Towards Optimal Cardinality Estimation of Unions and Intersections with Sketches

Daniel Ting
Facebook
1730 Minor Ave
Seattle, WA
dting@fb.com

ABSTRACT

Estimating the cardinality of unions and intersections of sets is a problem of interest in OLAP. Large data applications often require the use of approximate methods based on small sketches of the data. We give new estimators for the cardinality of unions and intersection and show they approximate an optimal estimation procedure. These estimators enable the improved accuracy of the streaming MinCount sketch to be exploited in distributed settings. Both theoretical and empirical results demonstrate substantial improvements over existing methods.

CCS Concepts

•**Mathematics of computing** → **Probabilistic algorithms;** •**Theory of computation** → **Sketching and sampling;** •**Computing methodologies** → *Distributed algorithms;*

Keywords

cardinality estimation; data sketching; randomized algorithms

1. INTRODUCTION

Consider a dataset $\mathcal{D}$. The basic approximate distinct count problem is to construct a memory efficient summarization S of $\mathcal{D}$ and to estimate the cardinality of the set of unique items A using just the sketch S. Under this basic formulation, a sketch answers a cardinality question that must be specified before computing the sketch. We consider the extended problem of estimating the cardinality of set unions, $|A_1 \cup A_2|$, and intersections, $|A_1 \cap A_2|$, using using multiple sketches. This has two important consequences. First, combinatorially many cardinality questions can be accurately answered by sketches. Second, the significant improvements in accuracy using the HIPS or optimal martingale estimator of [10] and [23] can be realized in distributed settings. More specifically, we wish to define union $\tilde{\cup}$ and intersection $\tilde{\cap}$ operations on sketches and derive cardinality estimates are optimal or nearly optimal.

This problem of estimating the number of distinct elements in a dataset in a memory efficient manner appears in a wide range of applications. For example, in data analytics and OLAP, a web company may count distinct users accessing a service [21]. The summarizations are used in networking to detect denial of service attacks by counting network flows [15], and in databases to optimize query plans [22]. Other applications include graph analysis where they are used to estimate the diameter of a graph [9], [5].

The extended problem for unions has particular importance in OLAP and distributed settings. The streaming cardinality estimator given by [23], [10] are optimal in the sense that no unbiased estimator has lower asymptotic variance. This is much stronger than typical space-complexity results in the theory literature [2], [19] that just guarantee an optimal rate. It guarantees optimality of the constant in front of the rate. This optimality is evident in practice as the estimators require half the space of existing methods to achieve the same error on MinCount sketches. The streaming estimator, however, cannot be applied directly in distributed settings. In a map-reduce setting, the mappers computing efficient streamed sketch summaries need a method to combine them at a reducer. A union operation on the sketches would allow the efficiency gains to be transfered to distributed settings and turn the streamed sketch into a mergeable summary [1]. These efficiency gains can be tremendous as empirical results show efficient cardinality estimation for unions can reduce variance and the space requirements by an order of magnitude when many sketches are merged. In OLAP settings, one is often interested in cardinality estimates on multi-dimensional data, for example counting the number of users accessing a web service by geographic location, time window, and other factors [21]. This results in exponentially many cardinalities to estimate. Unions and intersections allow a limited number of sketches to answer a multitude of cardinality questions.

Cardinality estimates for intersections are not as well studied as for unions. While most sketches for approximate distinct counting have a natural but inefficient union operation, they do not have one for intersections. Intersection cardinality estimates are often computed using the inclusion-exclusion principle or by using Jaccard similarity [13], [4] when an estimator is available. These methods still require first accurately estimating the cardinality of the union or the Jaccard similarity of a set with the union.

Our contributions are as follows. We introduce two tech-

KDD '16, August 13 - 17, 2016, San Francisco, CA, USA

ISBN 978-1-4503-4232-2/16/08...$15.00

DOI: http://dx.doi.org/10.1145/2939672.2939772

niques for computing unions and intersections on approximate distinct counting sketches: pseudo-likelihood based methods and re-weighting component estimators. These are applied to Streaming MinCount sketches in this paper but may be applied to other sketches as well. The resulting estimators demonstrate state of the art performance on both real and synthetic datasets. The pseudo-likelihood based MinCount method is conjectured to be asymptotically efficient as it is nearly the same as the asymptotically efficient maximum likelihood estimator. This establishes a nearly ideal baseline for comparison. Re-weighted component estimators are much simpler to implement and generalize, and they are empirically shown to be as efficient as the pseudo-likelihood based estimators. We also derive variance estimates which allow confidence intervals to be given for the cardinality estimates. In the special case where the streams are different permutations of the same set, we show that merging the streaming estimates yields a more accurate estimate than the estimator on a single stream. Thus, unlike existing methods, the union operation on streaming sketches exploits information about the order of elements in each stream. The variance after averaging over all possible orderings is shown to be 2/3 the variance of the streaming MinCount estimator. In addition to improved estimation, the resulting methods yield mergeable summaries [1] under both union and intersection operations unlike existing methods.

2. EXISTING BASIC METHODS

The basic approximate distinct counting problem has been well studied in the literature beginning with the seminal paper by Flajolet and Martin [17]. Methods include Linear Probabilistic Counting (LPCA) [25], HyperLogLog [16], the Multiresolution Bitmap [15], the S-bitmap [7], α-stable distribution methods [12], and MinCount [18]. Kane, et al [19] also gave a method that is optimal in terms of space-complexity.

As described in [23], all these methods generate a stochastic process called the area process that is governed solely by the number of distinct elements encountered. A sketch S can be mapped to a set $\mathcal{R}(S) \subset (0,1)$ called the remaining area. Given a random hash, each element in a stream is hashed to some value U. If U is still in the remaining area, $U \in \mathcal{R}(S)$, the sketch is updated by cutting out a portion of the remaining area containing U. Otherwise, it does not affect the sketch. Since a value U cannot be cut out twice, duplicates do not affect the sketch. Under the assumption that the hash function is a strong universal hash, the original distribution of the data is irrelevant. A sketch is a function of the independent $Uniform(0,1)$ random variates that are the hashed distinct items. It is a random quantity whose distribution depends only on the number of distinct items. Cardinality estimation is thus a statistical parameter estimation problem where the sketch is the observed data.

Using an optimal estimation procedure can lead to substantial efficiency gains. For example, the LogLog and SuperLogLog methods [14] share the same sketch as HyperLogLog and differ only in the estimation procedure. However, LogLog requires 1.56 times the memory of HyperLogLog to achieve the same error and 2.4 times the memory of Streaming HyperLogLog. Likewise, the original cardinality estimation method for the α-stable sketch proposed in [12] uses over twice the memory as that used in [8]. The disadvantage of using the optimal streaming methods described in [23] and [10] is that they require the data to arrive in a single stream. They are not immediately applicable in distributed settings or in OLAP settings where the sketches are pre-aggregations that need to be further merged.

3. NAIVE ESTIMATION

We first consider basic estimation procedures for unions and intersections on distinct counting sketches. Many sketches used in approximate distinct counting have a natural union operation. For example, the LPCA sketch is a bitmap and is equivalent to a Bloom filter with the number of hashes $k = 1$. Taking the union of LPCA sketches simply takes the bitwise OR of the sketches. The resulting sketch is identical to the sketch computed on the union of the original sets. This property holds for other sketches such as HyperLogLog [16] and MinCount [18] as well. In other words, there is an operation $\tilde{\cup}$ such that $S(A_1)\tilde{\cup}S(A_2) = S(A_1 \cup A_2)$ where S is the function that generates a sketch from a streamed set. For notational convenience, we will denote $S_i = S(A_i)$.

