Streaming Algorithms

Meng-Tsung Tsai

03/16/2018

References

- "Pairwise Independence and Derandomization," Luby and Wigderson (2005)
- "Sketch Techniques for Approximate Query Processing," Cormode

Count Sketch

Problem Defintion

Input: a sequence of n tuples $(e_1, c_1), (e_2, c_2), \ldots, (e_n, c_n)$ where each e_i in $[U] = \{1, \ldots, U\}$ and each c_i in $\mathbf{Z}$. Let $|U|$ be a prime w.l.o.g.

Output: for each $k \in [U]$, output the frequency

$$f(k) = \sum_{i \in [n]} \mathbf{1}[e_i = k] c_i.$$

In words, $f(k)$ is the sum of c_i's in the sequence whose e_i equals k.

Goal: using $o(U \log C)$ bits to get an approximate $\hat{f}(k)$ for each $f(k)$ where $C = \sum_{i \in [n]} c_i$.

In the setting of Count-Min Sketch, we require all $c_i = 1$. Indeed, if all c_i is non-negative, one can apply the same analysis to prove the error bound. What happens if some c_i's are negative?

Negative c_i's

If some c_i's are negative, then the approximate $\hat{f}(k)$ obtained from the Count-Min sketch may be not an overestimate. (Why?)

Furthermore, if all c_i's are negative, then $\min_j T[j][h_j(k)]$ is the worst estimate of $f(k)$. (Why?)

We thus need an alternative.

Questions to Ponder

Each of k persons bids a price for a ruby ring, so you have prices $p_1, p_2, ..., p_k$.

(1) You don't know the exact price p^* of the ruby ring.

(2) You know that more than $k/2$ prices are within the range

$$[(1-\varepsilon)p^*, (1+\varepsilon)p^*] \text{ for some constant } \varepsilon > 0.$$

Can you also bid a price to within the above range with full confidence?

The median of $p_1, p_2, ..., p_k$.

Consequence

Given k estimates for a value. Some are overestimates, and some are underestimates.

If more than k/2 estimates are good esitmates, then the median of the k estimates cannot deviate from the value too much. In other words, the median is guaranteed to be a good estimate.

Count Sketch is similar to Count-Min Sketch, one of the differences is to replace

$$\min_j T[j][h_j[k]]$$

with

$$\text{median}_j\ T[j][h_j[k]].$$

Another difference

Recall that the expected noise

$$E[\mathscr{E}_j] = \sum_{\ell \neq k} f(\ell)\Pr[h_j(\ell) = h_j(k)] = (n-f(k))/w.$$

Observe that

$$E[\mathscr{E}_j^{+1}] = \sum_{\ell \neq k} f(\ell)\Pr[h_j(\ell) = h_j(k)+1] = (n-f(k))/w. \text{ (Why?)}$$

Observe further that

$$E[\mathscr{E}_j^{-1}] = \sum_{\ell \neq k} f(\ell)\Pr[h_j(\ell) = h_j(k)-1] = (n-f(k))/w.$$

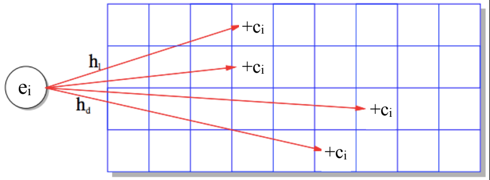
$$E[T[j][h_j(k)]-T[j][h_j(k)+1]] = f(k) + E[\mathscr{E}_j^{+1}] - E[\mathscr{E}_j^{-1}] = f(k).$$

Algorithm

Sample functions $h_1, h_2, \ldots, h_d$ independently, uniformly at random from $H_w = \{h_{a,b,c}(x) = ax^2+bx+c \,\%\, w : a, b, c \in \mathbf{Z}_p\}$ where $w \ll p = |U|$.

