

Is Min-Wise Hashing Optimal for Summarizing Set Intersection?

Rasmus Pagh *
IT University of Copenhagen
pagh@itu.dk

Morten Stöckel *
IT University of Copenhagen
mstc@itu.dk

David P. Woodruff
IBM Research - Almaden
dpwoodru@us.ibm.com

ABSTRACT

Min-wise hashing is an important method for estimating the size of the intersection of sets, based on a succinct summary (a “min-hash”) independently computed for each set. One application is estimation of the number of data points that satisfy the conjunction of $m \geq 2$ simple predicates, where a min-hash is available for the set of points satisfying each predicate. This has applications in query optimization and for approximate computation of COUNT aggregates. In this paper we address the question: *How many bits is it necessary to allocate to each summary in order to get an estimate with $1 \pm \varepsilon$ relative error?* The state-of-the-art technique for minimizing the encoding size, for any desired estimation error, is b -bit min-wise hashing due to Li and König (Communications of the ACM, 2011). We give new lower and upper bounds:

- Using information complexity arguments, we show that b -bit min-wise hashing is *space optimal* for $m = 2$ predicates in the sense that the estimator’s variance is within a constant factor of the smallest possible among all summaries with the given space usage. But for conjunctions of $m > 2$ predicates we show that the performance of b -bit min-wise hashing (and more generally any method based on “ k -permutation” min-hash) deteriorates as m grows.
- We describe a new summary that nearly matches our lower bound for $m \geq 2$. It asymptotically outperforms all k -permutation schemes (by around a factor $\Omega(m/\log m)$), as well as methods based on subsampling (by a factor $\Omega(\log n_{\max})$, where n_{\max} is the maximum set size).

Categories and Subject Descriptors

F.2.0 [Analysis of Algorithms and Problem Complexity]: General

*Pagh and Stöckel are supported by the Danish National Research Foundation under the Sapere Aude program.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
PODS’14, June 22–27, 2014, Snowbird, UT, USA.
Copyright 2014 ACM 978-1-4503-2375-8/14/06 ...\$15.00.
<http://dx.doi.org/10.1145/2594538.2594554>.

1. INTRODUCTION

Many basic information processing problems can be expressed in terms of intersection sizes within a preprocessed collection of sets. For example, in databases and data analytics, aggregation queries often use a conjunction of several simple conditions such as “*How many sales occurred in June 2013, in Sweden, where the sold object is a car?*” In this paper we consider the problem of quickly *estimating* the size of the intersection of several sets, where a succinct precomputed summary of s bits is available for each set. Specifically, we answer the question:

How many bits is it necessary to allocate to each summary in order to get an estimate with $1 \pm \varepsilon$ relative error?

Note that we require the summaries to be *independently* computed, which for example prevents solutions based on precomputing all answers. This restriction is motivated by yielding scalable and flexible methods for estimating intersection sizes, with no need for a centralized data structure.

Motivation.

Estimates of intersection size can be used directly as part of algorithms with approximation guarantees, but are also useful for exact computation. For example, when evaluating conjunctive database queries the order in which intersections are computed can have a large impact on performance. Good estimates of intersection sizes allow a query optimizer to make a choice that is near-optimal. In other settings, estimates of intersection sizes can be used as a filter to skip parts of an exact computation that would not influence the output (e.g., we might only be interested in a particular sales figure if it exceeds some threshold).

In data warehouses it is common to perform extensive pre-computation of answer sets and summaries of simple queries, so that these can be combined to answer more complex queries quickly (see e.g. [26, 29]). At PODS 2011 Wei and Yi [31] showed that a number of different summaries of sets fulfilling a range condition can be efficiently extracted from augmented B-tree indexes. The number of I/Os for creating a summary of all data in a given range is close to the number of I/Os needed for reading a precomputed summary of the same size. That is, the efficiency is determined by the size of each summary, which motivates the question of how small a summary can be. Though Wei and Yi do not consider this explicitly, it is easy to see that (at least when efficient updates of data is not needed) their ideas apply to the kind of summaries, based on min-wise hashing, that we consider in the upper bounds of this paper.

1.1 Brief history

Motivated by document similarity problems encountered in AltaVista, Broder [5] pioneered algorithms for estimating set intersection sizes based on independently pre-computed “summaries” of sets. More specifically he presented a summary technique called “min-wise hashing” where, given summaries $k_{\min}(A)$ and $k_{\min}(B)$ of sets A and B , it is possible to compute a low-variance, (asymptotically) unbiased estimator of the *Jaccard similarity* $J(A, B) = |A \cap B|/|A \cup B|$. Assuming that $|A|$ and $|B|$ are known, an estimate of $J(A, B)$ with small relative error can be used to compute a good estimate of $|A \cap B|$, and vice versa. In fact, we state many of our results in terms of the ratio between the size of the intersection and the largest set, which is $\Theta(J)$.

Li and König [20] presented “ b -bit min-wise hashing”, a refinement of Broder’s approach that reduces the summary representation size by storing a vector of b -bit hash values of elements from $k_{\min}(X)$. Even though the resulting hash collisions introduce noise in the estimator, this can be compensated for by a small increase in the size of $k_{\min}(X)$, yielding a significantly smaller summary with the same variance. Specifically, with $b = 1$ and using s bits of space, the variance is $2(1 - J)/s$.¹ In order to get an estimation error of at most εJ with probability (say) $1/2$, by Chebychev’s inequality it suffices that $(1 - J)/s < (\varepsilon J)^2$, i.e., $s > (1 - J)/(\varepsilon J)^2$. It is not hard to show that the estimator is well-concentrated, and this bound is tight up to constant factors. Increasing the value of b (while keeping the space usage fixed) does not improve the variance.

1.2 Our contribution

First, we show that the variance of *any* estimator for Jaccard similarity based on summaries of s bits must be $\Omega(1/s)$ for fixed J between 0 and 1. More specifically, there exists a distribution of input sets such that with constant probability any such estimator makes an error of $\Omega(\sqrt{1/s})$ with constant probability. This means that b -bit min-wise hashing cannot be substantially improved when it comes to estimating intersection size (or Jaccard similarity) of two sets, except perhaps when J is asymptotically close to 0 or 1.

Second, we show that it *is* possible to improve existing estimators for the intersection size of $m = \omega(1)$ pre-processed sets. In fact, we show that estimators (such as b -bit min-wise hashing) that are based on many permutations are inherently less precise than their one-permutation counterpart when considering the intersection of many sets. We then show that a suitable approximate encoding of one-permutation min-wise hashing summaries is always competitive with b -bit min-wise hashing, while reducing the space required for accurately estimating the intersection size of many sets.

2. PREVIOUS WORK

Problem definition. Let S_1, S_2, \dots be sets of size $n_i = |S_i|$ where all $S_i \subseteq [u]$ and the largest set is $n_{\max} = \max n_i$. A query is a subset $I \subseteq \mathbb{N}$ of the set indices and the output to the query is the intersection size $|\bigcap_{i \in I} S_i|$. For ease of

¹The variance bound stated in [20] is more complex, since it deals with min-wise hashing based on permutations, which introduces correlations. By replacing this with full independence one arrives at the stated variance.

notation we assume that the query is $I = \{1, \dots, m\}$ and intersection size to estimate is then $t = |S_1 \cap \dots \cap S_m|$.

In this paper we consider estimators for the intersection size t . As previously noted we focus on the setting where the sets S_1, S_2, \dots, S_m are available for *individual* pre-processing. Storing only a small summary of each set, which requires not even approximate knowledge of t , we provide an estimator for the intersection size t . Note that in this model, we allow ourselves only to pre-process the sets independently of each other, i.e., intersection sizes or other information that rely on more than the set currently being pre-processed cannot be stored. See [13] for work on (exact) set intersection in the model where information about all sets can be used in the pre-processing phase.

For the applications, we seek to obtain bounds that are parameterized on the size of the summary required of each set as a function of largest set n_{\max} , the intersection size t , and the relative error ε . Further, let s denote the space in bits stored per set and k the number of permutations or number values taken from one permutation for k -permutation and one-permutation min-wise hashing respectively.

2.1 Lower bounds

Several well-known problems in communication complexity imply lower bounds for special cases of the set intersection problem:

In the **Index** problem Alice is given a subset of $\{1, \dots, n\}$, and Bob is given a set of size 1. The task is to determine whether the intersection size is 0 or 1. It is known that even for randomized protocols with error probability $1/3$, the one-way communication complexity of this problem is $\Omega(n)$ bits (see [19]). Informally, this shows that the cost of estimating set intersection grows with the ratio between the intersection size t and the size n_{\max} of the largest set.

In the **GapAnd** problem Alice and Bob are both given subsets of $\{1, \dots, n\}$, and the task is to determine if the intersection size is below $n/4 - \sqrt{n}$ or above $n/4 + \sqrt{n}$ (if it is in-between, any result is okay). This is a variant of the well-studied **GapHamming** problem, for which the randomized one-way communication complexity is $\Omega(n)$ bits [16, 32]. In fact, the randomized two-way communication complexity for this problem is also $\Omega(n)$ bits [8], though in our application of first preprocessing the sets in order to then answer queries, we will only need the result for one-way communication. Informally, this lower bound means that the cost of estimating set intersection is inversely proportional with the square of the relative error.

Informally, our lower bound shows that these results generalize and compose, such that the lower bound is the *product* of the cost due to **Index** and the cost due to **GapAnd**, each with constant error probability. That is, our lower bound will be $\Omega(n_{\max} \varepsilon^{-2}/t)$, which we can use to bound the variance of any estimator for Jaccard similarity. The intuitive idea behind the lower bound is to compose the two problems such that each “bit” of **GapAnd** is encoded as the result of an **Index** problem. Unlike typical arguments in information complexity, see, e.g., the PODS 2010 tutorial by Jayram [17], we instead measure the information a protocol reveals about *intermediate bits* in Claim 13, rather than about the inputs themselves. See the beginning of Section 3 for a more detailed intuition.