Unlike the union, typically no natural intersection operation exists for approximate distinct counting sketches. A simple method to obtain cardinality estimates for the intersection is to use union estimates and the inclusion-exclusion principle, $|A_1 \cap A_2| = |A_1| + |A_2| - |A_1 \cup A_2|$. In the case where there is an estimate $\hat{J}(S_1, S_2)$ for the Jaccard similarity, two existing intersection cardinality estimator are given by [4] and [13]. Throughout, we will use a hat to denote estimated quantities, with $\hat{N}(S)$ denoting a cardinality estimate using sketch S. The naive and Jaccard based estimates of the cardinality of the union and intersection are given by

$$\begin{aligned}
\hat{N}_{naive}(S_1\tilde{\cup}S_2) &= \hat{N}(S(A_1 \cup A_2)) && (1)\\
\hat{N}_{naive}(S_1\tilde{\cap}S_2) &= \hat{N}(S_1) + \hat{N}(S_2) - \hat{N}(S(A_1 \cup A_2))\\
\hat{N}_{Jaccard,1}(S_1\tilde{\cap}S_2) &= \hat{J}(S_1, S_2)\hat{N}(S_1\tilde{\cup}S_2)\\
\hat{N}_{Jaccard,2}(S_1\tilde{\cap}S_2) &= \frac{\hat{J}(S_1, S_2)}{1 + \hat{J}(S_1, S_2)}\left(\hat{N}(S_1) + \hat{N}(S_2)\right)
\end{aligned}$$

These estimation strategies turn out to be suboptimal. For the intersection estimators, one reason is that the error is often roughly proportional to the size of the union or the larger set, while a good procedure should give error that is bounded by the size of the smaller set. This can also lead to pathologies where the naive cardinality estimate of the intersection is negative. If a simple correction to replace these negative estimates by zero is used, the resulting estimator is provably biased.

4. STREAMING MINCOUNT

In this paper we focus on applying new estimation techniques to Streaming MinCount sketches. The techniques may be used for other sketches, and their application is discussed in section 13. The choice of Streaming MinCount is driven by two reasons. The first is that it simplifies the calculations since collision probabilities are negligible. The second is that the uniqueness of hash values allows for a closed intersection operation and gives more flexibility to the sketch.

The basic MinCount sketch stores the k minimum hash values and is also known as the K-minimum values [2] or bottom-k [11] sketch. When the hash values are uniformly distributed on $(0,1)$, the estimator for the MinCount sketch is $(k-1)/\tau$ where τ is the largest stored hash value. It is easy to see that this estimator approximates the cardinality since the k^{th} smallest hash value out of n roughly uniformly spaced values is approximately k/n. It can be shown that the estimator is the unique minimum variance unbiased estimator [6] when only the final set of hash values is observed.

The Streaming MinCount sketch augments the basic MinCount sketch with a running estimate of the cardinality. This estimate is given by

$$\hat{N}(S) = \sum_{t=1}^{n} \frac{Z_t}{\tau_t} \tag{2}$$

where τ_t is the threshold for sketch S after encountering t elements and $Z_t = 1$ if the sketch changed at time t. The threshold for a sketch is the largest stored hash value. The streaming estimator operates by incrementing the running estimate by 1 in expected value for each newly encountered element. By exploiting the sequence of sketches rather than just the final sketch, the streaming update procedure reduces the variance of the estimator by half.

5. IMPROVED ESTIMATORS

The usual union operation throws away valuable information for the MinCount sketch. To see this consider the case where the sets A_1, A_2 are disjoint and of equal cardinality. The best estimator would simply add $\hat{N}(S_1 \cup S_2) = \hat{N}(S_1) + \hat{N}(S_2)$ which has variance $(|A_1|^2 + |A_2|^2)/k = |A_1 \cup A_2|^2/2k$. However, the MinCount sketch throws away half of the hash values to maintain the size limit of k hash values. The resulting union estimate correspondingly has twice the variance, $|A_1 \cup A_2|^2/k$. This is the strategy employed by [4].

We introduce a simple improvement by constructing the largest possible merged MinCount sketch rather than fixing the size at k. Let $\tau(S)$ be the threshold, the largest stored hash, for sketch S. Let $h(S)$ denote the set of hashes stored in sketch S and $h(S, \tau')$ be the set that is $\leq \tau'$. For convenience, denote $\tau_i = \tau(S_i)$. Improved union and intersection operators $\cup_0, \cap_0$ on MinCount sketches are defined by

$$\begin{aligned}
\tau_{min} &:= \tau(S_1 \cup_0 S_2) := \tau(S_1 \cap_0 S_2) := \min\{\tau_1, \tau_2\} \\
h(S_1 \cup_0 S_2) &:= h(S_1, \tau_{min}) \cup h(S_2, \tau_{min}) \\
h(S_1 \cap_0 S_2) &:= h(S_1) \cap h(S_2)
\end{aligned}$$

In other words, the hash values larger than the minimum threshold are discarded, and the sketches are merged by taking the union or intersection of the remaining hash values.

The resulting union sketch is exactly the same as a MinCount sketch of size $|h(S_1 \cup S_2)|$ constructed from $A_1 \cup A_2$. The intersection sketch is similar but allows for the threshold to not be in the intersection. This yields a closed intersection operator. It generates a new sketch and not just an estimate of the cardinality. The union and intersection estimators are

$$\hat{N}_{improved}(S_1 \cup_0 S_2) = \frac{|h(S_1 \cup_0 S_2)| - 1}{\tau_{min}} \tag{3}$$

$$\hat{N}_{improved}(S_1 \cap_0 S_2) = \frac{|h(S_1 \cap_0 S_2)| - \delta(S_1, S_2)}{\tau_{min}} \tag{4}$$

where $\delta(S_1, S_2) = 1$ if $\tau_{min} \in h(S_1 \cap_0 S_2)$ and is 0 otherwise.

This improvement may also be applied to the Discrete MaxCount sketch introduced by [23]. However, there is no obvious generalization of this improvement to the HyperLogLog sketch. For the Streaming MinCount sketch, although it does not define a union or intersection cardinality estimator that exploits the gains from the streaming estimates $\hat{N}(S_i)$, it allows a closed intersection operation to be defined.

6. STATISTICAL EFFICIENCY

The improvements proposed in the previous section are suboptimal in terms of statistical efficiency. Statistical efficiency is a far more stringent optimality criterion than optimal space-complexity when comparing estimators. A consistent estimator $\hat{\theta}_n$ of a parameter θ is asymptotically (statistically) efficient if its asymptotic variance is equal to the Cramer-Rao lower bound on the variance [24]. Whereas space-complexity ensures that the error rate is optimal, asymptotic efficiency ensures both the rate and the constant factor governing the rate are optimal. This is an important distinction, since the rate is typically meaningless for estimation. Under mild regularity conditions, parameter estimation problems on i.i.d. observations invariably have an optimal rate of $\Theta(1/\sqrt{n})$, and almost any reasonable estimation procedure achieves that rate. Only the constant governing the rate yields meaningful comparisons for estimators. By contrast, space-complexity is meaningful when considering the problems of constructing and encoding a sketch but not for estimation [23].

Under mild regularity conditions, the maximum-likelihood estimator (MLE) is an asymptotically efficient estimator. Although we are not able to derive the exact MLE, in section 8.2 we derive both the conditional likelihood given the cardinality of the input sets as well as a full pseudo-likelihood. We conjecture that maximizing the full pseudo-likelihood gives an estimator that is asymptotically equivalent to the MLE. Although a full proof is out of the scope of this paper, we give a heuristic proof sketch that may be able to establish the asymptotic efficiency.

In addition to the class of likelihood based estimators, we derive a class of re-weighted estimators based on linear combinations of the streaming cardinality estimates. These estimators have the advantage of being easy to implement and easy to generalize to other sketches. Although we show how to optimally weight the linear combinations, the resulting estimators are not theoretically guaranteed to be asymptotically efficient, but empirical results suggest that they are efficient or close to efficient.

7. NOTATION AND ASSUMPTIONS

The remainder of the paper focuses on exploiting the efficiency gains from streaming estimation to improve cardinality estimation for union and intersection.

