Leveraging Discarded Samples for Tighter Estimation of Multiple-Set Aggregates

Edith Cohen
AT&T Labs–Research
Florham Park, NJ 07932, USA
edith@research.att.com

Haim Kaplan
Tel Aviv University
Tel Aviv, Israel
haimk@cs.tau.ac.il

ABSTRACT

Many datasets, including market basket data, text or hypertext documents, and events recorded in different locations or time periods, can be modeled as a collection of sets over a ground set of keys. Common queries over such data, including similarity or association rules are represented as the weight or selectivity of keys that satisfy some selection predicate defined over keys' attributes and memberships in particular sets.

On massive data sets, exact computation of such aggregates can be inefficient or infeasible, and therefore, approximate queries are processed over sketches of the sets. Sketches based on coordinated random samples are scalable and flexible and well suited for many applications. Queries are resolved by producing a sketch of the union of the sets used in the predicate, from the sketches of these sets, and then applying an estimator to this union-sketch.

We derive novel tighter (unbiased) estimators that leverage sampled keys that are present in the union of applicable sketches but excluded from the union sketch. We establish analytically that our estimators dominate estimators applied to the union-sketch for *all queries and data sets*. Empirical evaluation on synthetic and real data reveals that on typical applications we can expect a 25%-75% reduction in estimation error.

Categories and Subject Descriptors: G.3: probabilistic algorithms; H.2 database management
General Terms: Algorithms, Measurement, Performance

1. INTRODUCTION

We consider datasets modeled as a collection $\mathcal{S}$ of (possibly intersecting) *sets*, defined over a ground set I of (possibly weighted) *keys*. A classic example is documents over features or terms, according to presence in the document.

Basic aggregates over such data are *weight* and *selectivity* of subpopulations of keys. A query specifies a *subpopulation* of I by a selection predicate. The weight aggregate is the sum of the weights of the keys that satisfy the predicate. If keys have uniform weights, the weight aggregate is known as DV (distinct values) count. An example of a weight query is the number of terms present both in document A and in document B and are at least 5 characters long. Selectivity queries are defined with respect to some (sub) collection of sets: The result is the ratio of the sum of the weights of all keys in the union of these sets for which the predicate holds and the total weight of the union of these sets. An important selectivity aggregate is the *Jaccard coefficient* of A and B defined as $|A \cap B|/|A \cup B|$, which measures the similarity between A and B. A common technique to enhance this similarity metric is to assign larger weights to features/terms that are less frequent in the corpus. For weighted keys, the Jaccard coefficient generalizes to $w(A \cap B)/w(A \cup B)$ (the ratio of the weight of the intersection and the weight of the union). Approximate weight aggregates are used to compute more complex (approximate) aggregates, such as variance [15] of a subpopulation of keys or ratio of the weights of two subpopulations of keys.

The selection predicates that specify subpopulations are defined using conditions on keys' attributes *and* memberships in the different sets. We distinguish between *attribute-based* conditions, that are based on properties available through the identifier of the key (length, origin, or frequency of a term) and *membership-based* conditions that are based on the key's set memberships. For example, terms common to two documents A, B are specified using the predicate with membership-based conditions "in A and in B". The predicate "in A and not in B and length ≥ 5" has both attribute-based (length of a term) and membership-based conditions.

We list additional example datasets that fall in this framework.

• **Sensor nodes recording daily vehicle traffic in different locations in a city:** Keys are distinct vehicles (license plate numbers) and sets are location-date pairs (all vehicles observed at that location that date). Example queries with membership-based conditions are: "number of distinct vehicles which operated in Manhattan on election day, 2008" (size of the union of all Manhattan locations in election day). Queries can be restricted to particular classes of vehicles (e.g., taxi cubs or heavy trucks) by adding attribute-based conditions.

• **Market-basket dataset:** Keys are goods, each with an associated marketing cost (these are the weights). Each customer (basket) defines a set which is the set of goods she purchased. Example queries are "the total marketing cost of baby products purchased by male customers from Union county." This predicate has attribute-based condition

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. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SIGMETRICS/Performance'09, June 15–19, 2009, Seattle, WA, USA.
Copyright 2009 ACM 978-1-60558-511-6/09/06 ...$5.00.

(product type) and membership-based conditions (specification of the customer segment as a union of sets).

• **"Inverted" market-basket dataset:** Keys are baskets (customers) and sets are goods (all baskets containing that particular good). A query that asks "what is the likelihood that a certain item is purchased given that another item is purchased" (this is an "association rule" [1, 39]) can be expressed using a predicate with membership-based conditions. If A is the set of customers purchasing, say beer, and B is the set of customers purchasing diapers then the selectivity of $A \cap B$ with respect to B is just the likelihood that a person purchases beer given that she/he purchased diapers. This query can be narrowed down to a particular customer segment (e.g., by zip code or gender) if we add an attribute-based conditions to the predicate.

• **Hyperlinked documents**: Sets and keys are documents, where the set of document A includes all documents with hyperlinks to document A. Documents may be weighted by access data or page rank. Example queries are "the total weight of documents referencing at least 5 out of the 10 documents in Q." This predicate has membership-based conditions.

• **P2P network:** Keys are files and sets are all neighborhoods of all peers (sets of files shared by peers in that neighborhood). Example queries are "the weight of files stored in the 5-hop neighborhoods of peer A or peer B," or "number of distinct files in a particular subject in the 3-hop neighborhood of peer A." Such queries can be used to keep the search focused on peers that contains many files in a particular topic or peers that are more similar to the querying peer [14, 47, 50].

Exact computation of such queries requires retrieving the full content of all sets relevant to the predicate, computing the union, and applying the predicate to all keys in the union, adding up the weights of keys that satisfy the predicate. On massive or distributed data, the high cost of exact computation prohibits running a large number of queries (that is required for clustering or association rule mining). In some cases, such as network traffic data, the full data set may no longer be available at the time the query is formulated. The practical solution is to produce a summary that supports approximate processing of such queries. A suitable format is a set of sketches, one for each set.

Sample-based sketches, where the sketch of a set is a random sample of keys with some auxiliary information, are a popular choice due to scalability and flexibility. A preferred weighted sampling design is *bottom-k* (order) sampling [42, 12, 43, 7, 39, 46, 17, 24, 19, 2, 29]. The sample is obtained by assigning a random rank value to each key, depending on its weight, and including the k keys with smallest rank values. With appropriate rank distributions, bottom-k samples generalize successive weighted sampling without replacement [42, 30] where keys are successively drawn proportionally to their weight and *priority sampling (sequential poisson sampling)* [41, 43, 24]. The sketch of a set supports tight unbiased estimators for weight and selectivity aggregates over the set [19, 17, 24].