We note that using the output bits of multiple instances of one problem as the input bits to another problem was also

Method	Required space (bits)	Time
INCLUSION-EXCLUSION	$s \geq \varepsilon^{-2} (mn/t)^2 + \log n$	2^m
SUBSAMPLING	$s \geq \varepsilon^{-2} (n/t) \log m \log^2 n$	sm
<i>b</i> -BIT MIN-WISE HASHING*	$s \geq \varepsilon^{-2} (mn/t)$	sm
NEW UPPER BOUND	$s \leq \varepsilon^{-2} (n/t) \log(m) \log(n/\varepsilon t)$	sm
GENERAL LOWER BOUND	$s \geq \varepsilon^{-2} (n/t)$	-

Table 1: Comparison of estimators of intersection size t for relative error ε and constant error probability, with m sets of maximum size n . Bounds on the summary size s ignore constant factors. The subsampling bound assumes that no knowledge of t is available, and thus $\log n$ levels of subsampling are needed. *The bound for b -bit min-wise hashing assumes that the number of hash functions needed in the analysis of min-wise summaries is optimal, see appendix A.

used in [33], though not for our choice of problems, which are specific to and arguably very useful for one-way communication given the widespread usage of **Index** and **GapAnd** problems in proving encoding size or “sketching” lower bounds. We note that our problems may become trivial for 2-way communication, if e.g., one set has size n_{\max} while the other set has size 1, while the lower bounds for the problems considered in [33] are qualitatively different, remaining hard even for 2-way communication.

2.2 Min-wise hashing techniques

Min-wise hashing was first considered by Broder [5] as a technique for estimating the similarity of web pages. For completeness, below we define min-wise independence along with the standard algorithm to compute an unbiased estimator for resemblance.

DEFINITION 1 ([6, Eq. 4]). Let S_n be the set of all permutations on $[n]$. Then a family $\mathcal{F} \subseteq S_n$ is min-wise independent if for any set $X \subseteq [n]$ and any $x \in X$, when permutation $\pi \in \mathcal{F}$ is chosen at random we have

$$\Pr[\min \pi(X) = x] = 1/|X|.$$

In particular, for two sets $X, Y \subseteq [n]$ and a randomly chosen permutation $\pi \in \mathcal{F}$ we have

$$\Pr(\min \pi(X) = \min \pi(Y)) = J = \frac{|X \cap Y|}{|X \cup Y|}.$$

This can be used to compute an estimate of the Jaccard similarity. Specifically, given k independent min-wise permutations π_1, \dots, π_k then

$$\hat{J} = \frac{1}{k} \sum_{i=1}^k [\min \pi_i(X) = \min \pi_i(Y)]$$

is an unbiased estimator of J (where $[\alpha]$ is Iverson Notation for the event α) with variance $\text{Var}(\hat{J}) = J(1-J)/k$.

In both theory and practice it is often easier to use a hash function with a large range (e.g. size u^3) instead of a random permutation. The idea is that the probability of a collision among the elements of a given set should be negligible, meaning that with high probability the order of the hash values induces a random permutation on the set. We will thus use the (slightly misleading) term “one-permutation” to describe methods using a single hash value on each set element.

Min-wise summaries. For a given set X the k -permutation min-wise summary of size k is the vector

$$(\min \pi_1(X), \dots, \min \pi_k(X)).$$

The *one-permutation* min-wise summary (sometimes called bottom- k sketch) of size k for a permutation π is the set $k_{\min}(X) = \{\pi(x) \mid x \in X, \pi(x) < \tau\}$, where τ is the $k+1$ ’th largest permutation rank (hash value) of the elements in X . That is, intuitively k -permutation summaries store the single smallest value independently for each of k permutations, while one-permutation summaries store the k smallest values for one permutation. It is not hard to show that $|k_{\min}(X \cup Y) \cap k_{\min}(X) \cap k_{\min}(Y)|/k$ is a good estimator for J , where $k_{\min}(X \cup Y)$ can be computed from $k_{\min}(X)$ and $k_{\min}(Y)$. For k -permutation min-wise summaries, if π_1, \dots, π_k are independent min-wise permutations then

$$\frac{1}{k} \sum_{i=1}^k |\min \pi_i(X) \cap \min \pi_i(Y)|$$

is analogously an estimator for J .

2.3 Previous results on set intersection

For m sets S_1, \dots, S_m let the *generalized* Jaccard similarity be $J = |\cap_i S_i|/|\cup_i S_i|$. If we multiply an estimate of the generalized Jaccard similarity of several sets and an estimate of the size of the union of the sets, we obtain an estimate of the intersection size. Using existing summaries for distinct element estimation (also based on hashing, e.g. [18, 15]) we get that previous work on (generalized) Jaccard similarity implies results on intersection estimation [6, 7, 5, 9]. Recently, b -bit variations of min-wise hashing were proposed [22] but so far it is not clear how they can be used to estimate Jaccard similarity of more than three sets [21]. See Section 5.2 for further discussion.

The problem of computing aggregate functions (such as set intersection) over sets when hash functions are used to sample set elements has been widely studied [10, 12, 11]. In the general case of arbitrary aggregate functions, Cohen and Kaplan [11] characterizes for a given aggregate function f if an unbiased estimator for f with finite variance can be achieved using one- or k -permutation summaries. For the specific case of set intersection, *RC* (Rank Conditioning) estimators [10, 12] have been shown to provide an unbiased estimator based on both one- and k -permutation summaries and these can be extended to work with limited precision, analogous to b -bit min-wise hashing. Further, experimental work show that estimators based on one-permutation summaries outperform those based on k -permutation summaries [10] on the data sets used.

In contrast, this paper provides an explicit worst-case analysis of the space requirement needed to achieve ε error with error probability at most δ for set intersection us-

ing one-permutation summaries, where signatures (5.2) are used to shave off a logarithmic factor for the upper bound, making the bound close to being tight.

Table 1 shows the performance of different algorithms along with our estimator based on one-permutation min-wise hashing. The methods are compared by time/space used to achieve an (ε, δ) -estimate of the intersection size t of m sets of maximum size n for constant δ .

DEFINITION 2. *Let $z \in \mathbb{R}$ and let \hat{z} be a random variable. We say that \hat{z} is an (ε, δ) -estimate of z if $\Pr[|\hat{z} - z| \geq \varepsilon z] \leq \delta$. We use ε -estimate as shorthand for $(\varepsilon, 1/3)$ -estimate.*

The Jaccard estimator computed using k -permutation min-wise hashing, as described in Section 2.2, can trivially be used to estimate intersection when cardinality estimate of the union of the sets is given (by simply multiplying by the union estimate). However, there are instances of sets where J can be as low as $t/(t + m(n - t))$ for a “sunflower”, i.e., m sets of n elements that are disjoint except for the t intersection elements. Following from Chernoff bounds, such an instance requires to store $\frac{1}{J\varepsilon^2}$ elements to get an ε -estimation of J with constant probability. See Appendix A for a discussion of the bound for “ b -bit min-wise hashing” in Table 1.

In contrast, the one-permutation approach described in this paper stores $s \geq \frac{n}{t} \frac{\log m \log u}{\varepsilon^2}$ bits for m sets of maximum size n , while maintaining estimation time sm . Recent work investigated a *different* way of doing min-wise hashing using just one permutation [23], but this method seems to have the same problem as k -permutation min-wise hashing for the purpose of m -way set intersection. Intersection estimation can also be done by applying inclusion-exclusion to union size estimates of all subset unions of the m sets. To achieve error εt then by Chernoff bounds for sampling without replacement we need sample size $s > (\sum_i n_i / (\varepsilon t))^2$. As there are $2^m - 1$ estimates to do for m sets this yields time $2^m s$. Bloom filters [3] also support set intersection operations, and cardinality estimation, but to work well need the assumption that the sets have similar size. Therefore we will not discuss them further.

3. OUR RESULTS

We show a lower bound for the size of a summary for two-way set intersection by a reduction from one-way communication complexity. More specifically, any summary that allows a $(1 + \varepsilon)$ -approximation of the intersection size implies a one-way communication protocol for a problem we call **GapAndIndex**, which we think of as the composition of the **Index** and **GapAnd** communication problems. Namely, Alice has $r = \Theta(1/\varepsilon^2)$ d -bit strings x^1, \dots, x^r , while Bob has r indices $i_1, \dots, i_r \in [d]$, together with bits b_1, \dots, b_r . Bob’s goal is to decide if the input falls into one of the following cases, for a constant $C > 0$:

- a For at least $\frac{r}{4} + C\varepsilon r$ of the $j \in [r]$, we have $x_{i_j}^j \wedge b_j = 1$.
- b For at most $\frac{r}{4} - C\varepsilon r$ of the $j \in [r]$, we have $x_{i_j}^j \wedge b_j = 1$.

If neither case occurs, Bob’s output may be arbitrary (i.e., this is a *promise* problem).

A straightforward reduction shows that if you have an algorithm that can $(1 + \varepsilon)$ -approximate the set intersection size $|A \cap B|$ for sets A and B , then you can solve **GapAndIndex**

with the parameter d roughly equal to $|A|/t$ and $|B| \leq 4t$. Let the randomized communication complexity $R_{1/3}^{1-way}(\mathbf{f})$ of problem \mathbf{f} be the minimal communication cost (maximum message transcript) of any protocol computing \mathbf{f} with error probability at most $1/3$.

The crux of our lower bound argument is to show:

THEOREM 3. *For $r = \Theta(1/\varepsilon^2)$, $d = n_{max}/t$,*

$$R_{1/3}^{1-way}(\text{GapAndIndex}) = \Omega(dr).$$

In terms of the parameters of the original set intersection problem, the space lower bound is proportional to the ratio d between the largest set size and the intersection size multiplied by ε^{-2} . Since $d = \Theta(1/J)$ this is $\Omega(\varepsilon^{-2}/J)$, which is a lower bound on the space needed for a $1 \pm \varepsilon$ approximation of J . If we consider the problem of estimating J with *additive* error $\leq \varepsilon_{add}$ with probability $2/3$, observe that in this case $\varepsilon = \Theta(\varepsilon_{add}/J)$, so the lower bound becomes $\Omega(J/\varepsilon_{add}^2)$. Conversely, for fixed $J > 0$ and space usage s we get $\varepsilon_{add} = \Omega(1/\sqrt{s})$ with probability $1/3$ so the variance is $\Omega(1/s)$.