To fix some notation, suppose each sketch S_i contains k_i hash values and estimates the cardinality $|A_i| = n_i$. The total cardinality is $n_{tot} = |\cup_i A_i|$. The threshold τ_i for a sketch S_i is the maximum stored hash value. The proportion of the total that belongs to set A_i is denoted by $p_i = |A_i|/n_{tot}$. For simplicity, only pairwise unions and intersections are analyzed in detail, but merging multiple sketches is discussed in section 11 In the pairwise case, define $q_0 = |A_1 \cap A_2|/n_{tot}, q_1 = |A_1 \backslash A_2|/n_{tot}$, and $q_2 = |A_2 \backslash A_1|/n_{tot}$.

For the analysis of the methods, we consider the asymptotic regime where for each i, $k_i/n_{tot} \to 0$ and $q_i \to c_i > 0$

as $n_{tot} \to \infty$ for some constants c_i. The hash function is assumed to be a strong universal hash that generates i.i.d. $Uniform(0,1)$ random variates.

8. LIKELIHOOD BASED METHODS

As described in section 2, cardinality estimation is a statistical parameter estimation problem. Under this formulation, statistical theory provides two important results. The useful pieces of information in the sketch are encoded by sufficient statistics, and the maximum likelihood estimator is an asymptotically efficient estimator.

8.1 Sufficient statistics

The notions of sufficiency and ancillarity are important in statistical estimation. A statistic is sufficient if it contains all the information necessary to reconstruct the likelihood function. Under the likelihood principle, the likelihood contains everything useful for parameter estimation. A statistic is ancillary if it is irrelevant for estimating the parameter. More formally, it is a statistic whose distribution does not depend on the parameter of interest. This gives a basic strategy for finding good estimators. Find the smallest sufficient statistic containing as few ancillary statistics as possible. Propose an estimator that is a function of the sufficient statistic.

For the MinCount sketch, the exact values of the stored hashes relative to the threshold are irrelevant for estimation and, hence, ancillary statistics. The value of the threshold, the largest hash, is a sufficient statistic when only the final set of hashed values is observed. Furthermore, this threshold is a complete and minimal sufficient statistic, so by the Lehman-Scheffe theorem, the usual MinCount estimator is a minimum variance unbiased estimator as shown by [6].

Likewise, the exact values of the stored hashes are irrelevant when estimating the cardinality of the union or intersection of two sets. Assuming the sizes of the sketches for A_1, A_2 are fixed at k, a set of sufficient statistics for the cardinality of the union and intersection is given by the thresholds of the individual sketches, τ_1 and τ_2, the number of common stored hashes $|h(S_1 \cap_0 S_2)|$, the total number of hashes less than or equal to the smaller of the two thresholds $|h(S_1 \cup_0 S_2)|$, and the streaming estimates of the cardinality $\hat{N}(S_i)$ and variance $\hat{\mathrm{Var}}(S_i)$.

8.2 Likelihood

The first class of estimators we present are the likelihood based estimators. Given a sufficient statistic, the parameter of interest may be estimated using the asymptotically efficient maximum likelihood estimator (MLE). The asymptotic variance of the MLE is given by the inverse Fisher information. Under mild regularity conditions, this is the expected Hessian of the negative log-likelihood evaluated at the true parameters. This yields a natural estimate for the variance of the MLE as well. Compute the Hessian of the negative log-likelihood at the estimated rather than the true parameters and take its inverse.

A closed form for the full likelihood of the Streaming MinCount sketch is not known. Instead, we first derive the likelihood function for the basic MinCount sketch. By plugging in the streaming estimates into this likelihood, it can be used to construct a surrogate likelihood or a pseudo-likelihood [3] for the intersection or union. This surrogate is a form of profile-likelihood for the Streaming Mincount sketches that include the streaming estimates as part of the sketches. We then derive an approximation to the full likelihood.

We first derive the generative process for a pair of basic MinCount sketches from sets A_1, A_2 in lemma 1. This allows derivation of the likelihood. A rough description of the process is as follows. First, generate a threshold τ_1 and propose values for $|h(S_1 \cap_0 S_2)|$ and $|h(S_2, \tau_1) \backslash h(S_1, \tau_1)|$ without considering the constraint that S_2 contains at most k_2 hash values. If their total is less than k_2 and the constraint on S_2 is not violated, then $\tau_1 = \tau_{min}$, and to complete sketch S_2, simply draw the requisite number of additional points above τ_1 to compute τ_2. Otherwise, $\tau_2 < \tau_1$ and the proposed values are thinned using sub-sampling.

LEMMA 1. *Given sets A_1, A_2 and a random universal hash, MinCount generates random sketches S_1, S_2 of sizes k_1, k_2 respectively. The sufficient statistics for the parameters $|A_1|$, $|A_2|$, and $|A_1 \cap A_2|$ are τ_1, τ_2, $|S_1 \cup_0 S_2|$, and $|S_1 \cap_0 S_2|$. These statistics have the distribution given by the following process.*

$$\begin{aligned} \tau_1 &\sim Beta(k_1, |A_1| - k_1 + 1) \\ U &\sim HyperGeometric(|A_1 \cap A_2|, |A_1 \backslash A_2|, k_1) \\ V &\sim Binomial(|A_2| - U, \tau_1) \\ C &\sim Bernoulli(U/k_1) \end{aligned}$$

If $U + V < k_2$ then

$$\begin{aligned} |S_1 \cap_0 S_2| &= U, \quad |S_1 \cup_0 S_2| = k_1 + V \\ \frac{\tau_2 - \tau_1}{1 - \tau_1} &\sim Beta(k_2 - U - V, |A_2| - k_2 + 1) \end{aligned}$$

If $U + V = k_2$ and $C = 1$,

$$|S_1 \cap_0 S_2| = U, \quad |S_1 \cup_0 S_2| = k_1 + V, \quad \tau_2 = \tau_1$$

Otherwise, $\tau_2 < \tau_1$ and

$$\begin{aligned} |S_1 \cap_0 S_2| &\sim HyperGeometric(U, V, k_2) \\ \tau_2 &\sim \tau_1 \times Beta(k_2, U + V - C - k_2 + 1) \\ |S_1 \cup_0 S_2| &\sim k_2 + Binomial(k_1 - U - (1 - C), \tau_2/\tau_1) \end{aligned}$$

PROOF. The order statistics of the hash values determine the MinCount sketch. The order statistics of n uniform random variates are jointly Dirichlet distributed with parameter $1 \in \mathbb{R}^n$, and the marginal distributions are Beta distributed. The labeling of uniform random variates as belonging to $A_1 \backslash A_2$, $A_2 \backslash A_1$, or $A_1 \cap A_2$ follows a multinomial distribution. The lemma follows from conditional distributions for the Dirichlet and multinomial distributions and their properties. □

Note that in the case where $U + V - C < k_2$ or, equivalently $\tau_2 \geq \tau_1$, every generated variable is also observed in the final sketch. There are no hidden variables that need to be integrated out to form the likelihood. By symmetry, exchanging the indices of the variables and parameters does not change the likelihood. This observation reduces the computation of the log-likelihood to the simple case with no hidden variables and gives the following theorem.

THEOREM 2. *Let variables $x_1, x_2, x_{1\cap 2}$ represent estimates of $|A_1|, |A_2|, |A_1 \cap A_2|$, respectively. The log-likelihood of the*

MinCount sketch may be written as follows.

$$\begin{aligned}&\ell_0(x_1, x_2, x_{1\cap 2}; S_1, S_2)\\&= 1(\tau_2 \geq \tau_1) \log p\left(\tau_1, \tau_2, H \middle| x_1, x_2, x_{1\cap 2}\right)\\&+ 1(\tau_1 > \tau_2) \log p\left(\tau_2, \tau_1, H \middle| x_1, x_2, x_{1\cap 2}\right)\end{aligned}$$

where $H = (|S_1 \cap_0 S_2|, |S_1 \cup_0 S_2|)$ and the conditional probabilities are given by the generative process in lemma 1.