Multiple-set aggregates are estimated using the *union-sketch reduction* to estimators over a single-set. The reduction applies when sketches of different sets are *coordinated*, that is, the same set of rank values of keys is used for all sets. It is known that without coordination (independently sampling each set), it is not possible to obtain strong estimators [9].

For a multiple-set aggregate and selection predicate with relevant sets $\mathcal{S} \in \mathcal{A}$, a size-$k$ sketch of the union $\bigcup_{A \in \mathcal{S}} A$ is constructed from size-k sketches of the sets in $\mathcal{S}$ [12, 7, 6]. A "single set" weight or selectivity estimator can then be applied to estimate our multiple-set aggregate, by applying it to the union sketch of $\mathcal{S}$.

Coordinated bottom-k sketches can be computed efficiently for diverse data sources including centralized or distributed with *explicitly* or *implicitly* represented sets [17]. Sets are explicitly represented when the data source can be modeled as a list of keys for each set or a list of sets for each key (the inverted data). In the former case, random hash functions are used to decouple the sampling of different sets [7, 8, 23, 6, 39, 2]. Examples of explicitly-represented sets includes item-basket associations in a market basket data, links in web pages, and features in documents [7, 3, 39, 46, 2].

Sets are implicitly represented when memberships are specified indirectly (as in our p2p example) through some metric on a set of points. Implicit representation can be more concise than the corresponding explicit representation. Keys are associated with points and sets are specified by a point and distance pair (*neighborhood*) and include all keys within that distance from the point [12, 21, 20, 38, 33, 16]. Examples are nodes in a graph with the shortest path or reachability metric, the Euclidean plane, or time stamps or sequence numbers on a data stream [12, 20, 16, 33]. When we are interested in multiple distances (neighborhoods) of a point (applications include aggregates with time or spatial decay [20, 16]), *all-distances sketches* succinctly represent coordinated sketches of *all* neighborhoods of the point and can be computed efficiently over the implicit representation of the dataset [12, 16, 17].

Our contributions. Our main contribution is the derivation of tighter unbiased estimators for multiple-sets weight and selectivity aggregates. These estimators are applicable to a set of coordinated sketches and therefore apply to the *same* set of sketches as the union-sketch method. We will show that they involve *similar computational tasks* as the union-sketch method and they dominate all previous methods in terms of estimation quality.

Combinations of sketches. A close look at the union-sketch approach reveals that we discard potentially useful information present in the *union of the size-k sketches of the sets* by restricting our attention to the *size-k sketch of the union of these sets*. If there are t relevant sets, the union of the sketches includes at least k but up to $t*k$ distinct keys. In Section 3 we consider two more inclusive *combinations* of the sketches of the sets than the union-sketch: The *short combination of sketches* (SCS), and the *long combination of sketches* (LCS). The LCS includes all keys in the union of the sketches, it contains the SCS, which contains the k keys in the sketch of the union.

Combination RC estimators for weight aggregates. In Section 4 we develop unbiased estimators for subpopulation weight that leverage the additional keys contained in the LCS and SCS. Fully exploiting this additional information was a subtle and challenging task: The SCS can be viewed as a variable-size sequential sample of the union where the number of included keys depends on set memberships of previously selected keys. The LCS can not be expressed as a sequential weighted sample of a set. The challenge lies in bene-

fiting from additional keys without introducing bias – we can not simply apply a single-sketch estimator to combinations.

We build on the powerful Rank Conditioning (RC) estimators that are the best known estimators applicable to a sketch of a single set [19, 24].[1] *Adjusted weights* are assigned to sampled keys and the weight estimate of a subpopulation is the sum of the adjusted weights of sampled keys that belong to the subpopulation. For multiple-set aggregates, our combination RC estimators assign positive adjusted weights to all keys in the combination whereas the basic union-sketch method assigns them only to k keys. We prove that our estimators are unbiased for every subpopulation and furthermore, the covariance of the adjusted weights of any two different keys is zero. This guarantees that the variance of our estimate for a subpopulation is not larger than the sum of the variances of the adjusted weights of the keys in the subpopulation.

We prove that (for *any* selection predicate and data set) the SCS RC estimators are at least as tight (at most the variance) as the union sketch RC estimators. Similarly to union-sketch estimators, the SCS estimators are applicable to general *select* predicates. The LCS RC estimators are at least as tight as the SCS RC estimators but are applicable to a more limited class of predicates that are attribute-based selections from a union of sets. Therefore, our SCS RC estimator strictly dominates all union-sketch based estimators, and for applicable *select* predicates, LCS RC dominates all other methods. In Section 5 we demonstrate how the different estimators are applied.

Selectivity estimators. Other estimators, such as maximum likelihood and selectivity estimators, that are traditionally applied to the union-sketch can be extended to yield tighter results on combinations. Section 6 outlines the derivation of SCS unbiased selectivity estimators that strictly improve over traditional unbiased estimators for Jaccard similarity [7, 6].

Empirical evaluation. Section 7 summarizes results of extensive experiments on real and synthetic data. We quantify the power of SCS and LCS-based estimators compared to estimators applied to the union-sketch. Synthetic data was designed to study how performance depends on the relations between the sets and on the number of sets used in the predicate. Real data allowed us to use natural selection predicates and demonstrate potential applications. We discuss related work in Section 8 and conclude in Section 9.

2. PRELIMINARIES

This section provides necessary background and definitions.

A *weighted set* (I, w) consists of a set of keys I and a weight function w assigning a $w(i) \geq 0$ to each key $i \in I$. A *rank assignment* maps each key i to a random rank $r(i)$. The ranks of keys are drawn independently using a family of distributions $\mathbf{f_w}$, where the rank of a key with weight $w(i)$ is drawn according to $\mathbf{f_{w(i)}}$. For a set J and a rank assignment r we denote by $r_i(J)$ the ith smallest rank of a key in J, we also abbreviate and write $r(J) = r_1(J)$. Random rank assignments are used to obtain *sketches* (samples with some auxiliary information) of sets as follows.

[1]There are tighter estimators when the exact total weight of the set is known, but this is not the case in our multiple-set aggregates since the weight of the union of sets can not be exactly recovered from sketches of the sets.

The *k-mins sketch* [12, 7] of a set J is produced from k independent rank assignments, $r^{(1)}, \ldots, r^{(k)}$. The sketch of a set J is the k-vector $(r^{(1)}(J), r^{(2)}(J), \ldots, r^{(k)}(J))$. Depending on the application we may store with each of these ranks, attributes associated with the corresponding key.