Our second result is a simple estimator for set intersection of an arbitrary number of sets, based on one-permutation min-wise hashing. The intuition behind our result is that when using k -permutation min-wise hashing, the probability of sampling intersection elements relies on the size of the union, while in contrast our one-permutation approach depends on the maximum set size, hence we save almost a factor of the number of input sets in terms of space. We show the following:

THEOREM 4. *Let sets $S_1, \dots, S_m \subseteq [u]$ be given and let $n_{max} = \max_i |S_i|$, $t = |S_1 \cap \dots \cap S_m|$ and k be the summary size $|k_{min}(S_i)|$. For $0 < \varepsilon < 1/4$, $0 < \delta < 1/\sqrt{k}$, consider the estimator*

$$X = \frac{\left| \bigcap_{i \in [m]} k_{min}(S_i) \right| n_{max}}{k}.$$

With probability at least $1 - \delta\sqrt{k}$:

$$t \in \begin{cases} [X/(1 + \varepsilon); X/(1 - \varepsilon)] & \text{if } X > 3n_{max} \log(2m/\delta)/k\varepsilon^2 \\ [0; 4n_{max} \log(2m/\delta)/k\varepsilon^2] & \text{otherwise} \end{cases}$$

That is, we either get an (ε, δ) -estimate or an upper bound on t . Whenever $k \geq \frac{4n_{max} \log(2m/\delta)}{\varepsilon^2 t}$ we are in the first case with high probability. We note that the lower and upper bounds presented are parameterized on the estimand t , i.e., the bounds depend on the size of what we are estimating. This means that the error bound ε will depend of t , so the relative error is smaller for larger t .

Theorem 4 follows from two main arguments: First we show that if the summary of each set is constructed by selecting elements independently using a hash function then we get a good estimate with high probability. As our summaries are of fixed size, there is a dependence between the variables denoting whether an element is picked for a summary or not. The main technical hurdle is then to bound the error introduced by the dependence.

We then extend the use of signatures, which are well-known to reduce space for k -permutation min-wise hashing, to one-permutation min-wise hashing as used in our estimator. This reduces the number of bits s by a logarithmic factor. Section 5.2 discusses this further.

4. LOWER BOUND

4.1 Preliminaries

We summarize terms and definitions from communication complexity that are used in the lower bound proof.

Communication model. We consider two-player one-way communication protocols: Alice is given input x , Bob is given input y and they need to compute function $f(x, y)$. Each player has his/her own private randomness, as well as a shared uniformly distributed public coin \mathbf{W} of some finite length. Since the protocol is 1-way, the *transcript* of protocol Π consists of Alice's single message to Bob, together with Bob's output bits. For a protocol Π , the maximum transcript length in bits over all inputs is called the *communication cost* of Π . The *communication complexity* $R_\delta(f)$ of function f is the minimal communication cost of a protocol that computes f with probability at least $1 - \delta$.

Mutual information. For random variables X and Y with support \mathcal{X} and \mathcal{Y} and let $p(x, y), p(x), p(y)$ be the joint and marginal distributions respectively. The *entropy* and *conditional entropy* are defined as:

$$H(X) = - \sum_{x \in \mathcal{X}} p(x) \log p(x)$$

$$H(X | Y) = \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} p(x, y) \log \frac{p(y)}{p(x, y)}$$

The *mutual information* is given as:

$$I(X; Y) = H(X) - H(X | Y) = \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} p(x, y) \log \frac{p(x, y)}{p(x)p(y)}$$

We make use of the following rule:

FACT 5. (Chain Rule) For discrete random variables X, Y, Z it holds that $I(X, Y; Z) = I(X; Z) + I(X; Y | Z)$.

For a protocol Π that uses public random coins \mathbf{W} and has transcript $\Pi(X, Y, Z)$ for random variables $X, Y, Z \sim \mu$, the *conditional information cost* of Π with respect to distribution μ is $I(X, Y, Z; \Pi(X, Y, Z) | \mathbf{W})$. For function f we have that the *conditional information complexity* $CIC_\delta^\mu(f)$ is the minimal conditional information cost of any δ -error protocol Π with respect to distribution μ .

Fano's inequality. We make use of Fano's equality which intuitively relates the error probability of a function between random variables to the conditional entropy between them.

DEFINITION 6. Given domains \mathcal{X} and \mathcal{Y} and random variables X, Y on these domains with distribution μ , we say a function $g: \mathcal{Y} \rightarrow \mathcal{X}$ has error δ_g if

$$\Pr_{X, Y \sim \mu}[g(Y) = X] \geq 1 - \delta_g.$$

FACT 7. Let X and Y be a random variables chosen from domains \mathcal{X} and \mathcal{Y} respectively according to distribution μ . There is a deterministic function $g: \mathcal{Y} \rightarrow \mathcal{X}$ with error δ_g , where $\delta_g \leq 1 - \frac{1}{2^{H(\mathcal{X} | \mathcal{Y})}}$.

FACT 8. (Fano's inequality.) Let X and Y be a random variables chosen from domains \mathcal{X} and \mathcal{Y} respectively according to distribution μ . For any reconstruction function $g: \mathcal{Y} \rightarrow \mathcal{X}$ with error δ_g ,

$$H_b(\delta_g) + \delta_g \log(|\mathcal{X}| - 1) \geq H(X | Y).$$

4.2 A Communication Problem and its Application to Set Intersection

Let $r = \Theta(1/\varepsilon^2)$, and $d = n_{max}/t$. We consider a two-party one-way communication problem:

DEFINITION 9. In the **GapAndIndex** problem, Alice has bit vectors $x^1, \dots, x^r \in \{0, 1\}^d$ while Bob has indices $i^1, \dots, i^r \in [d]$, where $[d] = \{1, 2, \dots, d\}$, together with bits $b^1, \dots, b^r \in \{0, 1\}$. Let $\mathbf{x} = (x^1, \dots, x^r)$, $\mathbf{i} = (i^1, \dots, i^r)$, and $\mathbf{b} = (b^1, \dots, b^r)$ and $C > 0$ be a fixed constant. The output of **GapAndIndex**($\mathbf{x}, \mathbf{i}, \mathbf{b}$) is:

$$1 \text{ if } \sum_{j=1}^r (x_{i_j}^j \wedge b^j) \geq \frac{r}{4} + C\varepsilon r$$

$$0 \text{ if } \sum_{j=1}^r (x_{i_j}^j \wedge b^j) \leq \frac{r}{4} - C\varepsilon r.$$

This is a promise problem, and if neither case occurs, the output can be arbitrary.

If the input $(\mathbf{x}, \mathbf{i}, \mathbf{b})$ is in either of the two cases we say the input *satisfies the promise*.

We say a one-way randomized protocol Π for **GapAndIndex** is δ -error if

$\forall \mathbf{x}, \mathbf{i}, \mathbf{b}$ satisfying the promise :

$$\Pr[\Pi(\mathbf{x}, \mathbf{i}, \mathbf{b}) = \text{GapAndIndex}(\mathbf{x}, \mathbf{i}, \mathbf{b})] \geq 1 - \delta,$$

where the probability is over the public and private randomness of Π .

Let κ be the set of randomized one-way δ -error protocols Π . We note that κ is finite for any problem with finite input, as we can always have one player send his/her entire input to the other player.

Then,

$$R_\delta^{1-way}(\text{GapAndIndex}) = \min_{\Pi \in \kappa} \max_{\mathbf{x}, \mathbf{i}, \mathbf{b}, \text{ randomness of } \Pi} |\Pi(\mathbf{x}, \mathbf{i}, \mathbf{b})|,$$

where $|\Pi(\mathbf{x}, \mathbf{i}, \mathbf{b})|$ denotes the length of the transcript with these inputs. Since the protocol is 1-way, we can write this length as $|M(\mathbf{x})| + 1$, where $M(\mathbf{x})$ is Alice's message function in the protocol Π given her input \mathbf{x} , and we add 1 for Bob's output bit. Here, implicitly M also depends on the private randomness of Alice, as well as the public coin \mathbf{W} .

Let μ be the uniform distribution on $\mathbf{x} \in (\{0, 1\}^d)^r$. We use the capital letter \mathbf{X} to denote random \mathbf{x} distributed according to μ . We introduce a distribution on inputs solely for measuring the following notion of information cost of the protocol; we still require that the protocol is correct on every input satisfying the promise with probability $1 - \delta$ over its public and private randomness (for a sufficiently small constant $\delta > 0$).

For a uniformly distributed public coin \mathbf{W} , let

$$CIC_\delta^{\mu, 1-way}(\text{GapAndIndex}) = \min_{\Pi \in \kappa} I(M(\mathbf{X}); \mathbf{X} | \mathbf{W}),$$

where for random variables Y, Z and W , $I(Y; Z | W) = H(Y | W) - H(Y | Z, W)$ is the conditional mutual information. Recall that the conditional entropy $H(Y | W) = \sum_w H(Y | W = w) \cdot \Pr[W = w]$, where w ranges over all

values in the support of W . For any protocol Π ,

$$\begin{aligned} \max_{\mathbf{x}, \mathbf{i}, \mathbf{b}} |\Pi(\mathbf{x}, \mathbf{i}, \mathbf{b})| &= \max_{\mathbf{x}} |M(\mathbf{x})| + 1 \\ &> \max_{\mathbf{x}} |M(\mathbf{x})| \\ &\geq H(M(\mathbf{X}) | \mathbf{W}) \\ &\geq I(M(\mathbf{X}); \mathbf{X} | \mathbf{W}), \end{aligned}$$

which implies that

$$R_\delta^{1-\text{way}}(\text{GapAndIndex}) \geq CIC_\delta^{\mu, 1-\text{way}}(\text{GapAndIndex}).$$

We now consider the application to set intersection. Let $r = 10/\varepsilon^2$ and d be the desired ratio between the intersection size t and the largest set. The idea is to give Alice a subset of elements from $[dr]$, where the characteristic vector of her subset is x^1, \dots, x^r . Also, for each $j \in [r]$, Bob is given the element $d \cdot (j-1) + i^j$ if and only if $b^j = 1$. If Alice and Bob's sets are constructed in this way then the intersection is either of size at most $r/4 - C\varepsilon r$, or of size at least $r/4 + C\varepsilon r$. Hence, a 1-way protocol for approximating the intersection size up to a relative $(1 + \Theta(\varepsilon))$ -factor can be used to distinguish these two cases and therefore solve the **GapAndIndex** promise problem.