To obtain an estimator that exploits the streaming cardinality estimates, simply replace $x_1 = \hat{N}(S_1)$ and $x_2 = \hat{N}(S_2)$ to obtain a marginal profile likelihood.

$$\hat{N}_{profile}(S_1 \cap S_2) = \arg\max_{x_{1\cap 2}} \ell_0(\hat{N}(S_1), \hat{N}(S_2), x_{1\cap 2}) \quad (5)$$

$$\hat{N}_{profile}(S_1 \cup S_2) = \hat{N}(S_1) + \hat{N}(S_2) - \hat{N}_{profile}(S_1 \cap S_2).$$

After taking a union or intersection, the resulting sketch contains the new cardinality estimate along with the hash values stored by the improved MinCount sketch. Note that although the union cardinality estimate uses the inclusion-exclusion principle, it is simply a reparameterization of the intersection cardinality, and maximum likelihood estimates are invariant under reparameterization.

Since some of the distributions have discrete parameter spaces, we relax the optimization of the log-likelihood by replacing factorials with the corresponding continuous gamma function when doing maximum likelihood estimation.

8.3 Full pseudo-likelihood

The profile-likelihood given above has a few deficiencies. It is asymptotically inefficient; it not a sufficiently good approximation to the true likelihood, and it does not account for the distribution $p(\hat{N}(S_1), \hat{N}(S_2)|h(S_1), h(S_2))$. Unfortunately, this conditional distribution of the streaming estimates given the unaugmented sketches is intractable to compute. Instead, we approximate it with a tractable distribution to form what we refer to as a full pseudo-likelihood.

We use a bivariate normal as the tractable distribution. This choice is well-motivated as one would expect that the central limit theorem to yield this as the limit distribution. The mean of this distribution is given by the streaming cardinality estimates. An approximate covariance matrix $\Sigma(S_1, S_2)$ for the streaming cardinality estimates is derived in section 10. To simplify calculations, one may also use other choices such as a diagonal matrix.

Mathematically, the likelihood is given by

$$\begin{aligned}\ell(x_1, x_2, x_{1\cap 2}; S_1, S_2) &= \ell_0(x_1, x_2, x_{1\cap 2}; S_1, S_2)\\&+ \log \phi\left(n_A, n_B | \hat{N}(S_1), \hat{N}(S_2), \Sigma(S_1, S_2)\right)\end{aligned}$$

where ϕ is a bivariate normal density and $\Sigma(S_1, S_2)$ is an approximate covariance matrix for the streaming cardinality estimates. The resulting pseudo-likelihood cardinality estimators for the intersection and union are

$$\hat{N}_{pseudo}(S_1 \cap S_2) = \arg\max_{x_{1\cap 2}} \max_{x_1, x_2} \ell(x_1, x_2, x_{1\cap 2}) \quad (6)$$

$$\hat{N}_{pseudo}(S_1 \cup S_2) = \arg\max_{x_1+x_2-x_{1\cap 2}} \max_{x_1, x_2} \ell(x_1, x_2, x_{1\cap 2}). \quad (7)$$

Note that this also allows the estimates of $|A_1|$ and $|A_2|$ to be improved when there is substantial overlap between the sets. In particular, when $A_1 = A_2$, the estimate of $|A_1|$ will be an average of the estimates $\hat{N}(S_1)$ and $\hat{N}(S_2)$.

The variance estimates for the pseudo-likelihood cardinality estimators are

$$\hat{\mathrm{Var}}(\hat{N}_{pseudo}(S_1 \cap S_2)) = D^2_{x_{1\cap 2}} \ell \Big|_{\hat{N}_{pseudo}} \quad (8)$$

$$\hat{\mathrm{Var}}(\hat{N}_{pseudo}(S_1 \cup S_2)) = D^2_{x_1+x_2-x_{1\cap 2}} \ell \Big|_{\hat{N}_{pseudo}} \quad (9)$$

where D denotes the directional derivative and $\hat{N}_{pseudo}$ is the maximizer of the pseudo-likelihood.

8.4 Optimality of pseudo-likelihood

Since the pseudo-likelihood is a surrogate for the true likelihood, asymptotic efficiency results for the maximum likelihood estimator do not directly apply. However, we conjecture that the pseudo-likelihood estimator is asymptotically equivalent to the maximum likelihood estimator. This result would not be surprising since one would expect that the streaming cardinality estimates converge to a bivariate normal distribution by the central limit theorem. Furthermore, by asymptotic results for M-estimators [24], such as the MLE, if the quadratic approximation of the pseudo-log-likelihood and log-likelihood converge to the same limit, their asymptotic distributions are the same. Since we estimate the conditional covariance for the streaming estimates given the basic sketch, we have approximated all the quadratic terms of the log-likelihood.

9. RE-WEIGHTED ESTIMATORS

We provide a second class of union and intersection cardinality estimators that are easier to implement than the pseudo-likelihood based estimators. They are also easier generalize to sketches other than MinCount and to multiset operations. These estimators are formed by taking the weighted average of several unbiased or consistent estimators of the cardinality. We derive how to optimally combine these component estimators to obtain an estimator that performs nearly identically to the pseudo-likelihood based estimator. For the estimators presented in this paper, we use component estimators where each estimator is a multiplicative scaling of a single streaming cardinality estimate.

We first describe the optimal procedure for combining multiple component estimators. Then we propose a set of component estimators and derive the variances and covariances needed to combine them. The procedure to optimally combine estimators is given by the following lemma.

LEMMA 3 (OPTIMAL WEIGHTING). *Given unbiased (or consistent) estimators $\hat{N}_1, ..., \hat{N}_m$ with non-singular covariance matrix Σ, the optimal weights that sum to 1 and minimize variance are given by*

$$w_{opt} \propto \Sigma^{-1} 1_m \quad (10)$$

where 1_m is the m-vector of all ones. The resulting estimator is unbiased (or consistent) and has variance $(1_m^T \Sigma^{-1} 1_m)^{-1}$.

PROOF. Since all the estimators are unbiased (or consistent), the re-weighted estimator is also unbiased (or consistent). Given unnormalized weights v, the variance of the estimator $v^T \hat{N}/v^T 1$ is $\mathrm{Var}(v^T \hat{N}/v^T 1) = v^T \Sigma v / v^T 1 1^T v$. Minimizing this Rayleigh quotient give the optimal weights. □

Typically, the covariance Σ is not known, so a plug-in estimate of Σ is used to generate the weights. An alternate weighting is to treat the estimators as being independent and simply use the diagonal of the covariance. In this case, the simple re-weighted estimator is

$$\hat{N}_{simple} = \sum_{i=1}^{m} \frac{z\hat{N}_i(S)}{\mathrm{Var}(\hat{N}_i(S))} \tag{11}$$

where z is a normalization constant that ensures weights sum to 1. Although this does not provide guarantees on the improvement, we find that this performs nearly as well as the optimal weighting.

We now consider the two ingredients needed to define a re-weighted estimator: a set of consistent estimators that form the components of the combined estimator and the covariance of these estimators to determine the weights.

9.1 Component estimators

The idea for defining the component estimators is to have maximally uncorrelated estimators. We minimize correlation by using component estimators which make use of only one of the streaming cardinality estimate in each component. This gives that the resulting weighted estimator is expressed as a linear combination of streaming cardinality estimates. These component estimators are shown to be unbiased or consistent and, hence, suitable candidates for a re-weighted estimator. The approximate variance of the estimators is computed in order to allow for computation of the weights.

To derive the component estimators, note that both the cardinality of the union and the intersection may be decomposed into a ratio times the cardinality of one of the sets in the operation.

$$|A_1 \cup A_2| = \frac{|A_1 \cup A_2|}{|A_i|}|A_i|, \quad |A_1 \cap A_2| = \frac{|A_1 \cap A_2|}{|A_i|}|A_i|.$$

Define $\alpha_i = |A_1 \cap A_2|/|A_i|$ and $\beta_i = |A_1 \cup A_2|/|A_i|$. Simple estimators for α_i and β_i are

$$\begin{aligned} \hat{\alpha}_i &= |h(S_1 \cap_0 S_2)|/|h(S_i, \tau_{min})| \\ \hat{\beta}_i &= |h(S_1 \cup_0 S_2)|/|h(S_i, \tau_{min})| \end{aligned} \tag{12}$$

These give a consistent, plug-in estimator for the cardinality of an intersection and union as shown in the following lemma, proven in the long version of the paper, and theorem.

