Some Set Ops Material

three authors

ABSTRACT

no abstract yet

1. ESTIMATING SET EXPRESSIONS USING THETA SKETCHES

The primary driver of the design of our theta-sketch system is the problem of producing an estimate Y_r of the number n_r of unique identifiers in a stream $\mathcal{M}_r$ that is never materialized, but instead is defined by a set expression $\xi()$ over a collection $\{\mathcal{M}_j\}$ of streams that *are* materialized and hence can be directly manipulated by the sketching system.

Our system's procedure for producing this estimate Y_r is extremely simple, as shown by the pseudocode in Algorithm ??. We compute the theta sketches, take the minimum, blah, blah, blah.

Empirically, we have found that the system works just fine (no bias, reasonably low variance) for several practical choices for SA(), including Section ??'s novel algorithm based on Morris-style probabilistic counting, and a KMV-style algorithm in which θ is the $k+1$'st minimum value and S is the set of k hash values smaller than θ. More on this later.

It remains an open problem to formally analyze the distribution of Y_r for these practical choices of SA(). However, we have been able to approximately analyze the distribution of Y_r when SA() is the "meta algorithm" whose pseudocode is presented as Algorithm ??. This meta algorithm takes as an argument any "base" algorithm for producing a theta sketch. For example, the base algorithm could be the Morris-style algorithm or the KMV-style algorithm mentioned above. When the meta algorithm processes a stream, it first generates a new "private" fully independent hash function H_p, then computes a theta sketch (θ, S_p) for the stream by running the base algorithm with the private hash function. It then discards the hash function $H_p()$ and the set S_p, and computes a different set S by "rescanning" the stream using the globally shared hash function SH(), collecting all hash values less than θ. Finally, it returns the theta sketch (θ, S), which has the theoretically convenient property that the value of θ is independent of the hash function which generated the hash values in S.

We will now characterize the expected value and variance of Y_r,

Algorithm 1 Set Expression Estimation Procedure

```
1: Inputs: (SetExpression $\xi$, $m$ Streams $\{\mathcal{M}_j\}$ of Size $\{n_j\}$,
   $k$, Shared Hash Function SH(), Sketching Algorithm SA()))
2: for all Streams $\mathcal{M}_j$ do
3:     $(\theta_j, S_j) \leftarrow \mathrm{SA}(k, SH(), \mathcal{M}_j)$   .
4: end for
5: $\theta_r \leftarrow \min(\{\theta_j\})$.
6: $S_r = \{h | (h \in \cup S_j) \wedge (h < \theta_r) \wedge (\xi(h) = \mathrm{true})\}$
7: Output the estimate $Y_r = \frac{|S_r|}{\theta_r}$ and the sketch $(\theta_r, S_r)$.
```

the estimate of $|\mathcal{M}_r|$ produced by Algorithm ??, under the assumption that the sketching algorithm SA() called by Algorithm ?? is in fact the meta algorithm presented as Algorithm ??.

We will show that Y_r is unbiased pretty much no matter what base algorithm is called by the meta algorithm.

However, the variance of Y_r does depend on details of the base algorithm. Also, the analysis is quite complicated, so at a certain point we will resort to approximations and/or numerical calculations. However, even though we will not end up with exact results in a nice closed form, our analysis shows that this method works very well.

1.1 Expected Value of Y_r

Preliminaries: WLOG, we assume that $n_1 = \max\{n_i\}$. Also, whenever we write $f(x|\mathrm{BA})$ we really mean $f(x|\mathrm{BA}(), k, \{n_i\}, n_r)$.

$$E(Y_r|\mathrm{BA}) \tag{1}$$

$$= \sum_{\theta_r} \Pr(\theta_r|\mathrm{BA}) \sum_b \frac{b}{\theta_r} \mathrm{Bino}(b; n_r, \theta_r) \tag{2}$$

$$= \sum_{\theta_r} \Pr(\theta_r|\mathrm{BA}) \frac{1}{\theta_r} (n_r \cdot \theta_r) \tag{3}$$

$$= n_r \cdot 1 \tag{4}$$

So Y_r is unbiased.

