Streaming Algorithms

Meng-Tsung Tsai

03/30/2018

Reminder

Programming Assignment #0 is due by tonight. You need to submit your program on OJ (https://oj.nctu.me).

Written Assignment #1 is due by 23:59, Apr 10. You need to LaTex your solution and submit it on New E3 (https://e3new.nctu.edu.tw).

You are encouraged to discuss with your classmates, TA, or me. However, the writeup shall be your own.

Schedule

Our 1-hour class on Apr 3 is rescheduled to Apr 10. The class on Apr 10 is from 15:30 - 17:20.

Apr 6 is a holiday. We have no class.

The next lecture will cover approximate quantiles and it takes roughly 2 hours. It is undesired to split the materials into two 1-hour classes, seperated by a long break.

References

- "Pairwise Independence and Derandomization," Luby and Wigderson (2005)
- "The space complexity of approximating the frequency moments," Alon, Matias, Szegedy (Gödel Prize 2005)
- "Sketch Techniques for Approximate Query Processing," Cormode

Frequency Moment

Problem Definition

Input: a sequence of n (possibly repeat) elements $a_1, a_2, ..., a_n$ in [U] = {1, ..., U} and an integer k. Define

$$m_j = \sum_{1 \le i \le n} \mathbf{1}[a_i = j].$$

Output:

$$F_k = \sum_{i \in U} (m_j)^k.$$

Goal: use $o(n \log |U|)$ bits.

We have seen the cases of k = 0, 1. In today's lecture, we will see the cases of larger k.

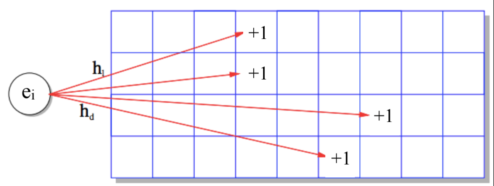
Estimating F_2

Recall Count-Min Sketch

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

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

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



let $\hat{f}(k) = \min_{j \in [d]} \{T[j][h_j(k)]\}$;

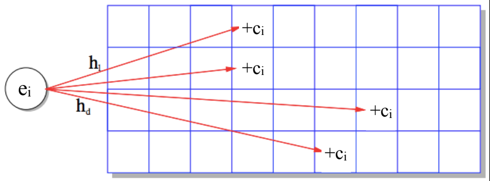
One can get an overestimate for each count up to an error of εF_1 w.h.p.

Recall Count Sketch

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

One can get an over-/under-estimate for each count up to an error of $(\varepsilon F_2)^{1/2}$ w.h.p.

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

It is easy to get F_1. If we have an accurate estimate on F_2, then we may identify which one has a smaller guaranteed error.

Estimating F_2 by Count Sketch (First Attempt)

$\hat{F}_2 \leftarrow 0$;

for k = 1 to |U| {

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

  $\hat{F}_2 \mathrel{+}= \left(\hat{f}(k)\right)^2$;

}

Issues

1. |U| can be much larger than n. For example, U could be $[1, 2^{64}]$ for 64-bit integers. A loop of 2^{64} iterations requires too much time.

2. Counting $\left(\hat{f}(k)\right)^2$ individually have more error than a more careful analysis.

Estimating F_2 by Count Sketch (Second Attempt)

for (j = 0; j < d; j++){

$\hat{F}_{2j} \leftarrow 0$;

for (k = 0; k < w; k+=2) {

$\hat{F}_{2j}$ += (T[j][k] - T[j][k+1])^2;

}

}

$\hat{F}_2 \leftarrow \text{median}_{j \in [d]} \hat{F}_{2j}$;

$$\hat{F}_{2j} = \sum_{k=1}^{|U|} (f(k))^2 + 2\left(\sum_{h(i)=h(i'),i<i'} f(i)f(i')\right) - 2\left(\sum_{h(i')=h(i)+1,h(i)\equiv 0 \pmod 2} f(i)f(i')\right)$$

Estimating F_2 by Count Sketch (Second Attempt)

Given

$$\hat{F}_{2j} = \sum_{k=1}^{|U|} (f(k))^2 + 2\left(\sum_{h(i)=h(i'),i<i'} f(i)f(i')\right) - 2\left(\sum_{h(i')=h(i)+1,h(i)\equiv 0 \pmod 2} f(i)f(i')\right),$$

it follows from the pairwise independence of h that

$$E\left[\hat{F}_{2j}\right] = F_2.$$

Alon et al. show that, if h is 4-wise independent function,

$$\mathrm{Var}\left[\hat{F}_{2j}\right] = O\left(F_2^2/w\right).$$

By Chebyshev Inequality, with a constant probability

$$\left|\hat{F}_{2j} - F_2\right| \leq \lambda F_2/\sqrt{w}.$$

The median of all $\hat{F}_{2j}$ has error bounded by $O(F_2/w^{1/2})$ w.h.p.

Inner Product

Problem Definition

Input: a sequence of n tuples $(a_1, c_1), (a_2, c_2), \ldots, (a_n, c_n)$ where a_i's in [U] and a sequence of m elements $(b_1, p_1), (b_2, p_2), \ldots, (b_m, p_m)$ in [U]. Define

$$c_a(k) = \sum_{i\in[n]} \mathbf{1}[a_i = k]c_i \text{ and } c_b(k) = \sum_{i\in[m]} \mathbf{1}[b_i = k]p_i \text{ for } k \in U.$$

Output:

$$Q = \sum_{k\in U} c_a(k)c_b(k).$$

Goal: use o(n log |U|) bits.

Example: (a_i, c_i) denotes that item a_i is sold c_i times, (b_j, p_j) denotes that item b_j has price p_j. Then, Q is the total sale price.

Estimating Q by Count Sketch

```
for (j = 0; j < d; j++){
```

$\hat{Q}_j \leftarrow 0;$

```
  for (k = 0; k < w; k+=2) {
```

$\hat{Q}_j$ += (T_a[j][k] - T_a[j][k+1])(T_b[j][k]-T_b[j][k+1]);

```
  }
}
```

$\hat{Q} \leftarrow \text{median}_{j\in[d]}\ \hat{Q}_j$;

We need to use the same function to hash a_i's and b_i's to the same row j.

By similar analysis, $|\hat{Q} - Q| = O(\sqrt{F_2(a)F_2(b)/w})$ w.h.p.

Discussions

Higher Moments

We will not cover the analysis for higher moments, but one can guess the following code shall return a good estimate, if ?

```
for (j = 0; j < d; j++){
```

$\hat{F}_{tj} \leftarrow 0;$

```
  for (k = 0; k < w; k+=2) {
```

$\hat{F}_{tj}$ += (T[j][k] - T[j][k+1])^t;

```
  }
}
```

$\hat{F}_t \leftarrow \text{median}_{j\in[d]}\ \hat{F}_{tj};$

T is built by 2t-wise hash functions. Why 2t?

Original AMS F_2 Sketch

Originally, Alon et al. use the approach that w = 1 and d is sufficient large. In that case, every update takes O(d) time and it turns out much slower than what we learn today.