LEMMA 4. *The estimator $\hat{\alpha}_i$ is an unbiased estimator of $|A_1 \cap A_2|/|A_i|$ conditional on $|h(S_i, \tau_{min})|$ being non-zero. It is consistent under the conditions in section 7. Similarly, $\hat{\beta}_i$ is a consistent estimator of $|A_1 \cup A_2|/|A_i|$ under the conditions in section 7.*

THEOREM 5. *The estimators*

$$\hat{N}_i(S_1 \cap S_2) = \hat{\alpha}_i\hat{N}(S_i), \quad \hat{N}_i(S_1 \cup S_2) = \hat{\beta}_i\hat{N}(S_i)$$

are consistent estimators of the $|A_1 \cap A_2|, |A_1 \cup A_2|$ under the conditions given in section 7.

PROOF. This immediately follows from Slutzky's theorem since $\hat{\alpha}_i, \hat{\beta}_i, \hat{N}(S_i)$ are consistent. □

Another possible choice for component estimators is to pretend that all items in A_2 appear after A_1 in the stream. A more accurate streaming cardinality estimate can be calculated by not updating the threshold. However, the resulting re-weighted estimator is suboptimal when A_1 and A_2 are disjoint and equal in size. It is equivalent to averaging the suboptimal improved union estimate with optimal sum of streaming estimates.

9.2 Component variances

The variance of each component estimator is required to apply the simple re-weighting scheme. This is approximated by decomposing the component estimator so that the multipliers and cardinality estimates can be treated as essentially independent. Write $\sqrt{k_i}(\hat{\alpha}_i\hat{N}(S_i) - n_i\alpha_i) = \sqrt{k_i}\hat{\alpha}_i(\hat{N}(S_i) - n_i) + \sqrt{k_i}(\alpha_i - \alpha_i)n_i$. Assuming the central limit theorem applies to $\hat{N}(S_i)$, Slutzky's lemma and the central limit theorem imply that the quantity converges in distribution to a normal distribution with variance less than or equal to $\alpha_i^2 k\mathrm{Var}(\hat{N}(S_i)) + n_i^2 k\mathrm{Var}(\hat{\alpha}_i)$. The missing term measuring $Cov(\hat{N}(S_i), \hat{\alpha}_i)$ is excluded as the terms are negatively correlated. The same argument applies to $\hat{\beta}_i$.

In this case, the expectation and variance of $\hat{N}(S_i)$ are already given in [23] and the expectations are known or approximated by lemma 4. The only remaining quantities to compute are $\mathrm{Var}(\hat{\alpha}_i)$ and $\mathrm{Var}(\hat{\beta}_i)$. The variances of $\hat{\alpha}_i, \hat{\beta}_i$ may be approximated using the following formulas.

$$\begin{aligned} \mathrm{Var}\left(\hat{\alpha}_i\right) &\approx \frac{\alpha_i(1-\alpha_i)}{|h(S_i, \tau_{min})|} \\ \mathrm{Var}\left(\hat{\beta}_i\right) &\approx \frac{1-p_i}{p_i^2|h(S_i, \tau_{min})|} = \frac{\beta_i(\beta_i - 1)}{|h(S_i, \tau_{min})|}. \end{aligned} \tag{13}$$

These approximate variances may be used in the simple re-weighting scheme given in equation 11.

$$\hat{\mathrm{Var}}(\hat{N}_i(S_1 \cap S_2)) \approx \hat{N}_i(S_1 \cap S_2)^2 \left(\frac{(1-\hat{\alpha}_i)}{\hat{\alpha}_i|h(S_i, \tau_{min})|} + \frac{1}{2k_i}\right)$$

$$\hat{\mathrm{Var}}(\hat{N}_i(S_1 \cup S_2))) = \hat{N}_i(S_1 \cup S_2)^2 \left(\frac{(\hat{\beta}_i - 1)}{\hat{\beta}_i|h(S_i, \tau_{min})|} + \frac{1}{2k_i}\right)$$

The component estimators may be compared to the improved intersection and union estimators given in section 5. Since $\alpha_i|h(S_i, \tau_{min})| \approx |h(S_1 \cap_0 S_2)|$ and $\beta_i|h(S_i, \tau_{min})| \approx |h(S_1 \cup_0 S_2)|$, the variance of the improved intersection estimator is approximately $|A_1 \cap A_2|^2/\alpha_i|h(S_i, \tau_{min})|$. The variance of the component intersection estimator, and similarly the component union estimator, can be approximated by

$$\begin{aligned} \mathrm{Var}(\hat{N}_i(S_1 \cap S_2)) &\approx \mathrm{Var}(\hat{N}_{improved}(S_1 \cap S_2)) \\ &\quad - |A_1 \cap A_2|^2 \left(\frac{1}{|h(S_i, \tau_{min})|} - \frac{1}{2k_i}\right) \end{aligned} \tag{14}$$

$$\begin{aligned} \mathrm{Var}(\hat{N}_i(S_1 \cup S_2)) &\approx \mathrm{Var}(\hat{N}_{improved}(S_1 \cup S_2)) \\ &\quad - |A_1 \cup A_2|^2 \left(\frac{2-\beta_i}{|h(S_i, \tau_{min})|} - \frac{1}{2k_i}\right) \end{aligned} \tag{15}$$

For the intersection estimates, since $|h(S_i, \tau_{min})| \leq k_i$, both component estimators are better than the than the basic improved estimator even though they exploit the accuracy of only one of the streaming cardinality estimates $\hat{N}(S_i)$. Furthermore, the component with the smaller $|h(S_i, \tau_{min})|$ has the greater improvement over the basic improved estimator. A surprising consequence of this approximation is that this component should represent an improvement over the basic improved estimator even when the streaming cardinality estimate is not even used. In that case, the $1/2k_i$ term is replaced by $1/k_i$ in the improvement term. We believe the reason is that the basic improved estimator loses information about the size of the smaller set which constrains the size of the intersection. For the union estimates, the improvement depends on the multiplier β_i. If $\beta_i \leq 3/2$, then the component estimator beats the basic improved estimator.

10. COVARIANCE ANALYSIS

Computing the covariance of the streaming estimates is a necessary step for obtaining the full quadratic approximation to the log-likelihood used by the pseudo-likelihood estimator. It is also needed to compute the weights for the full re-weighted estimator. We present the main ideas and results for calculating the covariance and leave the detailed calculations to the long version of the paper.

In the streaming formulation of MinCount, the remaining area process $\mathcal{R}(S_t) = \tau_t$ implicitly depends on the order π in which elements arrive. This dependency is difficult to analyze. We consider an alternative formulation that expands and extends the proof ideas in [10] and is based on the ranks of the hash values. The Streaming MinCount estimator can be expressed as

$$\hat{N}(S) = k + \sum_{i=k+1}^{n} \frac{Z_i}{U_{(i)}} \tag{16}$$

where $U_{(i)}$ is the i^{th} smallest hash value and $Z_i = 1$ if and only if the estimator was incremented by $1/U_{(i)}$. Equivalently, $Z_i = 1$ if and only if fewer than k of the $i-1$ smallest hash values appear before the i^{th} smallest value. Mathematically, $Z_i = 1(|\{j < i : \pi_j < \pi_i\}| < k)$ where π_i is the position of $U_{(i)}$ in the stream.