A *bottom-k sketch* (or *order sample*) [42, 12, 43, 7, 39, 46, 17, 19, 2, 29] of a set J is defined based on a single rank assignment r as follows. Let $i_1, \ldots, i_k$ be the k keys of smallest ranks in J. The sketch consists of k pairs $(r(i_j), w(i_j))$, $j = 1, \ldots, k$, and $r_{k+1}(J)$. (If $|J| \leq k$ we store only $|J|$ pairs.) We denote a bottom-k sketch of a set A with respect to a rank assignment r by $s_k(A, r)$.

Consider a set $\mathcal{A}$ of sets over a set of keys I. *Coordinated* k-mins or bottom-k sketches are obtained by using the same rank assignment over I (for k-mins sketches, same set of rank assignments), when producing the sketches of all sets $A \in \mathcal{A}$. Coordinated sketches should include all rank values and keys' weights.

The union-sketch. Coordinated bottom-k and k-mins sketches have the property that for a set $\mathcal{S} \subset \mathcal{A}$ of sets we can compute the sketch of $\bigcup_{A \in \mathcal{S}} A$ from the sketches of the sets $A \in \mathcal{S}$. For k-mins sketches, the sketch of the union contains, for each rank function the key with minimum rank value across sets in $\mathcal{S}$. For bottom-k sketches, the keys in $s_k(\bigcup_{A \in \mathcal{S}} A, r)$ are the keys with k smallest ranks in $\bigcup_{A \in \mathcal{S}} s_k(A, r)$. Note that $r_{k+1}(\bigcup_{A \in \mathcal{S}} A)$ is the minimum rank of a key that is among the $(k+1)$ smallest ranks in at least one of $A \in \mathcal{S}$ but is not among the k smallest ranks in the union sketch. Therefore, $r_{k+1}(\bigcup_{A \in \mathcal{S}} A)$ can also be determined from the sketches of $A \in \mathcal{S}$.

The union-sketch reduction is a method that allows us to apply a weight/selectivity estimator designed for attribute-based *select* predicates over a single (k-mins or bottom-k) sketch to estimate the weight/selectivity of a subpopulation specified by a general *select* predicate (with membership and attribute based conditions) over coordinated (k-mins or bottom-k) sketches of a collection of sets $\mathcal{S}$.

We first identify all sets $\mathcal{S}$ relevant to the predicate. We retrieve the sketches of $\mathcal{S}$ and compute the sketch of the union. A very handy property of the union-sketch is that for each key x included in the sketch of the union we can determine which sets of $\mathcal{S}$ it is a member of. We therefore can treat each membership in a set in $\mathcal{S}$ as an attribute of the keys. We then apply our single-sketch estimator to the union-sketch of $\mathcal{S}$, treating membership-based conditions of the predicate as attribute-based conditions over the keys in the union-sketch.

As a concrete example, consider the inverted market-basket data set and the query "the number of baskets of at most 10 keys that contain beer or wine and cheese." To do so, we isolate the sketches of beer, wine, and cheese, and compute the union sketch. The union sketch is a random sample from the set of baskets that have beer, wine, or cheese. For each basket in the union we know if it has or does not have each one of the three goods. The size of the basket is an attribute. We can therefore identify all baskets in the sample for which the predicate "has beer or has wine and has cheese and has size ≤ 10" holds and estimate the distinct count.

WS **sketches.** The choice of which family of random rank functions to use matters only when keys are weighted. Otherwise, sketches produced using one rank function can be transformed to any other rank function. Rank functions $\mathbf{f_w}$ with some convenient properties are exponential distribu-

• **Keys:**

key	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}
w_j	1	2	1	3	1	1	1	1	1	1
u_j	0.487	0.72	0.3	0.832	0.765	0.599	0.131	0.886	0.73	0.341
$r_j = \frac{u_j}{w_j}$	0.487	0.36	0.3	0.208	0.765	0.599	0.131	0.886	0.73	0.341

• **Sets:**

$A_1 = \{i_1, i_3, i_5, i_7, i_9\}$
$A_2 = \{i_1, i_2, i_5, i_6, i_9, i_{10}\}$
$A_3 = \{i_3, i_4, i_5, i_6, i_7\}$
$A_4 = \{i_2, i_4, i_6, i_8, i_{10}\}$

• Keys sorted by increasing ranks (with matrix showing set memeberships of all keys):

keys	i_7 (0.131)	i_4 (0.208)	i_2 (0.36)	i_3 (0.3)	i_{10} (0.341)	i_1 (0.487)	i_6 (0.599)	i_9 (0.73)	i_5 (0.765)	i_8 (0.886)
A_1	$\vee$	$\times$	$\times$	$\vee$	$\times$	$\vee$	$\times$	$\vee$	$\vee$	$\times$
A_2	$\times$	$\times$	$\vee$	$\times$	$\vee$	$\times$	$\vee$	$\vee$	$\vee$	$\times$
A_3	$\vee$	$\vee$	$\times$	$\vee$	$\times$	$\times$	$\vee$	$\times$	$\vee$	$\times$
A_4	$\times$	$\vee$	$\vee$	$\times$	$\vee$	$\times$	$\vee$	$\times$	$\times$	$\vee$

• Table showing bottom-3 sketches $s_3(A_i, r)$, (3 least-ranked keys of A_i and $r_4(A_i)$, the 4^{th}-smallest rank value):

A	$s_3(A, r)$	$r_4(A)$
A_1	i_7 (0.131), i_3 (0.3), i_1 (0.487)	0.73
A_2	i_2 (0.36), i_{10} (0.341), i_6 (0.599)	0.73
A_3	i_7 (0.131), i_4 (0.208), i_3 (0.3)	0.599
A_4	i_4 (0.208), i_2 (0.36), i_{10} (0.341)	0.599

• For $\mathcal{S} = \{A_1, A_2\}$ and $\mathcal{S} = \{A_1, A_2, A_3, A_4\}$, keys included in:
◇ $s_3(\bigcup_{A\in\mathcal{S}} A, r)$ (The union-sketch of $\mathcal{S}$)
◇ $\text{SCS}_3(\mathcal{S}, r)$ (contains all keys in $\bigcup_{A_i\in\mathcal{S}} s_3(A_i, r)$ that have rank value below $r_4(\mathcal{S}) = \min_{A_i\in\mathcal{S}} r_4(A_i)$), and
◇ $\text{LCS}_3(\mathcal{S})$ (contains all keys in $\bigcup_{A_i\in\mathcal{S}} s_3(A_i, r)$).