$T[d][w] \leftarrow \{0, 0, \ldots, 0\}$.

foreach (e_i, c_i) {
  foreach h_j {
    $T[j][h_j(e_i)]$ += c_i;
  }
}



let $\hat{f}(k) = \text{median}_{j\in[d]} \{T[j][h_j(k)] - T[j][(h_j(k)+1)\%w]\}$;

$E[\hat{f}(k)] = f(k)$.

For each hash function h_j

Let each X_i be a random variable $\{-f(i), 0, +f(i)\}$. If $h_j(i) = h_j(k)+1$, $X_i = -f(i)$. If $h_j(i) = h_j(k)$, $X_i = f(i)$. Otherwise, $X_i = 0$.

In words, X_i is the contribution of e_i to $\hat{f}(k)$.

Let $X = \sum_{i\neq k} X_i$. Then, $\hat{f}(k) = f(k) + X$. Note that $E[X] = 0$.

$\text{Var}[X] = E[(X-E[X])^2] = E[\sum_{i\neq k}(X_i-E[X_i])^2 + \sum_{i\neq k}\sum_{\ell\neq k,\ell\neq i} (X_i-E[X_i])(X_\ell-E[X_\ell])] = \sum_{i\neq k} \text{Var}[X_i] + 0$ (due to 3-wise independence)
$= (2/w)\,(f(i))^2$

Let $F_2 = \sum_{i\in U} f(i)$. By Chebyshev inequality, we get

$$\Pr[|X-E[X]| \geq 2((2/w)F_2)^{1/2}] \leq 1/4.$$

For all hash functions $h_1, h_2, \ldots, h_d$

$\Pr[\text{median}_{j\in[d]} |\mathscr{E}_j| \geq 2((2/w)F_2)^{1/2}]$

$= \Pr[\sum_j \mathbf{1}[|\mathscr{E}_j| \geq 2((2/w)F_2)^{1/2}] \geq d/2]$ (Why?)

$= \Pr[\sum_j \mathbf{1}[|\mathscr{E}_j| \geq 2((2/w)F_2)^{1/2}] \geq (1+1)d/4] \leq \exp(-(1/6)(d/2))$

If we pick $d = 12 \log nU$, then this happens with probability $1/(nU)$.

By the union bound, we get

$$\Pr[|\hat{f}(k) - f(k)| = O((F_2/w)^{1/2}) \text{ for all } k \in U] \geq 1-1/n.$$

Result

By the Count Sketch, one can estimate each $f(k)$ to within the additive error $(\varepsilon F_2)^{1/2}$ with probability at least $1-1/n^{\Omega(1)}$ using $O((1/\varepsilon) \log nU)$ space and $O(n \log nU)$ time.

By Count sketch, can we output a set S so that all $k \in S$ have $f(k) \geq (n)^{1/2}$ with high probability?

Result

By the Count Sketch, one can estimate each f(k) to within the additive error $(\varepsilon F_2)^{1/2}$ with probability at least $1-1/n^{\Omega(1)}$ using $O((1/\varepsilon) \log nU)$ space and $O(n \log nU)$ time.

By Count Sketch, can we output a set S so that w.h.p.

(1) S contains every $k \in U$ that has $f(k) \geq n^{1/2}$, and
(2) every $k \in S$ has $f(k) \geq n^{1/2} - n^{1/3}$?

Count Sketch v.s. Count-Min Sketch

	Count Sketch	**Count-Min Sketch**
error	**under- or over-estimate**	**over-estimate**
error bound	$(\varepsilon F_2)^{1/2}$	εF_1

Let D be the distribution of the frequencies of input elements. If D is close to uniform, then Count-Min Sketch is better. If D is skew, then Count Sketch is better. However, the theoretical analysis is not tight, so their relative order is not predictable.

Compact Representation (1/2 space)

Current: T[j][0] T[j][1] T[j][2] T[j][3] T[j][4] T[j][5]

Alternative: T[j][0] T[j][1] T[j][2] T[j][3] T[j][4] T[j][5]

Compact Representation (1/2 space)

T[j][0]-T[j][1] T[j][2]-T[j][3] T[j][4]-T[j][5]

We can use a half number of entriess to maintain the difference of two consecutive entries.