When considering multiple sketches, denote the a^{th} streaming estimate by $\hat{N}(S_a) = k_a + \sum_{i=k_a+1}^{n_{tot}} Z_i^{(a)}/U_{(i)}$ where $U_{(i)}$ are the order statistics for all the hash values for $A_1 \cup A_2$. This rank formulation is the key to the analysis in this section. It separates the analysis into three independent components: the order statistics for the hash values $U_{(\cdot)}$, the random orderings $\pi^{(a)}$, and an indicator $Y_i^{(a)}$ denoting if $U_{(i)}$ belongs to stream a. By the independence of Z and U and bilinearity of the covariance operator, the covariance $Cov(\hat{N}(S_1), \hat{N}(S_2))$ of two streaming cardinality estimates can then decomposed into the cross terms

$$\begin{aligned} Cov\left(\frac{Z_i^{(1)}}{U_{(i)}}, \frac{Z_i^{(1)}}{U_{(i)}}\right) &= Cov\left(Z_i^{(1)}, Z_j^{(2)}\right) \mathbb{E}\left(\frac{1}{U_{(i)}} \frac{1}{U_{(j)}}\right) \\ &\quad + \mathbb{E}Z_i^{(1)} \mathbb{E}Z_j^{(2)} Cov\left(\frac{1}{U_{(i)}}, \frac{1}{U_{(j)}}\right) \end{aligned} \tag{17}$$

Of these terms, $Cov\left(Z_i^{(1)}, Z_j^{(2)}\right)$ depends on the order of elements in both streams. In particular, it depends on whether the common elements $A_1 \cap A_2$ appear in the same order or different orders in the streams. We consider two cases: the orders are the same and the orders are independent random permutations.

The covariances of the cardinality estimates are given by

$$\Sigma_{independent} \approx \frac{n_{tot}^2 k_1 k_2}{\kappa_{max}^2} \frac{q_0}{p_1 p_2} \left(\frac{1}{2\kappa_{min}} - \frac{1}{6\kappa_{max}}\right) \tag{18}$$

$$\Sigma_{identical} \approx \Sigma_{independent} + q_0 \left(\frac{1}{2} - \frac{q_0}{3p_1p_2}\right) \frac{n_{tot}^2 \kappa_{min}}{\kappa_{max}^2}$$

where $\kappa_{min} = \left\lfloor \min\left\{\frac{k_1}{p_1}, \frac{k_2}{p_2}\right\}\right\rfloor$ and $\kappa_{max} = \left\lfloor \max\left\{\frac{k_1}{p_1}, \frac{k_2}{p_2}\right\}\right\rfloor$.

The covariance $\Sigma_{independent}$ is of particular interest when $q_0 = 1$ so that the sets $A_1 = A_2$. The best estimate when the hash function is fixed takes the average of Streaming MinCount estimates over all possible orderings of the same set. This limit estimator reduces the variance to $\Sigma_{independent} = \frac{n^2}{3k}$. By contrast, when applied to a single stream the Streaming MinCount estimator has variance $\frac{n^2}{2k}$. Thus, even in the case where every cardinality estimate is for exactly the same set, the estimates can be improved by combining them. This result is borne out in figure 2 as the basic MinCount sketch requires 3 times the space as the averaged Streaming MinCount sketches when $q_0 = 1$ and the sets are all the same.

10.1 Conditional covariance

For the pseudo-likelihood described in section 8.2, the required normal approximation is for the conditional distribution $p(\hat{N}(S_1), \hat{N}(S_2)|\tau)$ rather than the unconditional covariance. Under a multivariate normal approximation, this conditional distribution is easy to approximate. Let $\hat{N}_{mincount}$ be the non-streaming MinCount estimator that is based only on the offset τ. Assume that the joint distribution of $\hat{N}(S_1)$, $\hat{N}(S_2)$, $\hat{N}_{mincount}(S_1)$, and $\hat{N}_{mincount}(S_2)$ is multivariate normal. The covariance may be denoted in block form by

$$\begin{pmatrix} \Sigma_{ss} & \Sigma_{sm} \\ \Sigma_{sm}^T & \Sigma_{mm} \end{pmatrix} \tag{19}$$

where s, m denote the streaming and non-streaming MinCount estimators.

Since the streaming estimate is optimal, it cannot be improved upon by conditioning on the threshold. In particular, under a re-weighting scheme a linear combination of the streaming and non-streaming estimators does not improve the variance. Hence, the non-streaming estimator is approximately equal to the streaming estimator plus uncorrelated noise, $\Sigma_{sm} \approx \Sigma_{ss}$. The last component is the covariance of the non-streaming MinCount estimates which is $Cov(\hat{N}_{mincount}(S_1), \hat{N}_{mincount}(S_2)) \approx q_0 \text{Var}(\hat{N}_{mincount}(S_\ell))$ where ℓ is the sketch with the higher threshold. Under a multivariate normality approximation where $\hat{n}_i$ denotes the current estimate for n_i, the conditional variance and mean $\mu_{ss,i|\tau} = \mathbb{E}(\hat{N}(S_i)|\tau)$ are given by

$$\Sigma_{ss|\tau} \approx \Sigma_{ss} - \Sigma_{ss}\Sigma_{mm}^{-1}\Sigma_{ss}$$

$$\mu_{ss|\tau} \approx \begin{pmatrix} \hat{N}(S_1) \\ \hat{N}(S_2) \end{pmatrix} - \Sigma_{ss}\Sigma_{mm}^{-1} \begin{pmatrix} \hat{N}_{mincount}(S_1) - \hat{n}_1 \\ \hat{N}_{mincount}(S_2) - \hat{n}_2 \end{pmatrix}.$$

10.2 Multiplier covariance

The covariance of the multipliers may be approximated by first conditioning on the value in the denominator. This gives rough estimates

$$Cov(\hat{\alpha}_1, \hat{\alpha}_2) \approx \frac{\rho_2(1-\rho_2)}{k_1} \tag{20}$$

$$Cov(\hat{\beta}_1, \hat{\beta}_2) \approx \frac{\rho_2(1-\rho_2)}{k_1} - \frac{\rho_2}{k_1}\left(\frac{p_1}{p_2} - \frac{q_0^2}{p_2^2}\right) \tag{21}$$

where $\rho_2 = q_2/p_2$ and $\tau_1 < \tau_2$. The multiplier covariances may be incorporated by the same argument as in section 9.2.

11. MULTI-SET OPERATIONS

Oftentimes the query of interest merges many sketches together rather than pairs. For example, if the data is processed on d mappers in a map-reduce framework, d sketches must be merged to estimate the cardinality.

Merging multiple estimators using re-weighting is simple. The component estimators are slightly modified, and the re-weighting is virtually unchanged since covariance calculations are pairwise calculations. The multipliers α_i, β_i estimate the size of a set relative to the intersection or union of all the sets. The only difference is that the minimum threshold $\tau_{min} = \min_i\{\tau_i\}$ and the number of relevant hashes $|h(\cap_0 S_i)|, |h(\cup_0 S_i)|$ are taken over all the sets rather than a pair.

Generalizing pseudo-likelihood based methods is more difficult. Although the same technique of ordering the thresholds to compute the likelihood may be used, the likelihood itself contains exponentially many parameters when there are more than two sketches. The intersection of any subset of the sets $\{A_i\}$ is a parameter. This makes the optimization problem difficult, and the finite sample performance of likelihood based methods is not well understood.

Another strategy is to exploit that pairwise intersections and unions are closed operations. The result of intersecting or taking the union of two sketches results in another sketch. These sketches may be further merged. For the profile-likelihood, which does not utilize any variance or covariance information for the estimates $\hat{N}(S_1)$ and $\hat{N}(S_2)$, the merge operation is straightforward. For other methods, a surrogate for the covariances of the merged estimates needed for computation of the cardinality estimate.