To get intersection size t without changing the problem, we duplicate each item $4t/r$ times, which means that the problem becomes distinguishing intersection size at most $t(1 - \Theta(\varepsilon))$ and at least $t(1 + \Theta(\varepsilon))$. By rescaling ε by a constant factor, a 1-way protocol for $(1 + \varepsilon)$ -approximating the intersection of Alice and Bob's sets with constant probability can be used to solve **GapAndIndex** with constant probability. Hence, its space complexity is $\geq CIC_{1/3}^{\mu, 1-\text{way}}(\text{GapAndIndex})$. This holds for any distribution μ for measuring information, though we shall use our choice of μ above.

4.3 The **GapAnd** Problem

For bit vectors \mathbf{z}, \mathbf{z}' of the same length, let $\text{AND}(\mathbf{z}, \mathbf{z}')$ be the vector \mathbf{z}'' in which $z_i'' = z_i \wedge z_i'$. For a vector \mathbf{z} , let $\text{wt}(\mathbf{z})$ denote its Hamming weight, i.e., the number of its coordinates equal to 1.

DEFINITION 10. *In the **GapAnd** problem, Alice and Bob have $\mathbf{z}, \mathbf{z}' \in \{0, 1\}^r$, respectively. We define **GapAnd** to be:*

$$\begin{aligned} &1 \text{ if } \text{wt}(\text{AND}(\mathbf{z}, \mathbf{z}')) \geq \frac{r}{4} + C\varepsilon r \\ &0 \text{ if } \text{wt}(\text{AND}(\mathbf{z}, \mathbf{z}')) \leq \frac{r}{4} - C\varepsilon r. \end{aligned}$$

This is a promise problem, and if neither case occurs, the output can be arbitrary.

4.4 The **Index** Problem

Consider the following **Index** problem.

DEFINITION 11. *In the **Index** problem, Alice has an input $\mathbf{Y} \in \{0, 1\}^d$ and Bob has an input $K \in [d]$, where \mathbf{Y} and K are independent and uniformly distributed over their respective domains. We define **Index** to be:*

$$\begin{aligned} &1 \text{ if } Y_K = 1 \\ &0 \text{ if } Y_K = 0 \end{aligned}$$

Suppose \mathbf{W} is the public coin and κ is the set of randomized one-way δ -error protocols Π . Let γ denote this distribution

on the inputs. Say a 1-way protocol Π for **Index** with private randomness R and public randomness \mathbf{W} is δ -error if

$$\Pr_{(\mathbf{Y}, K) \sim \gamma, R, \mathbf{W}} [\Pi(\mathbf{Y}, K, R, \mathbf{W}) = Y_K] \geq 1 - \delta.$$

Let $M(\mathbf{Y})$ be the message function associated with the 1-way protocol Π (which is a randomized function of R and \mathbf{W}). Let

$$CIC_\delta^{\gamma, 1-\text{way}}(\text{Index}) = \min_{\Pi \in \kappa} I(M(\mathbf{Y}); \mathbf{Y} | \mathbf{W}).$$

FACT 12. *For $\delta \leq \frac{1}{2} - \Omega(1)$, $CIC_\delta^{\gamma, 1-\text{way}}(\text{Index}) = \Omega(d)$.*

PROOF. We note that this fact is folklore, but existing references, e.g., Theorem 5.5 of [1] only explicitly state the bound for deterministic protocols, whereas we want such a bound for protocols with both private randomness and public randomness \mathbf{W} . We provide the simple proof here.

Let Π be a δ -error protocol with (randomized) message function M . Let $Y = (Y_1, \dots, Y_d)$. By the chain rule,

$$I(M(\mathbf{Y}); \mathbf{Y} | \mathbf{W}) = \sum_{i=1}^d I(M(\mathbf{Y}); Y_i | Y_1, \dots, Y_{i-1}, \mathbf{W}).$$

By independence and the fact that conditioning cannot increase entropy,

$$\sum_{i=1}^d I(M(\mathbf{Y}); Y_i | Y_1, \dots, Y_{i-1}, \mathbf{W}) \geq \sum_{i=1}^d I(M(\mathbf{Y}); Y_i | \mathbf{W}).$$

If Π is δ -error for $\delta = 1/2 - \Omega(1)$, then by Markov's inequality, for an $\Omega(1)$ fraction of i , $\Pi(\mathbf{Y}, i) = Y_i$ with probability $1/2 + \Omega(1)$. Call such an i *good*. Then

$$\begin{aligned} \sum_{i=1}^d I(M(\mathbf{Y}); Y_i | \mathbf{W}) &\geq \Omega(d) \cdot \min_{\text{good } i} I(M(\mathbf{Y}); Y_i | \mathbf{W}) \\ &= \Omega(d) \cdot \min_{\text{good } i} (1 - H(Y_i | M(\mathbf{Y}), \mathbf{W})). \end{aligned}$$

By Fano's inequality (Fact 8) and using that i is good, we have $H(Y_i | M(\mathbf{Y}), \mathbf{W}) = 1 - \Omega(1)$. This completes the proof. \blacksquare

4.5 Proof of Theorem 3

PROOF. It suffices to prove the theorem for a sufficiently small constant probability of error δ , since

$$R_{1/3}^{1-\text{way}}(\text{GapAndIndex}) = \Theta(R_\delta^{1-\text{way}}(\text{GapAndIndex})).$$

Let Π be a 1-way randomized (both public and private) δ -error protocol for **GapAndIndex**. For ease of presentation, we let $M = M(\mathbf{X})$ when the input \mathbf{X} is clear from context. Note that M also implicitly depends on Alice's private coins as well as a public coin \mathbf{W} . We need to show that $I(M; \mathbf{X} | \mathbf{W})$ is $\Omega(rd)$, for $r = \Theta(\varepsilon^{-2})$.

We start with the following claim, which does not directly look at the information Π conveys about its inputs, but rather the information Π conveys about certain bits in its input.

$$\text{CLAIM 13. } I(M; X_1^1, \dots, X_r^r | \mathbf{W}) = \Omega(r).$$

PROOF. We will need the following fact, which follows from work by Braverman et al. [4].

FACT 14. ([4]) Let ρ be the uniform distribution on bits c^1, \dots, c^r and d^1, \dots, d^r . Let $\mathbf{C} = (C^1, \dots, C^r)$ and $\mathbf{D} = (D^1, \dots, D^r)$ for vectors \mathbf{C} and \mathbf{D} drawn from ρ .

There is a sufficiently small constant δ for which for any private randomness protocol Π which errs with probability at most δ on GapAnd , over inputs \mathbf{C} and \mathbf{D} drawn from ρ and the private randomness of Π and the public randomness \mathbf{W} , satisfies

$$I(\Pi(\mathbf{C}, \mathbf{D}); \mathbf{C}, \mathbf{D} \mid \mathbf{W}) = \Omega(r).$$

PROOF. The work of Braverman et al. [4] establishes this for the problem of deciding if $\sum_{i=1}^r (C^i \oplus D^i) \geq r/2 + \sqrt{r}$ or $\sum_{i=1}^r (C^i \oplus D^i) \leq r/2 - \sqrt{r}$, which corresponds to the Hamming distance $\Delta(\mathbf{C}, \mathbf{D})$ of vectors drawn from ρ .

If $\text{wt}(\mathbf{C})$ denotes the Hamming weight of \mathbf{C} , then we have

$$\text{wt}(\mathbf{C}) + \text{wt}(\mathbf{D}) - 2 \cdot \text{And}(\mathbf{C}, \mathbf{D}) = \Delta(\mathbf{C}, \mathbf{D}),$$

where $\text{And}(\mathbf{C}, \mathbf{D})$ is the number of coordinates i for which $C^i = D^i = 1$. Therefore, if Alice and Bob exchange $\text{wt}(\mathbf{C})$ and $\text{wt}(\mathbf{D})$ using $2 \log r$ bits, then together with a protocol Π for GapAnd , they can solve this Hamming distance problem. It follows that

$$I(\Pi(\mathbf{C}, \mathbf{D}), \text{wt}(\mathbf{C}), \text{wt}(\mathbf{D}); \mathbf{C}, \mathbf{D} \mid \mathbf{W}) = \Omega(r),$$

and so by the chain rule for mutual information one has $I(\Pi(\mathbf{C}, \mathbf{D}); \mathbf{C}, \mathbf{D} \mid \mathbf{W}) = \Omega(r) - I(\text{wt}(\mathbf{C}), \text{wt}(\mathbf{D}); \mathbf{C}, \mathbf{D} \mid \mathbf{W}, \Pi(\mathbf{C}, \mathbf{D})) = \Omega(r) - H(\text{wt}(\mathbf{C}), \text{wt}(\mathbf{D})) = \Omega(r) - 2 \log r = \Omega(r)$. \blacksquare

First, observe that if \mathbf{I} denotes a uniformly random value of \mathbf{i} , then

$$\begin{aligned} & I(M; X_{I^1}^1, \dots, X_{I^r}^r \mid \mathbf{I}, \mathbf{W}) \\ &= H(M \mid \mathbf{I}, \mathbf{W}) - H(M \mid X_{I^1}^1, \dots, X_{I^r}^r, \mathbf{I}, \mathbf{W}) \\ &= H(M) - H(M \mid X_{I^1}^1, \dots, X_{I^r}^r) \\ &= I(M; X_{I^1}^1, \dots, X_{I^r}^r \mid \mathbf{W}), \end{aligned}$$

where we use that M and $X_{I^1}^1, \dots, X_{I^r}^r$ are jointly independent of \mathbf{I} , conditioned on \mathbf{W} .