S	combination type	content	# keys
A_1, A_2	$s_3\left(\bigcup_{A\in\mathcal{S}} A, r\right)$	i_7 (0.131), i_2 (0.36), i_3 (0.3)	3
A_1, A_2	$\text{SCS}_3(\mathcal{S}, r)$	i_7 (0.131), i_2 (0.36), i_3 (0.3), i_{10} (0.341), i_1 (0.487), i_6 (0.599)	6
A_1, A_2	$\text{LCS}_3(\mathcal{S}, r)$	i_7 (0.131), i_2 (0.36), i_3 (0.3), i_{10} (0.341), i_1 (0.487), i_6 (0.599)	6
A_1, A_2, A_3, A_4	$s_3\left(\bigcup_{A\in\mathcal{S}} A, r\right)$	i_7 (0.131), i_4 (0.208), i_2 (0.36)	3
A_1, A_2, A_3, A_4	$\text{SCS}_3(\mathcal{S}, r)$	i_7 (0.131), i_4 (0.208), i_2 (0.36), i_3 (0.3), i_{10} (0.341), i_1 (0.487)	6
A_1, A_2, A_3, A_4	$\text{LCS}_3(\mathcal{S}, r)$	i_7 (0.131), i_4 (0.208), i_2 (0.36), i_3 (0.3), i_{10} (0.341), i_1 (0.487), i_6 (0.599)	7

Figure 1: Example shows a set I of 10 keys $i_1, \ldots, i_{10}$ with respective weights $w_1, \ldots, w_{10}$ and 4 subsets $A_1, \ldots, A_4$; a random rank assignment r for I, using priority ranks (for each key i_j, draw $u_j \in U[0, 1]$ and compute rank value $r_j = u_j/w_j$); bottom-3 sketches $s_3(A_i, r)$ for $i = 1, \ldots, 4$; for $\mathcal{S} = \{A_1, A_2\}$ and $\mathcal{S} = \{A_1, A_2, A_3, A_4\}$, keys included in the union-sketch, the SCS, and the LCS of S.

tions with parameter w [42, 30, 12]. The density function is $\mathbf{f_w(x)} = \mathbf{we^{-wx}}$, and its cumulative distribution function is $\mathbf{F_w(x)} = \mathbf{1 - e^{-wx}}$. Equivalently, if $u \in U[0, 1]$ then $-\ln(u)/w$ is an exponential random variable with parameter w. A useful property used in [12, 16, 17, 19] is that the minimum rank $r(J) = \min_{i\in J} r(i)$ of a key in a set $J \subset I$ is exponentially distributed with parameter $w(J) = \sum_{i\in J} w(i)$.

Moreover, the probability that a key $x \in J$ is the minimum rank key is $w(x)/w(J)$. Hence, a k-mins sketch of a set J is a weighted random sample of size k, drawn **with replacement** from J. We call a k-mins sketch using exponential ranks a WSR sketch. On the other hand, a bottom-k sketch of a set J with exponential ranks corresponds to a weighted k-sample drawn **without replacement** from J [42, 30]. We call such a sketch a WS sketch.

PRI **sketches.** If the rank value of a key with weight w is selected uniformly at random from $[0, 1/w]$ then the bottom-k sketch is a *priority sketch* (also known as *Sequential Poisson Sample*) [41, 43, 24]. This is the equivalent to choosing rank value u/w, where $u \in U[0, 1]$ or using density function $\mathbf{f_w(x)} = \mathbf{w}$ for $0 \le x \le 1/w$ and $\mathbf{f_w(x)} = \mathbf{0}$ otherwise and cumulative distribution $\mathbf{F_w(x)} = \min\{\mathbf{1}, \mathbf{wx}\}$. Estimators for PRI sketches [24] have (nearly) minimum sum of per-key variances $\sum_{i\in I} \text{VAR}(a(i))$ [48].

Adjusted weights. As mentioned in the introduction one technique to obtain estimators for the weights of keys is by assigning an adjusted weight $a(i) \ge 0$ to each key i in the sample (adjusted weight $a(i) = 0$ is implicitly assigned to keys not in the sample). The adjusted weights are assigned such that $\text{E}[a(i)] = w(i)$, where the expectation is over the randomized algorithm choosing the sample. Using adjusted weights we can estimate the weight of any subpopulation $J \subset I$ by $\sum_{j\in J} a(j) = \sum_{j\in J|a(j)>0} a(j)$. The estimate is easily computed from the sample assuming we have sufficient auxiliary information to tell for each key in the sample whether it belongs to J or not. Moreover, for any numeric function $h()$ over keys' attributes such that $h(i) > 0$ only if $w(i) > 0$ and any subpopulation J, $\sum_{j\in J|a(j)>0} a(j)h(j)/w(j)$ is an unbiased estimate of $\sum_{j\in J} h(j)$.

Horvitz-Thompson (HT). Let Ω be the distribution over sketches. If we know $p^{(\Omega)}(i) = \Pr\{i \in s | s \in \Omega\}$ for every $i \in s$ then we can assign to $i \in s$ the adjusted weight

$$a(i) = \frac{w(i)}{p^{(\Omega)}(i)} .$$

Since $a(i)$ is 0 when $i \notin s$, it is easy to see that $\text{E}[a(i)] = w(i)$. The estimator based on these adjusted weights is called the Horvitz-Thompson (HT) estimator [32]. It is well

known and easy to see that these adjusted weights are unbiased and have minimal variance *for each key* for the particular distribution Ω over rank assignments.

HT **on a partitioned sample space (HTp).** This is a method to derive adjusted weights when we cannot determine $\Pr\{i \in s | s \in \Omega\}$ from the information contained in the sketch s alone. For example, if s is a bottom-k sketch of (I, w), then $\Pr\{i \in s | s \in \Omega\}$ generally depends on all the weights $w(i)$ for $i \in I$ and cannot be determined from s.

For each key i we consider a partition of Ω into equivalence classes. For a sketch s, let $P^i(s) \subset \Omega$ be the equivalence class of s. This partition must satisfy the following requirement: Given s such that $i \in s$, we can compute the conditional probability $p^i(s) = \Pr\{i \in s' \mid s' \in P^i(s)\}$ from the information included in s.

We can therefore compute for all $i \in s$ the assignment $a(i) = w(i)/p^i(s)$ (implicitly, $a(i) = 0$ for $i \notin s$.) It is easy to see that within each equivalence class, $\mathrm{E}[a(i)] = w(i)$. Therefore, also over Ω we have $\mathrm{E}[a(i)] = w(i)$.