12. EMPIRICAL RESULTS

The new estimators were evaluated on both real and simulated data. The results on real data exactly match the theory and empirically demonstrate that there is no difference between applying the methods on real or simulated data. We examine the behavior of the estimators in a variety of cases. In all cases, the pseudo-likelihood and re-weighted estimators that exploit the streaming cardinality estimates had the best performance and performed similarly. To assess the performance of the estimators, we computed the Relative Efficiency (RE) of the estimators and the relative size required. The RE is defined as $RE(\hat{N}^{(1)}, \hat{N}^{(2)}) = \text{Var}(\hat{N}^{(2)})/\text{Var}(\hat{N}^{(1)})$. Since the variance of an estimator typically scales as $1/m$ where m is the number of samples used in the estimator, the RE measures how much more data $\hat{N}^{(2)}$ needs to achieve the same accuracy as $\hat{N}^{(1)}$. An estimator with RE of 1/2 requires twice as many samples to match the variance of the baseline estimator. The relative size is the space required to achieve the same error and is $1/RE$. Figure 1 show that, analogous to the improvement of streaming estimators in the single stream problem, the new methods require half the space of the improved union estimator. They achieve even greater reductions in space when compared to basic MinCount estimator. When intersection sizes are small, the new estimators require less than $1/15^{th}$ of the space of inclusion-exclusion based estimators. Compared to the basic improved estimator, the merged intersection estimates require half the space when exploiting the streaming sketches and one set is nearly a subset of the other.

In simulations for pairwise unions and intersections, we let $|A_1| = 2^{20}$ and considered cases with both balanced, $|A_2| = |A_1|$, or unbalanced, $|A_2| = |A_1|/4$, cardinalities. The size of the intersection was allowed to vary. Unless otherwise noted, the order of common elements in the streams was the less favorable order for our methods; the elements appear in the same order. We tried different sketch sizes in our simulations but only present the results where the size $k = 1024$ as there was no material difference when $k = 256$ or 4196.

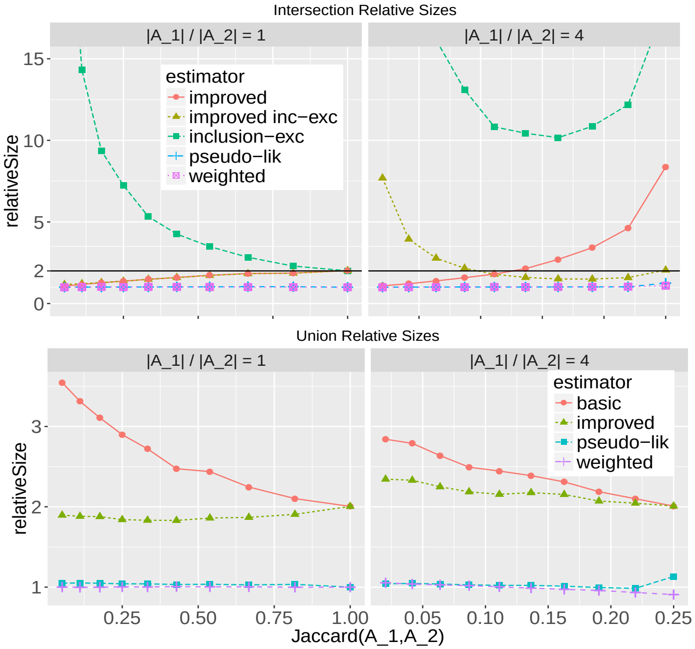

Figure 1: These figures show the sketch size needed by different estimators to achieve a given error relative to the profile-likelihood estimator. For intersections on sets with significantly different cardinalities, the new likelihood and re-weighted estimators require far less space.

For real data, we used the MSNBC dataset from the UCI machine learning repository [20] to validate our simulation results. This dataset consists of 1 million users' browsing history of 17 web page categories. We consider the problem of estimating how many users visited a pair of categories using only the 17 sketches counting unique visits to each category. For comparison, we simulated data using parameters that match the real set and intersection cardinalities. For each set of parameters 10,000 sketches were generated with independent random hashes. Figure 3 shows, as expected, that the estimated cardinalities using the real data have the same distribution as the simulated data. A p-value was computed for each set of parameters testing if the RRMSE for real and simulated data are different. Under the null hypothesis that there is no difference in distributions, the p-values are uniformly distributed. A Kolmogorov-Smirnov test yields a p-value > 0.6, so there is statistically no evidence in the test that the simulated data behaves differently from the real data.

There were only small difference in performance between the simple re-weighted estimator and the optimally weighted estimator except when the cardinalities were unbalanced and the smaller set was nearly a subset of the larger one. This difference only observed for intersections. For unions, the simple re-weighting performed marginally better than the optimal re-weighting, possibly due to additional error in estimating the covariance of the component estimators. We explain this asymmetry by noting that in the case where

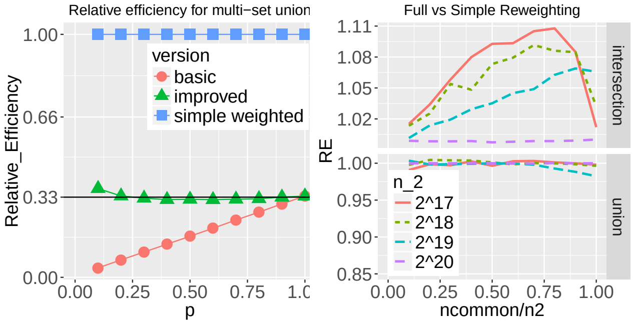

Figure 2: The left figure shows the RE with respect to the simple re-weighted estimator when taking the union of 100 random subsets A_i drawn with probability p from A_{tot}. The right figure shows the RE of the full re-weighted estimator to the simple re-weighted estimator. Using the full covariance structure improves intersection estimates but error in the covariance estimate hurts unions very slightly.

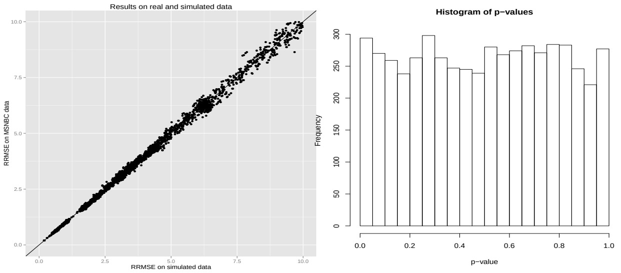

Figure 3: The left figure shows the RRMSE when estimators are run on real data versus simulated data. There is an exact correspondence, The right figure shows the distribution of p-values testing if there is a difference between the real versus simulated data. The distribution fits the null uniform distribution.

$A_2 \subset A_1$ and the order elements appear is the same, every item in A_2 that updates sketch S_1 also updates S_2. Thus, S_1 contains almost no information about $|A_2|$ that is not already in S_2, and the simple re-weighting adds noise by giving weight to component $\hat{N}_1(S_1 \cap_0 S_2)$. The reverse for unions is not true; S_2 contains useful information not in S_1 for estimating $|A_1|$. We note that the covariance estimate for the component estimators is imperfect as it does not simply select the smaller set when A_2 is a proper subset of A_1. However, the full re-weighted estimator empirically has relative efficiency $\geq 95\%$ of an optimal re-weighting, so any improvement would be small.

To simulate the distributed map-reduce setting, the simple re-weighted estimator was compared to the basic and improved MinCount estimators when taking the union of 100 sketches. Each sketch was generated by sub-sampling with probability p from a set of 10^6 elements. The sampled items were then randomly permuted before computing the sketch. Figure 2 shows the simple re-weighted estimator has high relative efficiency compared to the naive union estimator, especially when the size of the pairwise intersections is small. Likewise, the improved estimator has high relative efficiency compared to the basic estimator in those cases. Here, the re-weighted estimator requires 1/3 the space of the improved estimator rather than 1/2 due to the accuracy gains from averaging over independent permutations.

13. DISCUSSION

Although the calculations presented are specialized for the MinCount sketch, the techniques apply to other sketches as well such as the Flajolet-Martin (FM) sketch used in HyperLogLog or the Discrete Max-Count sketch in [23]. In particular, the simple re-weighted estimator can be easily derived for the FM sketch. The only ingredients needed are estimates of the Jaccard similarities $J(A_1 \cap A_2, A_i)$ and $J(A_i, A_1 \cup A_2)$ and their variances. The FM sketch, however, does not lead to a closed intersection operation that allows for further merging.