Hence, using also the independence of \mathbf{I} and \mathbf{W} ,

$$\begin{aligned} & I(M; X_{I^1}^1, \dots, X_{I^r}^r \mid \mathbf{W}) \\ &= I(M; X_{I^1}^1, \dots, X_{I^r}^r \mid \mathbf{I}, \mathbf{W}) \\ &= \sum_i I(M; X_{i^1}^1, \dots, X_{i^r}^r \mid \mathbf{I} = \mathbf{i}, \mathbf{W}) \cdot \Pr[\mathbf{I} = \mathbf{i}] \\ &\geq \frac{1}{2} \min_i I(M; X_{i^1}^1, \dots, X_{i^r}^r \mid \mathbf{I} = \mathbf{i}, \mathbf{W}). \end{aligned}$$

We claim that for each \mathbf{i} , $I(M; X_{i^1}^1, \dots, X_{i^r}^r \mid \mathbf{I} = \mathbf{i}, \mathbf{W}) = \Omega(r)$. To see this, define a 1-way protocol $\Pi_{\mathbf{i}}$ for GapAnd as follows. Alice and Bob are given inputs \mathbf{C} and \mathbf{D} to GapAnd , respectively, distributed according to ρ . For each $j \in [r]$, Alice sets $X_{i^j}^j = C^j$, while Bob sets $B^j = D^j$. Alice then chooses an independent uniform random bit for X_k^j for each j and $k \neq i^j$. The players then run the protocol $\Pi(\mathbf{X}, \mathbf{i}, \mathbf{B})$, and outputs whatever Π outputs.

By construction, $\Pi_{\mathbf{i}}(\mathbf{C}, \mathbf{D}) = \text{GapAnd}(\mathbf{X}, \mathbf{i}, \mathbf{B})$, and so the correctness probability of $\Pi_{\mathbf{i}}$ is at least $1 - \delta$.

Moreover, if $M_{\mathbf{i}}$ denotes the message function of Alice in $\Pi_{\mathbf{i}}$, then by construction we have that for a sufficiently small

constant δ ,

$$\begin{aligned} & I(M; X_{i^1}^1, \dots, X_{i^r}^r \mid \mathbf{I} = \mathbf{i}, \mathbf{W}) \\ &= I(M_{\mathbf{i}}(X_{i^1}^1, \dots, X_{i^r}^r); X_{i^1}^1, \dots, X_{i^r}^r \mid \mathbf{W}) = \Omega(r) \end{aligned}$$

using Fact 14. \blacksquare

By Claim 13 and the chain rule, for $\Omega(1)$ fraction of $j \in [r]$ we have $I(M; X_{I^j}^j \mid X_{I^1}^1, \dots, X_{I^{j-1}}^{j-1}, \mathbf{W}) = \Omega(1)$. Call such an index j *informative*. For each informative j , a value x of the vector $(X_{I^1}^1, \dots, X_{I^{j-1}}^{j-1})$ is *informative* if $I(M; X_{I^j}^j \mid (X_{I^1}^1, \dots, X_{I^{j-1}}^{j-1}) = x, \mathbf{W}) = \Omega(1)$. Since

$$I(M; X_{I^j}^j \mid X_{I^1}^1, \dots, X_{I^{j-1}}^{j-1}, \mathbf{W}) = \Omega(1),$$

it follows that an $\Omega(1)$ fraction of x are informative for an informative j .

We now lower bound $I(M(\mathbf{X}); \mathbf{X} \mid \mathbf{W})$. Let $\mathbf{X}^{<j} = (X^1, \dots, X^{j-1})$. Applying the chain rule, as well as the definition of informative and the bounds on informative j and x above,

$$\begin{aligned} & I(M(\mathbf{X}); \mathbf{X} \mid \mathbf{W}) \\ &= \sum_{j=1}^r I(M(\mathbf{X}); X^j \mid \mathbf{X}^{<j}, \mathbf{W}) \\ &\geq \sum_{j=1}^r I(M(\mathbf{X}); X^j \mid X_{I^1}^1, \dots, X_{I^{j-1}}^{j-1}, \mathbf{W}) \\ &= \sum_{j=1}^r \sum_x I(M(\mathbf{X}); X^j \mid (X_{I^1}^1, \dots, X_{I^{j-1}}^{j-1}) = x, \mathbf{W}) \\ &\quad \cdot \Pr[(X_{I^1}^1, \dots, X_{I^{j-1}}^{j-1}) = x] \\ &\geq \sum_{\text{inform. } j, x} I(M(\mathbf{X}); X^j \mid (X_{I^1}^1, \dots, X_{I^{j-1}}^{j-1}) = x, \mathbf{W}) \\ &\quad \cdot \Pr[(X_{I^1}^1, \dots, X_{I^{j-1}}^{j-1}) = x] \\ &\geq \Omega(r) \cdot \min_{\text{inform. } j, x} I(M(\mathbf{X}); X^j \mid (X_{I^1}^1, \dots, X_{I^{j-1}}^{j-1}) = x, \mathbf{W}), \end{aligned}$$

where the first inequality follows from the fact that X^j is independent of $\mathbf{X}^{<j}$, together with the fact that conditioning cannot increase entropy.

We now lower bound

$$\min_{\text{informative } j, x} I(M(\mathbf{X}); X^j \mid (X_{I^1}^1, \dots, X_{I^{j-1}}^{j-1}) = x, \mathbf{W}).$$

To do so, we build a 1-way protocol $\Pi_{j,x}$ with j and x hardwired, for solving the Index problem with a uniform distribution γ on its inputs. Suppose Alice is given the random input $Y \in \{0, 1\}^d$, and Bob is given the random input $K \in [d]$, where Y and K are uniformly distributed over $\{0, 1\}^d$ and $[d]$, respectively. Alice and Bob create inputs for protocol Π as follows. Namely, Alice sets $X^j = Y$, and uses the hardwiring of x to set $(X_{I^1}^1, \dots, X_{I^{j-1}}^{j-1}) = x$. Further, Alice uses her private randomness to fill in the remaining coordinates of X^1, \dots, X^{j-1} , as well as to choose X^{j+1}, \dots, X^r (all coordinates of such vectors are independent of Bob's inputs and uniformly distributed, so Alice can choose such inputs without any communication). Further, Bob sets $I^j = K$, and chooses $I^{j'}$ for $j' \neq j$ uniformly and independently in $[d]$. Bob also chooses his input \mathbf{B} to be independent of all other inputs and uniformly distributed.

Given this setting of inputs, in $\Pi_{j,x}$ Alice and Bob then run protocol Π on these inputs, resulting in a message func-

tion $M'(\mathbf{Y}) = M(\mathbf{X})$. Since j and x are informative, it follows that $I(M(\mathbf{X}); X_{I_j}^j | (X_{I_1}^1, \dots, X_{I_{j-1}}^{j-1}) = x, \mathbf{W}) = \Omega(1)$, which implies that $I(M'(\mathbf{Y}); Y_K | \mathbf{W}) = \Omega(1)$, or equivalently,

$$H(Y_K | M'(\mathbf{Y}), \mathbf{W}) = 1 - \Omega(1).$$

It follows from Fact 7 that Bob, given $M'(\mathbf{Y})$ and \mathbf{W} , can predict Y_k with probability $1/2 + \Omega(1)$, and solve Index on the uniform distribution γ . By Fact 12, it follows that $I(M'(\mathbf{Y}); \mathbf{Y} | \mathbf{W}) = \Omega(d)$. Notice, though, that by construction of $\Pi_{j,x}$ that $I(M'(\mathbf{Y}); \mathbf{Y} | \mathbf{W}) = I(M(\mathbf{X}); X^j | (X_{I_1}^1, \dots, X_{I_{j-1}}^{j-1}) = x, \mathbf{W})$.

We conclude that $I(M(\mathbf{X}); \mathbf{X} | \mathbf{W}) = \Omega(dr)$, which completes the proof. \blacksquare

5. UPPER BOUND

Recall that sets $S_1, S_2, \dots \subseteq [u]$ of sizes n_1, n_2, \dots where $n_{\max} = \max_i n_i$ are given, and we wish to obtain an (ε, δ) -estimate of $t = |S_1 \cap \dots \cap S_m|$ using one-permutation k -min summaries as described in Section 2.2. Theorem 4 defines an estimator (see Figure 1 for pseudocode). In our proof of Theorem 4 we will assume that the hash function used to construct the summaries is random and fully independent. In many applications it will be possible to achieve this by simply maintaining a hash table of values during the construction phase. However, Section 6 shows how to replace the full randomness assumption with concrete hash functions in case the number of different hash values is too large to store.

For an intersection query on m sets the main insight is that our estimator relies only on the maximum set size n_{\max} in contrast to the known k -permutation estimator that depends on the size of the union, making it less accurate given the same space (see Table 1). The space needed to store a summary that gives an (ε, δ) -estimate is $O\left(\frac{n_{\max} \log(m/\delta) \log u}{\varepsilon^2}\right)$ bits. In Section 5.2 we show that this can be reduced almost by a factor $\log u$ by use of signatures.

5.1 Proof of Theorem 4

Recall that $k_{\min}(S_i)$ denotes the size- k one-permutation min-wise summary of S_i and the indicator variable $\hat{X}_j^{(i)}$ denotes the event that item j is chosen for the size- k one-permutation min-wise summary of S_i as defined below: $\hat{X}_j^{(i)} = 1$ if $j \in k_{\min}(S_i)$ and $\hat{X}_j^{(i)} = 0$ otherwise.

High-level proof strategy. Observe that $\Pr[\hat{X}_j^{(i)} = 1] = k/n_i$. Our algorithm uses size k summaries so for each set S_i we have $\sum_{j=1}^{n_i} \hat{X}_j^{(i)} = k$, which causes negative dependence between the indicator variables [14], i.e., when an item is in the summary of a set then the other items have smaller probability of being in the that summary. The main technical hurdle is showing that even with such a dependence one can use the intersection size between the summaries to estimate the intersection size of the sets.