1.2 Variance of Y_r

Preliminaries: WLOG, we assume that $n_1 = \max\{n_i\}$. Also, whenever we write $f(x|\text{BA})$ we really mean $f(x|\text{BA}(), k, \{n_i\}, n_r)$.

$$\sigma^2(Y_r|\text{BA}) \tag{5}$$

$$= -n_r^2 + \sum_{\theta_r} \Pr(\theta_r|\text{BA}) \sum_b \frac{b^2}{\theta_r^2} \text{Bino}(b; n_r, \theta_r) \tag{6}$$

$$= -n_r^2 + \sum_{\theta_r} \Pr(\theta_r|\text{BA}) \frac{1}{\theta_r^2} [n_r\theta_r - n_r\theta_r^2 + n_r^2\theta_r^2] \tag{7}$$

$$= -n_r + n_r \sum_{\theta_r} \Pr(\theta_r|\text{BA}) \frac{1}{\theta_r} \tag{8}$$

$$= -n_r + n_r \cdot E(\frac{1}{\theta_r}|\text{BA}) \tag{9}$$

$$= -n_r + \frac{n_r \cdot n_1}{k} \cdot \left[\frac{k}{n_1} \cdot E\left(\frac{1}{\min(\{\theta_j\})}|\text{BA}\right)\right] \tag{10}$$

$$= -n_r + \frac{n_r \cdot n_1}{k} \cdot [1 + 2 \cdot \text{wcp}(\text{BA}, k, \{n_i\})] \tag{11}$$

where the "winners curse penalty" wcp() is defined as follows:

$$\text{wcp}(\text{BA}, k, \{n_i\}) = \frac{1}{2}\left(\frac{k}{n_1} E\left(\frac{1}{\min(\{\theta_j\})}|\text{BA}\right) - 1\right) \tag{12}$$

Then the standard error of the estimate Y_r is:

$$\text{S.E.}(Y_r|\text{BA}) \tag{13}$$

$$= \sqrt{\frac{n_1}{n_r} \frac{1}{k} (1 + 2\text{wcp}(\text{BA}, k, \{n_i\})) - \frac{1}{n_r}} \tag{14}$$

$$< \sqrt{\frac{n_1}{n_r} \frac{1}{k} (1 + 2\text{wcp}(\text{BA}, k, \{n_i\}))} \tag{15}$$

$$< (1 + \text{wcp}(\text{BA}, k, \{n_i\})) \cdot \sqrt{\frac{n_1}{n_r}} \cdot \sqrt{\frac{1}{k}} \tag{16}$$

1.3 Informal Analysis of wcp()

We will now perform an informal approximate analysis of wcp(). The analysis is approximate because it assumes that certain not-quite-gaussian probability distributions are gaussian. Having made that assumption, we will make use of the already existing analysis of the expected value of m iid gaussian random variables G_j each with mean μ and standard deviation σ, which states that:

$$E(max(\{G_j\}) = \mu + c_m\sigma \tag{17}$$

where the c_m's are known constants; approximate values for some of them are shown in table ??.

In fact, during this approximate analysis, we will be making *three* new assumptions, and will also be using a property of the "meta algorithm" that we haven't used until now.

- Assumption 1: All of the input stream cardinalties n_j are equal to n_1, the largest input stream cardinality. We believe, but have not proved, that this is a worst-case assumption.
- Assumption 2: The base algorithm's estimate of each input stream cardinality n_j has the form $\hat{n}_j = k/\theta_j$. This assumption is true for both KMV and MPC.
- Assumption 3: The base algorithm's distribution of estimates is assumed to be Gaussian, with the same mean and variance as the actual, not-quite-gaussian distribution of estimates.
- Property of the Meta Algorithm: the threshold θ_j for each input stream is computed using a different independent hash function.

Under those assumptions, we will now analyze wcp(BA,k, n_1, m).

$$\text{wcp}(\text{BA}, k, n_1, m) = \frac{k}{2n_1} E\left(\frac{1}{\min(\{\theta_j\})}|\text{BA}\right) - \frac{1}{2} \tag{18}$$

$$= \frac{1}{2n_1} E(\max(\{\hat{n}_j\})|\text{BA}) - \frac{1}{2} \tag{19}$$

$$\approx \frac{1}{2n_1}(n_1 + c_m\sigma(BA)) - \frac{1}{2} \tag{20}$$

$$= \frac{c_m\sigma(BA)}{2n_1} \tag{21}$$

$$\tag{22}$$

Now, specializing for BA=MPC and BA=KMV:

$$\text{wcp}(\text{KMV}, k, n_1, m) \approx \frac{c_m\sigma(KMV)}{2n_1} \tag{23}$$

$$< \frac{c_m \frac{n_1}{\sqrt{k-1}}}{2n_1} \tag{24}$$

$$= \frac{c_m}{2\sqrt{k-1}} \tag{25}$$

$$\text{wcp}(\text{MPC}, k, n_1, m) \approx \frac{c_m\sigma(MPC)}{2n_1} \tag{26}$$

$$< \frac{c_m \frac{n_1}{\sqrt{2k}}}{2n_1} \tag{27}$$

$$= \frac{c_m}{2\sqrt{2k}} \tag{28}$$

The following table contains some approximate values for c_m, wcp(KMV), and wcp(MPC).