Another possible generalization is to estimate set differences. For the likelihood based methods, this is exactly the same as estimating the intersection since it is simply another re-parameterization. For the re-weighting methods, the multipliers change but the analysis remains the same.

For the methods described in this paper, cardinality estimation for the union or intersection only requires constant space. There are a finite number of sufficient statistics and that these can be tabulated with constant memory if the hash values in each sketch can be accessed in sorted order. If a merged sketch and not just a cardinality estimate is required, a closed union operation may result in a larger sketch where the size is proportional to the number of sets in the union. The sketch can be truncated to have k hash values to reduce the space requirements while cardinality estimation is still performed with all the hash values.

One unexplored area is determining the optimal order of pairwise merges. If many sketches cannot be merged simultaneously, the desired cardinalities may still be computed using a sequence pairwise unions and intersections that exploit the closed union and set operations.

13.1 Running times

For all the improved methods in this paper, computing the sufficient statistic requires $O(km)$ time where m is the number of sketches. If the hash values are stored in sorted order, this is reduced to $O(k)$. For the naive estimators, the running time is $O(mk \log k)$ if a heap is used to select the k values. For the pairwise likelihood based methods, the pseudo-likelihood function is from an exponential family and hence, log-concave. Evaluating the gradient and Hessian take constant time, so the running time is proportional to the number of iterations needed to solve this concave optimization problem. For the re-weighted component estimators, if the estimated covariance matrix is diagonal, computing the re-weighted estimator takes $O(m)$ time. If a full covariance matrix is used, then inverting the matrix can take $O(m^3)$ time.

14. CONCLUSION

This paper presents and analyzes two new classes of methods for estimating cardinalities of intersections and unions from sketches. These methods are applied to Streaming MinCount sketches, and variance estimates are derived as well. All the methods theoretically and empirically outperform existing methods for estimating unions and intersections. The new methods also lead to mergeable summaries under intersection and union operations. This allows both intersections and unions on sketches to be chained together while existing methods only allow unions to be chained. Extensions to sketches other than the MinCount sketch are also discussed.

The pseudo-likelihood based estimators are derived as approximations to the asymptotically optimal maximum likelihood estimator. We conjecture that the full pseudo-likelihood estimator is asymptotically equivalent to the maximum likelihood estimator. The re-weighted component estimators are derived as near optimally weighted linear combinations of the streaming estimates. Empirically, the near optimally re-weighted component estimator matches the performance of the full maximum pseudo-likelihood estimator.

The derived estimators are useful in a variety of different settings. The re-weighted estimators can be easily generalized to handle multi-set unions rather than just pairwise unions. The simple re-weighted estimators can be easily generalized to other sketches. The profile-likelihood can be used for sequences of pairwise merges.

The theoretical analysis also allows us to separate and identify the information contained in different parts of the sketch. Since the full re-weighted and the conjectured optimal pseudo-likelihood methods perform nearly identically and since the multipliers for component estimators can be treated as nearly independent from the streaming cardinality estimates, we see that the stored hash values encode information about the proportional sizes of the sets relative to their union and intersection while the streaming cardinality estimate contain information about the absolute size. Section 10 shows that each streaming estimate contains information about the order of elements in the stream since the streaming cardinality variance can be decomposed into an irreducible component that depends on the order statistics for the hash values and a reducible component that depends on the ordering of elements in the stream. In the case of identical sets, averaging streaming estimates over all possible orderings variance reduces the variance to $n^2/3k$. In section 9.2, the single non-streaming component estimator based on the smaller set is shown to dominate the improved MinCount estimator that throws away the larger threshold. All these relevant pieces of information useful for estimation are contained in the sufficient statistics given in section 8.1 which the pseudo-likelihood estimator makes full use of.

Together these contributions advance the methodology, theory, and understanding for the approximate distinct counting problem.

15. REFERENCES

[1] P. K. Agarwal, G. Cormode, Z. Huang, J. M. Phillips, Z. Wei, and K. Yi. Mergeable summaries. *ACM Transactions on Database Systems*, 38(4):26, 2013.

[2] Z. Bar-Yossef, T. Jayram, R. Kumar, D. Sivakumar, and L. Trevisan. Counting distinct elements in a data stream. In *Randomization and Approximation Techniques in Computer Science*, pages 1–10. Springer, 2002.

[3] J. Besag. Statistical analysis of non-lattice data. *The Statistician*, pages 179–195, 1975.

[4] K. Beyer, R. Gemulla, P. J. Haas, B. Reinwald, and Y. Sismanis. Distinct-value synopses for multiset operations. *Communications of the ACM*, 52(10):87–95, 2009.

[5] P. Boldi, M. Rosa, and S. Vigna. Hyperanf: Approximating the neighbourhood function of very large graphs on a budget. In *WWW*, 2011.

[6] P. Chassaing and L. Gerin. Efficient estimation of the cardinality of large data sets. *arXiv preprint math/0701347*, 2011.

[7] A. Chen, J. Cao, L. Shepp, and T. Nguyen. Distinct counting with a self-learning bitmap. *Journal of the American Statistical Association*, 106(495):879–890, 2011.

[8] P. Clifford and I. Cosma. A statistical analysis of probabilistic counting algorithms. *Scandinavian Journal of Statistics*, 39(1):1–14, 2012.

[9] E. Cohen. Size-estimation framework with applications to transitive closure and reachability. *Journal of Computer and System Sciences*, 55(3):441–453, 1997.

[10] E. Cohen. All-distances sketches, revisited: HIP estimators for massive graphs analysis. In *PODS*, 2014.

[11] E. Cohen and H. Kaplan. Summarizing data using bottom-k sketches. In *PODC*, 2007.

[12] G. Cormode, M. Datar, P. Indyk, and S. Muthukrishnan. Comparing data streams using hamming norms (how to zero in). *IEEE Trans. Knowledge and Data Engineering*, 15(3):529–540, 2003.

[13] T. Dasu, T. Johnson, S. Muthukrishnan, and V. Shkapenyuk. Mining database structure; or, how to build a data quality browser. In *SIGMOD*, 2002.

[14] M. Durand and P. Flajolet. Loglog counting of large cardinalities. In *Algorithms-ESA 2003*, pages 605–617. Springer, 2003.

[15] C. Estan, G. Varghese, and M. Fisk. Bitmap algorithms for counting active flows on high speed links. In *Internet Measurement Conference*, 2003.

[16] P. Flajolet, É. Fusy, O. Gandouet, and F. Meunier. Hyperloglog: the analysis of a near-optimal cardinality estimation algorithm. In *AofA*, 2007.

[17] P. Flajolet and G. Nigel Martin. Probabilistic counting algorithms for data base applications. *Journal of computer and system sciences*, 31(2):182–209, 1985.

[18] F. Giroire. Order statistics and estimating cardinalities of massive data sets. *Discrete Applied Mathematics*, 157(2):406–427, 2009.

[19] D. Kane, J. Nelson, and D. Woodruff. An optimal algorithm for the distinct elements problem. In *PODS*, 2010.

[20] M. Lichman. UCI machine learning repository, 2013.

[21] A. Metwally, D. Agrawal, and A. E. Abbadi. Why go logarithmic if we can go linear?: Towards effective distinct counting of search traffic. In *EDBT*, 2008.

[22] P. G. Selinger, M. M. Astrahan, D. D. Chamberlin, R. A. Lorie, and T. G. Price. Access path selection in a relational database management system. In *Proceedings of SIGMOD*, pages 23–34. ACM, 1979.

[23] D. Ting. Streamed approximate counting of distinct elements: beating optimal batch methods. In *KDD*, 2014.

[24] A. van der Vaart. *Asymptotic Statistics*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2000.

[25] K. Whang, B. Vander-Zanden, and H. Taylor. A linear-time probabilistic counting algorithm for database applications. *TODS*, 1990.