To do this we analyze the case where for each S_i , the variables $X_1^{(i)}, X_2^{(i)}, \dots, X_{n_i}^{(i)}$ are independent random variables:

$$X_j^{(i)} = \begin{cases} 1 & \text{if } h(j) \leq k/n_i \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where $h : u \mapsto [0, 1]$ is a fully random hash function. Let the setting with negative dependence be called *the dependent*

case and the case using (1) be *the independent case*. The independent case conditioned on the sum of the variables being k is identically distributed as the dependent case. Therefore the final step is to bound the additional error probability of going from the independent case to the dependent one.

First we bound the probability of sampling k specific items given the number of sampled items is k . Let $i \in [m]$ and $\tilde{S}_i = \{x \in S_i | h(x) \leq k/n_i\}$ be a sample of $S_i \subseteq [u]$ picked according to (1). An important consequence of picking elements to be in summaries is that of *consistent sampling*: If the hash value of an element from the intersection is one of the k smallest hash values computed, it will be guaranteed to be sampled in all sets. The following lemma shows that any specific outcome of a sample has equal probability given we restrict a sample to be size k .

LEMMA 15. *If $S_i \subseteq [u]$ and $\{i_1, i_2, \dots, i_k\}$ is a specific size k outcome then*

$$\Pr \left[\tilde{S}_i = \{i_1, i_2, \dots, i_k\} \mid \sum_{j=1}^{n_i} X_j^{(i)} = k \right] = \frac{1}{\binom{n_i}{k}}$$

PROOF. We have:

$$\begin{aligned} \Pr \left[\tilde{S}_i = \{i_1, i_2, \dots, i_{n_i}\} \right] &= \binom{k}{n_i} \left(1 - \frac{k}{n_i}\right)^{n_i - k} \\ \Pr \left[\sum_{j=1}^{n_i} X_j^{(i)} = k \right] &= \binom{n_i}{k} \binom{k}{n_i} \left(1 - \frac{k}{n_i}\right)^{n_i - k} \end{aligned}$$

The final step of the lemma follows from Bayes theorem:

$$\begin{aligned} \Pr \left[\tilde{S}_i = \{i_1, i_2, \dots, i_k\} \mid \sum_{j=1}^{n_i} X_j^{(i)} = k \right] \\ = \frac{\Pr \left[\tilde{S}_i = \{i_1, i_2, \dots, i_{n_i}\} \right]}{\Pr \left[\sum_{j=1}^{n_i} X_j^{(i)} = k \right]} = \frac{1}{\binom{n_i}{k}}. \end{aligned}$$

\blacksquare

We show the lower bound on the probability of the size of any \tilde{S}_i being equal to its expectation k :

LEMMA 16. *For a sample \tilde{S}_i of S_i we have*

$$\Pr \left[|\tilde{S}_i| = k \right] = \Omega \left(\frac{1}{\sqrt{k}} \right). \quad (2)$$

PROOF. The mean μ of $|\tilde{S}_i|$ is the most likely outcome, i.e., $\Pr \left[|\tilde{S}_i| = k \right] \geq \Pr \left[|\tilde{S}_i| = j \right]$ for $1 \leq j \leq u$ holds due to $E \left[\sum_{j=1}^{n_i} X_j^{(i)} \right] = k$ and the mode of binomial distributions [24]. Next step is showing that $|\tilde{S}_i|$ is more likely to be within $2\sqrt{k}$ of the mean $\mu = k$ than not, that is, $\Pr \left[\left| \sum_{j=1}^{n_i} X_j^{(i)} - k \right| \geq 2\sqrt{k} \right] \leq \frac{1}{2}$. This follows from the Chernoff bounds on the sum $\sum_{j=1}^{n_i} X_j^{(i)}$:

$$\begin{aligned} \Pr \left[\left| \sum_{j=1}^{n_i} X_j^{(i)} - k \right| \geq 2\sqrt{k} \right] &\leq 2 \exp \left(\frac{-k \left(\frac{2\sqrt{k}}{k} \right)^2}{2} \right) \\ &\leq \frac{1}{2} \quad \forall k, i > 0. \end{aligned}$$

\blacksquare

Input: Sets $S_1, S_2, \dots \subseteq [u]$

Output: k -min summaries for all S_i

- 1 $h \leftarrow$ fully independent random hash function
- 2 **foreach** S_i **do**
- 3 $k_i \leftarrow$ the k th smallest $h(x)$ for $x \in S_i$
- 4 $k_{\min}(S_i) \leftarrow \{x \mid x \in S_i \wedge h(x) \leq k_i\}$

(a) Pre-processing the sets.

Input: k -min summaries and set sizes

$k_{\min}(S_1), n_1 = |S_1|, \dots$ and query set $M \subseteq \mathbf{N}$

Output: X : An (ε, δ) -estimation of $t = |\bigcap_{i \in M} S_i|$

- 1 $n_{\max} \leftarrow \max_{i \in M} n_i$
- 2 $X \leftarrow |\bigcap_{i \in M} k_{\min}(S_i)| n_{\max}/k$

(b) Computing the estimator. The output is an (ε, δ) -estimator whenever $X > 3n_{\max} \log(1/\delta)/k\varepsilon^2$ (See Theorem 4).

Figure 1: Pseudocode for performing pre-processing and computing the estimator.

Let S be the elements of the size- t intersection and \tilde{S}_{\max} be the sample of the largest set S_{\max} . We show that if the summary size k satisfies

$$k \geq \frac{2n_{\max} \log(2m/\delta)}{\varepsilon^2 t} \quad (3)$$

then properties 1 and 2 below are satisfied.

PROPERTY 1. $|\tilde{S}_{\max} \cap S|$ is an $(\varepsilon, \delta/2)$ -estimate of $t \frac{k}{n_{\max}}$.

PROPERTY 2. $\forall_i |\tilde{S}_i \cap S| \geq t(1-\varepsilon) \frac{k}{n_{\max}}$ with probability at least $1 - \delta/2m$.

We show that the given properties hold for sufficiently large k , given by (3).

LEMMA 17. If (3) and $0 \leq \varepsilon, \delta \leq 1$ then properties 1 and 2 hold.

PROOF. We show that property 1 holds when (3) holds. This follows from Chernoff bounds on $\sum_{j=1}^t X_j^{(max)}$:

$$\begin{aligned} \gamma_1 &= \Pr \left[|\tilde{S}_i \cap S| \notin \left[t \frac{k}{n_{\max}} (1-\varepsilon), t \frac{k}{n_{\max}} (1+\varepsilon) \right] \right] \\ &< 2 \exp \left(-\frac{\varepsilon^2 t k}{3n_{\max}} \right). \end{aligned}$$

Since $k \geq \frac{2n_{\max} \log(2m/\delta)}{\varepsilon^2 t}$ the error probability is $\gamma_1 \leq \frac{\delta}{2}$, thus property 1 holds.

Now we are to show that the given k implies property 2 holds, i.e., the size of the intersection between any single \tilde{S}_i sample and intersection S is at least the expected size of the intersection between the sample of the largest set, \tilde{S}_{\max} and S . The intersection of any sample \tilde{S}_i and S has expectation $\mu = E[|\tilde{S}_i \cap S|] = t \frac{k}{n_i}$. Since $n_{\max} \geq n_i$, it holds that $\forall_i t \frac{k}{n_{\max}} \leq t \frac{k}{n_i}$ and thus we bound error γ_2 :

$$\begin{aligned} \gamma_2 &= \Pr \left[\sum_{j=1}^t X_j^{(max)} < (1-\varepsilon) t \frac{k}{n_{\max}} \right] \\ &\leq \Pr \left[\sum_{j=1}^t X_j^{(i)} < (1-\varepsilon) t \frac{k}{|S_i|} \right] \leq \exp \left(-\frac{\varepsilon^2 t k}{2n_{\max}} \right). \end{aligned}$$

Since $k \geq \frac{2n_{\max} \log(2m/\delta)}{\varepsilon^2 t}$ the error probability is $\gamma_2 \leq \frac{\delta}{2m}$, thus property 2 holds. \blacksquare

We will now show that the independent case provides an estimator with the desired guarantees.

LEMMA 18. If (3) holds and for $0 \leq \varepsilon, \delta \leq 1$ then $|\bigcap_{i \in [m]} \tilde{S}_i| n_{\max}$ is an (ε, δ) -estimate of t .

PROOF. First we need that $|\tilde{S}_{\max} \cap S| \leq (1+\varepsilon) t \frac{k}{n_{\max}}$ with probability $\geq 1 - \delta$. By Lemma 17 this holds, as property 1 holds since k satisfies (3). We now argue:

$$\left| \bigcap_{i \in [m]} \tilde{S}_i \right| \geq (1-\varepsilon) t \frac{k}{n_{\max}} \text{ with probability } \geq 1 - \delta. \quad (4)$$

Let $z = (1-\varepsilon) t \frac{k}{n_{\max}}$, then by Lemma 17 we have that for each set S_i its sample \tilde{S}_i contains at least z items from S with probability $\geq 1 - \delta/2m$ where these z items are sampled from all sets as they are in S and hence (4) holds. To show that

$$\left| \bigcap_{i \in [m]} \tilde{S}_i \right| \leq (1+\varepsilon) t \frac{k}{n_{\max}} \text{ with probability } \geq 1 - \delta \quad (5)$$

holds we need that $|\tilde{S}_{\max} \cap S| \leq (1+\varepsilon) t \frac{k}{n_{\max}}$ with probability $\geq 1 - \delta$. This follows directly from property 1 holding since k satisfies (3) as shown in Lemma 17.

We now show that our estimator computes an (ε, δ) -estimate, i.e., it holds that,

$$\frac{|\bigcap_{i \in [m]} \tilde{S}_i| n_{\max}}{k} \in [(1-\varepsilon)t, (1+\varepsilon)t]$$

with probability at least

$$1 - \left(\frac{\delta}{2} + m \frac{\delta}{2m} \right) \geq 1 - \delta.$$

By (5) and (4) we have the relative error of at most ε as required. To bound the error probability we apply the union bound on the error probabilities given by Lemma 17. As we have error probability $\delta/2$ on property 1 and error probability $\delta/2m$ on property 2, by the union bound we get $\leq (\delta/2 + m\delta/2m) = \delta$ where the factor m on the second term comes from the union bound over all m sets. \blacksquare

For each set S_i let B_i denote the set of samples where property 1 or 2 does *not* hold. We have probability $\Pr[\tilde{S}_i \in B_i]$ of the estimator based on samples \tilde{S}_i being bad. We now relate the independent case where a sample has expected size k to the case where k -min summaries are used and thus we have samples of strictly size k .