The variance of the adjusted weight $a(i)$ obtained using HTp depends on the particular partition in the following way. (This follows from the convexity of the variance.)

Lemma 1. *[19] Consider two partitions of the sample space, such that one partition is a refinement of the other. Then the variance of $a(i)$ using* HTp *with the coarser partition is at most that of the* HTp *with the finer partition.*

Rank Conditioning (RC) adjusted weights. This is an HTp estimator for a single bottom-k sketch [19]. The partition of Ω which we use for assigning an adjusted weight to i is based on *rank conditioning*: For each possible rank value τ we have an equivalence class P^i_τ containing all sketches in which the kth smallest rank value assigned to a key other than i is τ. Note that if $i \in s$ then this is the $(k+1)$st smallest rank which is included in the sketch. It is easy to see that the inclusion probability of i in a sketch in P^i_τ is $p^i_\tau = \mathbf{F}_{w(i)}(\tau)$.

Assume s contains $i_1, \ldots, i_k$ and the $(k+1)$st smallest rank value r_{k+1}. Then for key i_j, we have $s \in P^{ij}_{r_{k+1}}$ and $a(i_j) = \frac{w(i_j)}{\mathbf{F}_{w(i_j)}(r_{k+1})}$.

3. COMBINING BOTTOM-K SKETCHES

Consider a weighted set I, a set $\mathcal{S}$ of subsets of I, a family of rank functions $\mathbf{F}_w$ $(w > 0)$, and a set of coordinated bottom-k sketches $s_k(A, r)$ for $A \in \mathcal{S}$, where r is drawn according to $\mathbf{F}_w$ $(w > 0)$.

The *short combination of sketches* (SCS) of $\mathcal{S}$, denoted $\mathrm{SCS}_k(\mathcal{S}, r)$, contains the prefixes of the sketches $s_k(A, r)$ $(A \in \mathcal{S})$ that include all keys with rank values smaller than $r_{k+1}(\mathcal{S}) = \min_{A \in \mathcal{S}} r_{k+1}(A)$. The SCS also includes the value $r_{k+1}(\mathcal{S})$. The SCS contains between k and $|\mathcal{S}|k$ keys. Its size depends on the rank assignment. Its expected size is larger when sets are of similar weights and have fewer common keys.

The $\ell \geq k$ keys in the SCS are the ℓ least-ranked keys in the union $\bigcup_{A \in \mathcal{S}} A$ and $r_{k+1}(\mathcal{S}) = r_{\ell+1}(\bigcup_{A \in \mathcal{S}} A)$. Moreover, ℓ is *maximal* for which we can identify the ℓ least-ranked keys in the union from information available in the sketches of $\mathcal{S}$. For WS sketches, the SCS can be viewed as the outcome of weighted sampling without replacement (ppswor) from the union of the sets $\mathcal{S}$ until we obtain k distinct samples from at least one of the sets in $\mathcal{S}$.

An important property of the SCS is that for every key x in $\mathrm{SCS}_k(\mathcal{S}, r)$ and a set $A \in \mathcal{S}$ we can determine if $x \in A$: Indeed $x \in A$ if and only if x is in $s_k(A, r)$. The SCS is the maximal set of keys that are included in the union of the sketches and have this property.

The *long combination of sketches* (LCS) of $\mathcal{S}$, denoted $\mathrm{LCS}_k(\mathcal{S}, r)$, includes all the information in the sketches $s_k(A, r)$, $A \in \mathcal{S}$.

The LCS includes the SCS, but we do not have complete set-membership information for all its keys. These definitions and relations are illustrated in Figure 1 through a detailed example of 4 sets defined over a ground set of 10 keys. The example demonstrates that the SCS and LCS contain more keys than the union-sketch.

In the sequel we derive estimators that reflect this relationship between combinations: SCS estimators are tighter than union-sketch estimators, reflecting the fact that the SCS contains the union-sketch. They are both applicable to arbitrary *select* predicates, reflecting the full membership information that is available for each included key. LCS based estimators are *tighter* than SCS based estimators, reflecting the fact that the LCS contains the SCS but LCS based estimators are *more limited* in that they are applicable only to restricted *select* predicates, reflecting the fact that we have less information for included keys.

• **union-sketch** RC **adjusted weights for** $i_j \in s_3(\bigcup_{A \in \mathcal{S}} A, r)$**:**
$\tau = r_4(\bigcup_{A \in \mathcal{S}} A)$, $a^{(\mathrm{union})}(i_j) = w_j / p(w_j, \tau)$

$\mathcal{S}$	τ	$p(w, \tau)$	$a^{(\mathrm{union})}(i_j)$
A_1, A_2	0.341	$\min\{0.341w, 1\}$	$\max\{w_j, 2.93\}$
A_1, A_2, A_3, A_4	0.3	$\min\{0.3w, 1\}$	$\max\{w_j, 3.33\}$

• SCS RC **adjusted weights for** $i_j \in \mathrm{SCS}_3(\mathcal{S}, r)$**:**
$\tau = r_4(\mathcal{S})$, $a^{(\mathrm{SCS})}(i_j) = w_j / p(w_j, \tau)$

$\mathcal{S}$	$r_4(\mathcal{S})$	$p(w, r_4(\mathcal{S}))$	$a^{(\mathrm{SCS})}(i_j)$
A_1, A_2	0.73	$\min\{0.73w, 1\}$	$\max\{w_j, 1.37\}$
A_1, A_2, A_3, A_4	0.599	$\min\{0.599w, 1\}$	$\max\{w_j, 1.67\}$

• LCS RC **adjusted weights for** $i_j \in \mathrm{LCS}_3(\mathcal{S}, r)$**,** $\mathcal{S} = \{A_1, A_2, A_3, A_4\}$**.**
Sets sorted by increasing $r_4(A_i)$: A_3, A_4, A_1, A_2

i_j	i_7	i_4	i_2	i_3	i_{10}	i_1	i_6
$f(\mathcal{S}, r, i_j)$	1	4	2	1	2	1	2
$\tau(\mathcal{S}, r, i_j)$	0.73	0.599	0.73	0.73	0.73	0.73	0.73
$a^{(\mathrm{LCS})}(i_j)$	1.37	3	2	1.37	1.37	1.37	1.37

Union-sketch, SCS/LCS RC adjusted weights for $\mathcal{S} = \{A_1, A_2\}$:

key	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}
w_j	1	2	1	3	1	1	1	1	1	1
union	0	2.93	2.93	0	0	0	2.93	0	0	0
SCS/LCS	1.37	2	1.37	0	0	1.37	1.37	0	0	1.37

Union-sketch, SCS, and LCS RC adjusted weights for $\mathcal{S} = \{A_1, A_2, A_3, A_4\}$:

key	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}
w_j	1	2	1	3	1	1	1	1	1	1
union	0	3.33	0	3.33	0	0	3.33	0	0	0
SCS	1.67	2	1.67	3	0	0	1.67	0	0	1.67
LCS	1.37	2	1.37	3	0	1.37	1.37	0	0	1.37

Figure 2: Upper box: Adjusted weights computation for example in Figure 1. SCS and LCS-adjusted weights for $\mathcal{S} = \{A_1, A_2\}$ are equal since $r_4(A_1) = r_4(A_2) = r_4(\{A_1, A_2\})$. Lower two tables: RC adjusted weights computed using the union-sketch, the SCS and the LCS.

