\text{For}\quad i \neq j, \qquad \text{Prob}[h(i) = h(j)] = \frac{1}{2}$$

$$\mathbf{E}(h(i)h(j)) = \mathbf{E}(h(i))\mathbf{E}(h(j)) = 0 \cdot 0 = 0$$

Improving the algorithm

- Space requirement is just one register
- Improve performance by using more space
- Compute multiple estimates using independent hash functions
  - This is where generating multiple hash functions is important
- Two different ways of combining estimates
  - $E = (1/k)(E_1 + E_2 + \ldots E_k)$
  - $E' = \text{Median}(E_1, E_2, \ldots, E_k)$
- These two methods are combined to get the AKS algorithm

Overall result

- Efficient approximation for F_2
  - $(1 + \varepsilon)$ approximation with probability at least $(1-\delta)$
  - Space requirement $O((1/\varepsilon^2) \log(1/\delta)(\log M + \log N))$
- Extend to higher moments

Deeper analysis, compute Var(Y^2)

$$\begin{aligned}\mathbf{E}(Y^4) &= \mathbf{E}((\sum_{j=1}^{M} f_j\, h(j))^4) \\ &= \mathbf{E}(\sum_{i=1}^{M}\sum_{j=1}^{M}\sum_{k=1}^{M}\sum_{\ell=1}^{M} f_i\, f_j\, f_k\, f_\ell\, h(i)h(j)h(k)h(\ell)) \\ &= \sum_{i=1}^{M}\sum_{j=1}^{M}\sum_{k=1}^{M}\sum_{\ell=1}^{M} f_i\, f_j\, f_k\, f_\ell\, \mathbf{E}(h(i)h(j)h(k)h(\ell))\end{aligned}$$

Need four-wise independence to simplify expectation expression

Universal Family of Hash Functions

- Really good practical hash functions exist
  - Fast and good distribution of keys
  - Cryptographic hash functions are difficult to invert and more work
- Choose a random hash function
  - Set of hash functions $H: U \to [1..m]$
- Universal property
  - For all x, y in U, with $x \neq y$, if h is chosen at random from H

$$\text{Prob}[h(x) = h(y)] \leq \frac{1}{m}$$

  - This is a minimal property for good hash functions
  - Practical university families exist, so mathematically sound algorithms could be implemented

Carter-Wegman hash functions

- Hashing from [0..p-1] to [0..p-1]
- p is a moderate sized prime ($p \cong 2^{32}$)
- $h_{ab}(x) = (ax + b) \bmod p$ where $0 \leq a < p$ and $0 \leq b < p$
- If a and b are chosen at random, $x \neq y$, then $\text{Prob}[h_{ab}(x) = h_{ab}(y)] = \frac{1}{p}$
- $h_{ab}(x)$ and $h_{ab}(y)$ are independent

Pairwise independence

- Suppose $x \neq y$, and $0 \leq x, y \leq p - 1$ and $0 \leq u, v \leq p - 1$
- The equations (with unknowns a and b) have a unique solution
  - $(ax + b) \bmod p = u$
  - $(ay + b) \bmod p = v$
- Hence $\text{Prob}[h(x) = u \text{ and } h(y) = v] = 1/p^2$ proving independence

k-wise independence

- Hash functions such that $h(x_1), h(x_2), \ldots, h(x_k)$ are probabilistically independent
- Important mathematical tool to prove rigorous bounds
- Parameterized hash functions to allow random generation
- In practice, other hash functions may be used which have a seed that can be set

Generalized Carter-Wegman hash functions 4-wise independence

- Hashing from [0..p-1] to [0..p-1]
- p is a moderate sized prime ($p \cong 2^{32}$)
- $h_{abcd}(x) = (ax^3 + bx^2 + cx + d) \bmod p$ where $0 \le a, b, c, d \le p-1$
- Proof of independence is similar to the 2-wise case.
- Show $h_{abcd}(w) = q$, $h_{abcd}(x) = r$, $h_{abcd}(y) = s$, $h_{abcd}(z) = t$ has a unique solution for a, b, c, d over [0..p-1]