LEMMA 19. If (3) holds and $0 \leq \varepsilon, \delta \leq 1$ then

$$\Pr[\tilde{S}_i \in B_i \mid |\tilde{S}_i| = k] \leq \delta \sqrt{k}.$$

For a specific itemset $I = \{i_1, i_2, \dots, i_k\}$ we have

$$\Pr \left[\tilde{S}_i = I \mid \sum_{j=1}^{n_i} X_j^{(i)} = k \right] = \Pr [k_{\min}(S_i) = I] = \frac{1}{\binom{n_i}{k}} \quad (6)$$

PROOF. An upper bound of the conditional probability can be obtained through Bayes theorem:

$$\Pr [\tilde{S}_i \in B \mid |\tilde{S}_i| = k] \leq \frac{\Pr [\tilde{S}_i \in B]}{\Pr [|\tilde{S}_i| = k]}.$$

The probability of the sample being of size k was bounded in (2) and by union bound on the error probabilities found in Lemma 17 we get.

$$\Pr [\tilde{S}_i \in B \mid |\tilde{S}_i| = k] \leq \frac{\Pr [\tilde{S}_i \in B]}{\Pr [|\tilde{S}_i| = k]} \leq \delta / \frac{1}{\sqrt{k}} = \delta \sqrt{k}.$$

Now we argue that (6) holds, i.e., that the conditional distribution of any sample $|\tilde{S}_i| = k$ is the same as that of $k_{\min}(S_i)$. This follows directly from Lemma 15 and from $\Pr [k_{\min}(S_i) = I] = \frac{1}{\binom{n_i}{k}}$. ■

PROOF. (Theorem 4.) By Lemma 18 we have that X is an (ε, δ) -estimate of t in the independent case whenever the expected number of elements k in our summaries satisfy (3). Lemma 19 relates the independent case to the dependent case with fixed summary size, showing that X is an $(\varepsilon, \delta\sqrt{k})$ -estimate when (3) holds. To show Theorem 4 we consider two cases for t .

1. If $t \geq 2n_{\max} \log(2m/\delta)/k\varepsilon^2$ then (3) is satisfied, so X is an $(\varepsilon, \delta\sqrt{k})$ -estimate of t . Since $\varepsilon < 1/4$ we get that $X < 3n_{\max} \log(2m/\delta)/k\varepsilon^2$ implies

$$X/(1-\varepsilon) < 4n_{\max} \log(2m/\delta)/k\varepsilon^2.$$

So as long as $t \in [X/(1+\varepsilon); X/(1-\varepsilon)]$, which happens with probability $1 - \delta\sqrt{k}$, we get a true answer regardless of whether the first or second answer is returned.

2. If $t < 2n_{\max} \log(2m/\delta)/k\varepsilon^2$ then the probability that $X > 3n_{\max} \log(2m/\delta)/k\varepsilon^2$ is at most $\delta\sqrt{k}$. This is because X is dominated by an estimator X' derived from X by artificially increasing the intersection size to that required by (3). This means that with probability $1 - \delta\sqrt{k}$ the algorithm correctly reports that t is in the interval $[0; 4n_{\max} \log(2m/\delta)/k\varepsilon^2]$. ■

5.2 Use of signatures for the upper bound

An advantage of k -permutation min-wise hashing is that it can easily be combined with signatures to decrease space usage, i.e., elements from u in the min-hash can be replaced with hash values using significantly fewer bit. As shown by Li and König [22], using b -bit signatures, where b is a small integer, allows us to increase k by a factor $\log(u)/b$ without increasing the space usage. With a suitable estimator that takes the signature collisions into account, the net result is an increase in precision for a given space usage. It is a nontrivial matter to extend the estimator to work for the intersection of more than two sets when b is small. The case of three sets was investigated in [21].

It seems to be less well known that one-permutation hashing allows a similar space saving. The idea is to consider signatures of $\log(k) + b$ bits, and store the *set* of signatures for each set $k_{\min}(X)$. By using an appropriate encoding of the signature set the space usage becomes roughly $k(b + \log e)$ bits, see e.g. [25]. There even exist methods that use word-level parallelism to compute the set of signatures that are in common between two such encodings [2, Lemma 3], meaning that there is a speedup in comparing two summaries that is similar to the factor saved in space usage. At least in theory, this means that the difference between the efficiency of k -permutation and one-permutation schemes compressed using signatures is not so large.

We now argue that if we choose a signature hash function $h : [u] \rightarrow \{0, 1\}^b$ where $b \geq \log(2k^2/\delta)$, a signature collision that affects the estimate will occur with probability at most $\delta/2$, independent of the number of sets considered. Recall that k is the size of a min-hash, and consider a specific set of min-hashes $k_{\min}(S_j)$, $j = 1, \dots, m$. If we replace $k_{\min}(S_j)$ by the set $h(k_{\min}(S_j))$ of signatures there is a chance that $|\cap_j h(k_{\min}(S_j))|$ is different from $|\cap_j k_{\min}(S_j)|$ because of collision of elements in some set I with at least one element in each min-hash. We define an *i-cover* as a set I where $|I| = i$ and $\forall j : I \cap k_{\min}(S_j) \neq \emptyset$, i.e., an *i-cover* is a set of i elements that includes an element from every minhash. We now argue that there is a low probability that there exists an *i-cover* with $i > 1$ for which all elements have the same signature under h . For now we assume that h is fully random, which means that the probability a particular *i-cover* colliding is at most

$$(2^{-b})^{i-1} = \left(\frac{\delta}{2k^2} \right)^{i-1}.$$

For $i \leq m$ we have at most k^i possible *i-covers*, so by a union bound the probability of any colliding *i-cover* occurring is at most

$$\sum_{i=2}^m k^i \left(\frac{\delta}{2k^2} \right)^{i-1} \leq \delta/2.$$

We conclude that with probability at least $1 - \delta/2$ we end up with exactly $|\cap_i k_{\min}(S_i)|$ signatures in the intersection, meaning that the result is the same as when storing the elements of $k_{\min}(S_1), \dots, k_{\min}(S_m)$. Hence one can simply think of the sets $k_{\min}(S_i)$, with the understanding that they can be replaced by a representation of size roughly $k \log(e2k^2/\delta)$ bits using a suitable encoding of signatures.

6. HASH FUNCTIONS OF LIMITED INDEPENDENCE

Until now we have assumed to have access to a fully random hash function on the sets. In this section we show that there are realizable hash functions of limited independence such that our results hold. Thorup [30] recently showed that for Jaccard similarity (and hence intersection size) estimation with one-permutation min-wise summaries it suffices to use a pairwise independent hash function. However, this does not extend to the setting where we seek the intersection size of many sets (see Theorem 20).

We argue that k -wise independence is sufficient for the hash function used to construct the one-permutation min-wise summaries and that m -wise independence is sufficient

for the hash functions used to create signatures as described in Section 5.2.

6.1 Hash functions for one-permutation min-wise summaries

We will argue that k -wise independent hash functions are sufficient for the hash function used to create the summaries.

For n variables X_1, \dots, X_n , $X = \sum_i^n X_i$, $\mu = E[X]$ and $\delta > 0$ then by [27] we have that if the variables X_1, \dots, X_n are $\lceil \frac{\mu\delta}{1-\mu/n} \rceil$ -wise independent, the Chernoff tail bounds hold. Examining the tail bounds used in Section 5.1 we see that if we impose the additional constraint $\delta \leq 1 - k/n$ then $\lceil \frac{\mu\delta}{1-\mu/n} \rceil \leq k$ and hence k -wise independence is sufficient for the construction of our summaries.

6.2 Hash functions for signatures

We will now argue that m -wise independent hash functions are sufficient to obtain error probability $\leq \delta$ when being used to create signatures. This follows directly from the fact that we consider collisions in terms of i -covers for $i \leq m$ and apply a summation of m terms to bound the error probability to be $\leq \delta/2$. For the family of hash functions we will use the construction of Siegel [28]. This construction gives a RAM data structure of space $O(u\sqrt{\lg k/\lg u + \varepsilon} \lg v)$ bits when hashing from $\{0, \dots, u-1\}$ to $\{0, \dots, v-1\}$. A function from the family can be evaluated in constant worst-case time and it is k -wise independent with high probability. In particular, for $m = k = u^{O(1)}$ we have space usage $O(u^\varepsilon \lg v)$ for some constant $\varepsilon > 0$.

6.3 Lower bound for c -wise independent hash functions

Motivated by recent work by Thorup [30] showing 2-wise independent hash functions to work well for Jaccard estimation we will now consider an instance where any estimator based on the k smallest hash values of a c -wise independent hash function will not be unbiased. In particular the argument follows from the existence of small families of hash functions.

THEOREM 20. *Let $[u]$ be the universe of elements and $h : [u] \mapsto 0, \dots, p-1$ be any c -wise independent hash function for $c = O(1)$. There exists an instance on p^2 sets S_1, \dots, S_{p^2} with intersection size $t = |\cap_i S_i| = n - k$. For any estimator \tilde{t} for t that guarantees a relative error bound and is based on k size min-wise summaries constructed using h it holds that \tilde{t} is not unbiased.*

For Theorem 20 we construct an instance on p^c sets where one of the k one-permutation min-wise summaries will hold no elements from S with high probability.

PROOF. Let $h : [u] \mapsto \{0, \dots, p-1\}$ be a c -independent hash function where $c < \log_p m$. We will consider an instance on $m > p^c$ sets that has large intersection S , but where an unbiased estimator of the intersection size $|S|$ using the smallest k hash values is not possible with high probability.