4. COMBINATION RC ESTIMATORS

We derive RC estimators for $\mathrm{SCS}_k(\mathcal{S}, r)$ and $\mathrm{LCS}_k(\mathcal{S}, r)$. Our RC estimators assign adjusted weights that are positive for all keys included in the respective combination (other

	condition	relevant sets $\mathcal{S}$	keys	weight	RC union	RC SCS	RC LCS	best comb
P_1	$i_j \in \bigcup_{i\in[2]} A_i \wedge (j<8 \vee j \geq 4)$	A_1, A_2	i_5, i_6, i_7	3	2.93	2.74	2.74	LCS_3
P_2	$i_j \in \bigcap_{i\in[2]} A_i \wedge (j<8 \vee j \geq 4)$	A_1, A_2	i_5	1	0	1.37	$---$	SCS_3
P_3	$i_j \in$ at least two out of$A_1, \ldots, A_4$)	A_1, A_2, A_3, A_4	$i_1, \ldots, i_7, i_9, i_{10}$	12	10	11.68	$--$	SCS_3
P_4	$(i_j \in \bigcup_{i\in[4]} A_i \wedge j$ is odd	A_1, A_2, A_3, A_4	i_1, i_3, i_5, i_7, i_9	5	3.33	5	4.1	LCS_3

Figure 3: Example predicates for the dataset in Figure 1. Table shows for each predicate P, a minimum set of relevant sets, all keys that satisfy $P(i)$, weight of these keys, best applicable combination, and RC union, RC SCS, and RC LCS estimates, based on adjusted weights computation in Figure 2. (LCS adjusted weight is not shown for predicates where LCS is not applicable).

keys are implicitly assigned adjusted weight of zero), are unbiased for all keys in $U = \bigcup_{A\in\mathcal{S}} A$, and have zero covariances.

Let $p(w,\tau) \equiv \lim_{x\to\tau^-} \mathbf{F}_w(x)$ be the probability than a key with weight w obtains rank value that is smaller than τ.

SCS RC **adjusted weights** $a^{(\text{SCS})}(i)$**:**

- $r_{k+1}(\mathcal{S}) \leftarrow \min_{A\in\mathcal{S}} r_{k+1}(A)$.
- $\text{SCS}_k(\mathcal{S}, r) \leftarrow \{i \in \bigcup_{A\in\mathcal{S}} s_k(A,r) \,|\, r(i) < r_{k+1}(\mathcal{S})\}$
- for all $i \in \text{SCS}_k(\mathcal{S}, r)$, assigned the adjusted weight

$$a^{(\text{SCS})}(i) \leftarrow \frac{w(i)}{p(w(i), r_{k+1}(\mathcal{S}))} \ . \tag{1}$$

(For WS sketches, $a^{(\text{SCS})}(i) = w(i)/(1-\exp(-w(i)r_{k+1}(\mathcal{S})))$, and for PRI sketches $a^{(\text{SCS})}(i) = \max\{w(i), 1/r_{k+1}(\mathcal{S})\}$). Figure 2 demonstrates the computation of SCS RC adjusted weights and (RC adjusted weights for the union sketch.

We show that $a^{(\text{SCS})}$ are unbiased:

LEMMA 2. *For all $i \in U$, $\mathrm{E}[a^{(\text{SCS})}(i)] = w(i)$.*

PROOF. We apply HTP. For a key i we partition the space of all rank assignments according to the rank values assigned to the keys $U \setminus \{i\}$. Consider a subspace R in this partition. Fix some $r \in R$ and let

$$\tau(R) = \min\{ \quad \min\{r_k(A\setminus\{i\}) \mid A \in \mathcal{S}, i \in A\} \\ \min\{r_{k+1}(A) \mid A \in \mathcal{S}, i \notin A\} \quad \} \ .$$

Clearly $\tau(R)$ is independent of the choice of $r \in R$.

For $r \in R$, the key i is included in $\text{SCS}_k(\mathcal{S}, r)$ if and only if $r(i) < \tau(R)$, which happens with probability $p(w(i), \tau(R))$. If indeed i is included in $\text{SCS}_k(\mathcal{S}, r)$ then $r_{k+1}(\mathcal{S}) = \tau(R)$. □

We show that $a^{(\text{SCS})}$ have zero covariances:

LEMMA 3. *For $i, j \in U$, $i \neq j$, $\text{COV}[a^{(\text{SCS})}(i), a^{(\text{SCS})}(j)] = 0$.*

PROOF. We partition the space of rank assignments and show that in each set of the partition, $\mathrm{E}[a(i)a(j)] = w(i)w(j)$. The partition is according to the rank values assigned to all keys in $U \setminus \{i,j\}$. Let R be a subspace in the partition, and let r be a rank assignment in R. Define

$$\tau(R) = \min\{ \quad \min\{r_{k-1}(A\setminus\{i,j\}) \mid A \in \mathcal{S}, i, j \in A\}, \\ \min\{r_k(A\setminus\{i\}) \mid A \in \mathcal{S}, i \in A, j \notin A\}, \\ \min\{r_k(A\setminus\{j\}) \mid A \in \mathcal{S}, j \in A, i \notin A\}, \\ \min\{r_{k+1}(A) \mid A \in \mathcal{S}, i, j \notin A\} \quad \} \ .$$