k	m	c_m	wcp(KMV)	wcp(MPC)
256	2	0.564	1.767%	1.247%
	4	1.029	3.223%	2.275%
	8	1.424	4.457%	3.146%
4096	2	0.564	0.441%	0.312%
	4	1.029	0.804%	0.569%
	8	1.424	1.112%	0.786%
65536	2	0.564	0.110%	0.078%
	4	1.029	0.201%	0.142%
	8	1.424	0.278%	0.197%

These approximate values suggest the following general statements.

1. wcp is at most a few percent in practical scenarios.
2. wcp grows with the number of streams involved in the set expression, but only logarithmically in m.
3. wcp is smaller for base algorithms whose estimates are more concentrated (e.g. for MPC as opposed to KMV).

Finally, we claim that the above approximate values of wpc aren't very far from the true values, despite the fact that we used a Gaussian approximation without justifying it. For example, the approximate value for wcp(MPC,k=256) listed above 1.247 percent, while the true value (computed numerically using eqn ?? and the recursion developed in section ??) is 1.245 percent.

2. INFORMAL ANALYSIS OF THE I/E RULE FOR SET INTERSECTION

Just for the purposes of obtaining a curve that we can compare against the one derived in the previous section, we will now informally analyze the Include/Exclude rule as applied to the problem of estimating the size of the intersection of two sets. We will use a and b to denote the sizes of the two sets, and will in fact only consider the case $a = b$. We will use u and r to denote the sizes of the union and intersection of the two sets. Instead of analyzing a specific sketching method, we will just say that it is some method that is unbiased, whose variance is $n^2/k - n$, and which uses a different independent hash function when processing each stream. The input to the include/exclude rule will be the independent estimates of a, b, and u, obtained by separately processing the first stream, the second stream, and the stream defined by their union. Then:

$$\sigma^2(\mathrm{est}(r)) \tag{29}$$

$$= \left(\frac{a^2}{k} - a\right) + \left(\frac{b^2}{k} - b\right) + \left(\frac{u^2}{k} - u\right) \tag{30}$$

$$= 2\left(\frac{a^2}{k} - a\right) + \left(\frac{4a^2 - 4ar + r^2}{k}\right) - (2a - r) \tag{31}$$

$$= \frac{1}{k}(6a^2 - 4ar + r^2) - 4a + r \tag{32}$$

$$\mathrm{S.E.}(\mathrm{est}(r)) = \sqrt{\frac{1}{r^2} \cdot \left(\frac{1}{k}(6a^2 - 4ar + r^2) - 4a + r\right)} \tag{33}$$

$$\approx 2.45 \cdot \frac{a}{r} \cdot \sqrt{\frac{1}{k}} \tag{34}$$

where the final approximation applies when $r \ll a$ and $a \gg k$.

This formula can be compared with the theta-sketch formula

$$\mathrm{S.E.}(Y_r|\mathrm{MPC}) \approx 1.01 \cdot \sqrt{\frac{a}{r}} \cdot \sqrt{\frac{1}{k}} \tag{35}$$

The biggest difference between them is that the I/E rule doesn't have a square root on the inverse Jaccard Ratio.

theoretical-ie-vs-theta.pdf

The impact of this difference is shown graphically in Figure ??, which plots equation ?? and equation ?? for a hypothetical setup where $a = b = 100000$, and r varies from 1 to 100000. For I/E, we plotted equation ?? with k=4096. For theta-sketches, we plotted equation ?? with k=256 and wcp = 0.01247. We gave I/E a bigger value of k to reflect the fact that HLL uses fewer bits per sample than the theta-sketch system. Because it gets a bigger value of k, I/E is actually better when the intersection is almost as big as the input sets. However, this advantage cannot make up for the worse slope of the I/E curve, which causes its standard error to be extremely bad when the intersection is small.