For any h there exists a set M_z of size k where $h(M_z) = \{0, \dots, k-1\}$, i.e., the k elements of M_z map to the k smallest possible hash values. Let $S_i = M_i \cup S$ for $0 \leq i < p^c$ be n -sized sets where S is the intersecting elements to be specified later. We have $h(S_z) = M_z \cup h(S) = \{0, \dots, k-1\} \cup h(S)$

and $z \in \{0, \dots, p^c\}$, i.e., by the existence of size p^c families of hash function there is a hash function that hashes k elements from a particular set S_z to the k smallest possible hash values. It follows that if $\forall j \in h(S) j \geq k$ then the set of the k smallest hash values will contain no elements from S , even though we have size $|S| = n - k$. For a uniformly random $n - k$ -sized set S we have $\Pr[\forall t \in h(S) t \geq k] = \left(1 - \frac{k}{p}\right)^{n-k}$ which is ≈ 1 for $k \ll n$.

Hence if we consider the intersection S of all $m > p^c$ sets S_i it will hold with high probability that this instance will have intersection size $|S| = n - k$ but no elements from S in the set of the k smallest hash values. Consider the case of there being no elements from S in the set of the k smallest hash values and let \tilde{t} be an estimate of $|S|$. Any estimate \tilde{t} of $|S|$ with relative bounded error that is based on p^c min-wise summaries will be unable to distinguish the case of $|S| = 0$ from $|S| = n - k$ when there are no elements from S in the set of the k smallest hash values. Thus when presented with such a set the estimate will always be that $\tilde{t} = 0$. Let ϕ be the probability of there being no elements in from S in the set of the k smallest hash values. Then let the outcome of the random variable X be the estimate \tilde{t} . We have $E[X] \leq \phi 0 + (1 - \phi)n$ where $\phi = \left(1 - \frac{k}{p}\right)^{n-k}$.

To obtain an unbiased estimator $E[X] = n - k$ for this instance we need $(1 - \phi)n \geq n - k$ hence $\phi < k/n$. By the upper bound $\left(1 - \frac{k}{p}\right)^{n-k} < \left(\frac{k}{p}\right)^{n-k}$ we have that there is a constant w s.t. $n > k^{wn}$ implies $\phi > k/n$. Thus when n is exponential in k we have that ϕ is large enough to make any estimator based on the k smallest hash values biased. ■

7. REFERENCES

- [1] Z. Bar-Yossef. *The complexity of massive data set computations*. PhD thesis, University of California at Berkeley, 2002.
- [2] P. Bille, A. Pagh, and R. Pagh. Fast evaluation of union-intersection expressions. In *Proceedings of the 18th International Symposium on Algorithms And Computation (ISAAC '07)*, pages 739–750.
- [3] B. H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Commun. ACM*, 13(7):422–426, 1970.
- [4] M. Braverman, A. Garg, D. Pankratov, and O. Weinstein. Information lower bounds via self-reducibility. In *CSR*, pages 183–194, 2013.
- [5] A. Z. Broder. On the resemblance and containment of documents. In *In Compression and Complexity of Sequences (SEQUENCES)*, pages 21–29, 1997.
- [6] A. Z. Broder, M. Charikar, A. M. Frieze, and M. Mitzenmacher. Min-wise independent permutations. *Journal of Computer and System Sciences*, 60:327–336, 1998.
- [7] A. Z. Broder, S. C. Glassman, M. S. Manasse, and G. Zweig. Syntactic clustering of the web. In *Selected papers from the sixth international conference on World Wide Web*, pages 1157–1166, 1997.
- [8] A. Chakrabarti and O. Regev. An optimal lower bound on the communication complexity of gap-hamming-distance. *SIAM J. Comput.*, 41(5):1299–1317, 2012.

- [9] F. Chierichetti, R. Kumar, S. Lattanzi, M. Mitzenmacher, A. Panconesi, and P. Raghavan. On compressing social networks. In *Proceedings of the 15th ACM SIGKDD*, KDD '09, pages 219–228, 2009.
- [10] E. Cohen and H. Kaplan. In J. R. Douceur, A. G. Greenberg, T. Bonald, and J. Nieh, editors, *SIGMETRICS/Performance*, pages 251–262.
- [11] E. Cohen and H. Kaplan. What you can do with coordinated samples. In *APPROX-RANDOM*, volume 8096 of *Lecture Notes in Computer Science*, pages 452–467. Springer Berlin Heidelberg, 2013.
- [12] E. Cohen, H. Kaplan, and S. Sen. Coordinated weighted sampling for estimating aggregates over multiple weight assignments. *Proc. VLDB Endow.*, 2(1):646–657, Aug. 2009.
- [13] H. Cohen and E. Porat. Fast set intersection and two-patterns matching. In *Proceedings of the 9th Latin American Conference on Theoretical Informatics*, LATIN'10, pages 234–242, Berlin, Heidelberg, 2010. Springer-Verlag.
- [14] D. Dubhashi and D. Ranjan. Balls and bins: A study in negative dependence. *RANDOM STRUCTURES & ALGORITHMS*, 13:99–124, 1996.
- [15] P. Flajolet, E. Fusy, O. Gandouet, and F. Meunier. Hyperloglog: the analysis of a near-optimal cardinality estimation algorithm.
- [16] P. Indyk and D. P. Woodruff. Tight lower bounds for the distinct elements problem. In *Proceedings of Foundations of Computer Science (FOCS)*, pages 283–288, 2003.
- [17] T. S. Jayram. Information complexity: a tutorial. In *PODS*, pages 159–168, 2010.
- [18] D. M. Kane, J. Nelson, and D. P. Woodruff. An optimal algorithm for the distinct elements problem. In *Proceedings of the 29th Symposium on Principles of Database Systems (PODS)*, pages 41–52, 2010.
- [19] I. Kremer, N. Nisan, and D. Ron. On randomized one-round communication complexity. *Computational Complexity*, 8(1):21–49, 1999.
- [20] P. Li and A. C. König. Theory and applications of b-bit minwise hashing. *Commun. ACM*, 54(8):101–109, Aug. 2011.
- [21] P. Li, A. C. König, and W. Gui. b-bit minwise hashing for estimating three-way similarities. In *Proceedings of Annual Conference on Neural Information Processing Systems (NIPS)*, pages 1387–1395, 2010.
- [22] P. Li and C. König. b-bit minwise hashing. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 671–680, 2010.
- [23] P. Li, A. Owen, and C.-H. Zhang. One permutation hashing for efficient search and learning. *CoRR*, abs/1208.1259, 2012.
- [24] P. Neumann. Über den Median der Binomial- und Poissonverteilung. *Wissenschaftliche Zeitschrift der Humboldt-Universität zu Berlin. Reihe Mathematik/Naturwissenschaften*, 16:62–64, 1967.
- [25] R. Pagh. Low redundancy in static dictionaries with constant query time. *SIAM Journal of Computing*, 31(2):353–363, 2001.
- [26] F. Rusu and A. Dobra. Sketches for size of join estimation. *ACM Trans. Database Syst.*, 33(3), 2008.
- [27] J. P. Schmidt, A. Siegel, and A. Srinivasan. Chernoff-hoeffding bounds for applications with limited independence. *SIAM J. Discret. Math.*, 8(2):223–250, May 1995.
- [28] A. Siegel. On universal classes of extremely random constant-time hash functions. *SIAM J. Comput.*, 33(3):505–543, Mar. 2004.
- [29] R. R. Sinha and M. Winslett. Multi-resolution bitmap indexes for scientific data. *ACM Trans. Database Syst.*, 32(3):16, 2007.
- [30] M. Thorup. Bottom-k and priority sampling, set similarity and subset sums with minimal independence. *STOC*, 2013.
- [31] Z. Wei and K. Yi. Beyond simple aggregates: indexing for summary queries. In M. Lenzerini and T. Schwentick, editors, *PODS*, pages 117–128. ACM, 2011.
- [32] D. P. Woodruff. Optimal space lower bounds for all frequency moments. In *SODA*, pages 167–175, 2004.
- [33] D. P. Woodruff and Q. Zhang. Tight bounds for distributed functional monitoring. In *STOC*, pages 941–960, 2012.

APPENDIX

A. SPACE OF k -PERMUTATION MIN-WISE SUMMARIES ON SUNFLOWER SETS

Sunflower sets. (Section 2.3). This hard instance gives the upper bound for k -permutation min-wise hashing of Table 1. For m sets $S_1 \dots S_m$ each of size n , let $t = |\cap_i S_i|$ be the intersection size of all sets. Then a sunflower instance has the property $\forall_{i \neq j} |S_i \cap S_j| = t$, i.e., the m sets are disjoint except for the t intersection elements. The union size of such an instance is $|\cup_i S_i| = t + mn - mt = t + m(n - t)$ as there are t elements in the intersection and each of the m sets hold additional $n - t$ elements. It follows that the Jaccard similarity for a sunflower instance is $t/(t + m(n - t))$.

LEMMA 21. *Given m sets of size n with intersection size t . To obtain an $(\epsilon, O(1))$ -estimate of t using k -permutation min-wise hashing one needs to store $O(\frac{mn}{\epsilon^2})$ elements from each set.*

PROOF. The upper bound for k -permutation min-wise hashing of Table 1 is derived as follows. Let $X_1 \dots X_c$ be independent Bernoulli trials where $\Pr[X_i] = J$ and let $X = \sum_{i=1}^c X_i$ and $\mu = E[X] = cJ$. There exists a c for which there is constant probability of the event that the outcome of X is a relative factor ϵ from $E[X]$. This can be bounded applying a Chernoff-Hoeffding bound on X as follows.

$$\Pr[|X - E[X]| \geq (1 + \epsilon)E[X]] = \Pr[|X - cJ| \geq (1 + \epsilon)cJ] \\ = \delta \geq 2e^{(-cJ\epsilon^2)/3}$$

Then isolating c we have $c \geq \frac{3 \log(2/\delta)}{J\epsilon^2}$, which for $\delta = O(1)$ is $O(\frac{1}{J\epsilon^2}) = O(\frac{t+m(n-t)}{t\epsilon^2})$ following from the Jaccard similarity of the sunflower instance above. For $t < n/2$ we have $c = O(\frac{mn}{t\epsilon^2})$, the sample size required for k -permutation min-wise summaries. ■

We conjecture that this bound is tight, by tightness of Chernoff bounds.