Clearly $\tau(R)$ is independent of the choice of $r \in R$. For $r \in R$, it is easy to see that i and j are both included in the SCS if and only if $r(i) < \tau(R)$ and $r(j) < \tau(R)$, which happens with probability $p(w(i), \tau(R))p(w(j), \tau(R))$. Otherwise either i or j is not included in the SCS and $a(i)a(j) = 0$. In the case they are both included, $r_{k+1}(\mathcal{S}) = \tau(R)$, and therefore they are assigned adjusted weights of $w(i)/p(w(i), \tau(R))$ and $w(j)/p(w(j), \tau(R))$, respectively. It follows that

$$\mathrm{E}[a(i)a(j)] = \frac{p(w(i),\tau(R))p(w(j),\tau(R))w(i)w(j)}{p(w(i),\tau(R))p(w(j),\tau(R))} \\ = w(i)w(j) \ . \quad \square$$

LCS RC **adjusted weights** $a^{(\text{LCS})}(i)$ **:**

- Sort the sets $A \in \mathcal{S}$ by increasing $r_{k+1}(A)$ into the ordered set $A_1, A_2, \ldots, A_{|\mathcal{S}|}$ ($r_{k+1}(A_i) \leq r_{k+1}(A_j)$ if $i < j$).
- For all $i \in \text{LCS}_k(\mathcal{S}, r)$:
  $f(\mathcal{S}, r, i) \leftarrow \arg\max_h i \in s_k(A_h, r)$
  $\tau(\mathcal{S}, r, i) \leftarrow r_{k+1}(A_{f(\mathcal{S},r,i)})$.

$$a^{(\text{LCS})}(i) \leftarrow \frac{w(i)}{p(w(i), \tau(\mathcal{S}, r, i))} \ . \tag{2}$$

Figure 2 demonstrates the computation of RC LCS adjusted weights.

LEMMA 4. *For all $i \in U$, $\mathrm{E}[a^{(\text{LCS})}(i)] = w(i)$.*

PROOF. For a key $i \in U$, we partition the space of all rank assignments according to the rank values of keys in $U \setminus \{i\}$. Consider a subspace R in this partition, let r be a rank assignment in R, and let $\tau(R) = \max_{A\in\mathcal{S}|i\in A} r_k(A \setminus \{i\})$, which is independent of the choice of $r \in R$.

For $r \in R$, the key i is included in $\text{LCS}_k(\mathcal{S}, r)$ if and only if $r(i) < \tau(R)$. This happens with probability $p(w(i), \tau(R))$ and when it happens we clearly have that $r_{k+1}(A_{i_{f(\mathcal{S},r,i)}}) = \tau(R)$, which implies the lemma. □

LEMMA 5. *For $i, j \in U$, $i \neq j$, $\text{COV}[a^{(\text{LCS})}(i), a^{(\text{LCS})}(j)] = 0$.*

PROOF. Consider the subspace where all ranks of keys other than i and j are fixed. We compute $E[a(i)a(j)]$ in this subspace.

Let $\mathcal{S}_i$ be the collection of sets in $\mathcal{S}$ that contain key i and do not contain key j. Let $\mathcal{S}_j$ be the collection of sets in $\mathcal{S}$ that contain key j and do not contain key i. Finally let $\mathcal{S}_{i,j}$ be the collection of sets in $\mathcal{S}$ containing both i and j.

Define $r^{-i} = \max(\{r_k(A\setminus\{i\}) \mid A \in \mathcal{S}_i\} \cup \{r_{k-1}(A \setminus \{i,j\}) \mid A \in \mathcal{S}_{i,j}\}$, $r^{-j} = \max(\{r_k(A\setminus\{j\}) \mid A \in \mathcal{S}_j\} \cup \{r_{k-1}(A\setminus\{i,j\}) \mid A \in \mathcal{S}_{i,j}\}$, and $r^{-i,j} = \max(\{r_k(A\setminus\{i,j\}) \mid A \in \mathcal{S}_{i,j}\}$.

We split into cases according to the relations between r^{-i}, r^{-j}, and $r^{-i,j}$. If $r^{-i,j} \leq \min\{r^{-i}, r^{-j}\}$ or if $\max\{r^{-i}, r^{-j}\} \leq r^{-i,j}$, then i and j are both included (and $a(i)a(j) > 0$) if and only if $r(i) < r^{-i}$ and $r(j) < r^{-j}$. In which case $a(i) = \frac{w(i)}{p(w(i),r^{-i})}$ and $a(j) = \frac{w(j)}{p(w(j),r^{-j})}$. Therefore, under this conditioning,

$$\mathrm{E}[a(i)a(j)] = p(w(i), r^{-i})p(w(j), r^{-j})\frac{w(i)}{p(w(i), r^{-i})}\frac{w(j)}{p(w(j), r^{-j})} \\ = w(i)w(j) \ .$$

The remaining case is $r^{-i} < r^{-i,j} < r^{-j}$ (the case $r^{-j} < r^{-i,j} < r^{-i}$ is symmetric). j is included if and only if $r(j) \leq r^{-j}$, in which case $a(j) = \frac{w(j)}{p(w(j),r^{-j})}$. The inclusion condition and adjusted weight of i if included depend on $r(j)$, but if we *fix* $r(j)$, from the proof of Lemma 4, $\mathrm{E}[a(i)] = w(i)$. That is, if $a(i|y,x)$ denotes the adjusted weight of i if

$r(j) = x$ and $r(i) = y$, then for all x, $\int_0^\infty a(i|y,x)dy = w(i)$. Therefore, scriptsize

$$\mathrm{E}[a(i)a(j)] = \int_0^{r^{-j}} \mathbf{f}_{w(j)}(x)\frac{w(j)}{p(w(j),r^{-j})}\int_0^\infty a(i|y,x)dydx$$

$$= \int_0^{r^{-j}} \mathbf{f}_{w(j)}(x)dx\frac{w(j)w(i)}{p(w(j),r^{-j})} = p(w(j),r^{-j})\frac{w(j)w(i)}{p(w(j),r^{-j})}$$

$$= w(j)w(i) \quad \square$$

Consider the set $\mathcal{S}$ of subsets of I, a family of rank functions, and coordinated bottom-k sketches $s_k(A,r)$ for $A \in \mathcal{S}$. We compare the three RC adjusted weight assignments $a^{(C)}(i)$ $(i \in U)$, where C is

- union: single-sketch RC adjusted weights on the sketch of the union $s_k(U,r)$
- SCS: SCS RC adjusted weights on $\mathrm{SCS}_k(\mathcal{S},r)$
- LCS: LCS RC adjusted weights on $\mathrm{LCS}_k(\mathcal{S},r)$

LEMMA 6. *For any $J \subset U$,*

$$\mathrm{VAR}[a^{(\mathrm{LCS})}(J)] \leq \mathrm{VAR}[a^{(\mathrm{SCS})}(J)] \leq \mathrm{VAR}[a^{(union)}(J)] \ .$$

PROOF. Because all methods have zero covariances between different keys, it suffices to establish that relation for the variances of per-key adjusted weights, that is, for all $i \in U$,
$\mathrm{VAR}[a^{(\mathrm{LCS})}(i)] \leq \mathrm{VAR}[a^{(\mathrm{SCS})}(i)] \leq \mathrm{VAR}[a^{(\mathrm{union})}(i)]$.

Consider a key i and a subspace R of the sample space of rank assignments such that the rank values of all other keys are fixed. It suffices to show the variance relation in each such subspace.

Let $q^{(\mathrm{union})}(R,i)$, $q^{(\mathrm{SCS})}(R,i)$, $q^{(\mathrm{LCS})}(R,i)$ be the probabilities conditioned on R that i is included in the respective combination. Since the probability $p(w(i),\tau)$ is decreasing with τ and $r_{k+1}(\bigcup_{A\in\mathcal{S}} A) \leq r_{k+1}(\mathcal{S}) \leq \tau(\mathcal{S},r,i)$, we have that
$q^{(\mathrm{union})}(R,i) \leq q^{(\mathrm{SCS})}(R,i) \leq q^{(\mathrm{LCS})}(R,i)$.

For any combination $C \in \{\mathrm{union}, \mathrm{LCS}, \mathrm{SCS}\}$ the adjusted weight in R is the HT estimator $a^{(C)}(i) = w(i)/q^{(C)}(R,i)$. The variance of $a^{(C)}(i)$ is decreasing with the probability $q^{(C)}(R,i)$, which concludes the proof. $\square$

5. COMPUTING ESTIMATES

The input to our estimation procedure is a set of coordinated bottom-k sketches $s_k(A,r)$ for sets $A \subset I$, $A \in \mathcal{A}$, and a weight query specified by a predicate $P : I$. The desired output is an estimate of $\sum_{i\in I|P(i)} w(i)$.

We use the following two definitions:

- A set of *relevant sets* $\mathcal{S} \subset \mathcal{A}$ for a predicate P is a set of sets that suffices to determine the keys that satisfy P. For example, for the query "term is present in at least 2 out of books A, B, C," the relevant sets are A, B, and C. The query "term present in A and not in C" has relevant sets A and C. In both cases, these are minimum relevant sets. The first step of processing the query for P is determining (preferably a minimal) set $\mathcal{S}$ of relevant sets.
- The *best applicable combination* for P is the most inclusive combination $C \in \{\mathrm{SCS}, \mathrm{LCS}\}$ that allows us to evaluate the sum $\sum_{i\in C|P(i)} a(i)$ using information that is available from the sketches of $\mathcal{S}$. Since we get better estimates with the LCS, we should use the LCS when it is applicable.

We can evaluate $P(i)$ for all $i \in \mathrm{SCS}$ for general P. This is because we have full membership information in $\mathcal{S}$ sets for all keys in $\mathrm{SCS}(\mathcal{S})$. For $i \in \mathrm{LCS}$, we can determine membership of i only in sets $A \in \mathcal{S}$ such that $r(i) \leq r_{k+1}(A)$. Since the combination must be applicable to all rank assignments, we can apply the LCS only to predicates P that have the form of an attribute-based condition over keys in $\bigcup_{A\in\mathcal{S}} A$. As an example, the SCS is the best applicable combination for the intersection of two sets $A \cap B$.[2]

Input: set of coordinated bottom-k sketches $s_k(A,r)$ for sets $A \in \mathcal{A}$; predicate P
- Analyze P to determine:
  - ◇ A (minimum) set $\mathcal{S}$ of "relevant sets."
  - ◇ The best applicable combination $C \in \{\mathrm{SCS}, \mathrm{LCS}\}$:
    If P is an attribute-based condition over $\bigcup_{A\in\mathcal{S}} A$, $C \leftarrow$ LCS.
    Else, $C \leftarrow$ SCS.
- Retrieve the sketches $s_k(A,r)$ of the sets $A \in \mathcal{S}$.
- Compute adjusted weights $a^{(C)}(i)$ for $i \in C_k(\mathcal{S},r)$ using (1), if $C \equiv$ SCS, or (2), if $C \equiv$ LCS.
- **Output:** $\sum_{i\in C|P(i)} a(i)$.

Note that once adjusted weights are computed, they can be applied to multiple predicates that share the same relevant set $\mathcal{S}$ and best applicable combination C.

Figure 3 illustrates the evaluation of an approximate weight for some example predicates.

6. UNBIASED SELECTIVITY ESTIMATORS

We estimate selectivities through *adjusted selectivities* $\rho(i)$ such that $\mathrm{E}[\rho(i)] = w(i)/w(U)$ (for all $i \in U$).

We consider three types of sketches $M \in \{\mathrm{WSR}, \mathrm{WSRD}, \mathrm{WSRC}\}$ based on sampling with replacement from U. For an infinite sequence s of weighted sampling with replacement from U, we consider sampling with the following stopping rules. (i) WSR (k-mins): after k (not necessarily distinct) samples, (ii) WSRD: when seeing the $k+1$ distinct key, (iii) WSRC: with respect to $\mathcal{S}$, when, for at least one set $A \in \mathcal{S}$, we see the $(k+1)$st distinct key from A.

The respective M sketch is a set of keys and multiplicities $c^{(M)}(i,s)$ $(i \in U)$, (the number of times i was sampled before stopping). $c^{(M)}(U,s)$ denotes the sum of multiplicities of keys.

LEMMA 7. *For $M \in \{\mathrm{WSR}, \mathrm{WSRD}, \mathrm{WSRC}\}$, $\rho_1^{(M)}(i,s) = c^{(M)}(i,s)/c^{(M)}(U,s)$ are correct adjusted selectivities.*

PROOF. WSR**:** By definition, $c^{(M)}(U,s) \equiv k$ and we obtain the WSR k-mins estimator in [12, 7]. This well-known estimator, used in [12, 7] to estimate the resemblance of A_1 and A_2 (the sum of multiplicities of keys from $A_1 \cap A_2$ in the WSR k-mins sketch of $A_1 \cup A_2$, divided by k.), assigns to each key an adjusted selectivity equals to its multiplicity in the sketch times $1/k$.

WSRD**:** Consider a key i. Partition the probability space so that in each set of the partition the number of samples of keys from $U \setminus \{i\}$ until we get k distinct keys from $U \setminus \{i\}$ is fixed. We will show that $\rho(i)$ is an unbiased selectivity

[2]We can still use the LCS indirectly to estimate $w(A \cap B)$ using the inclusion-exclusion formula $w(A \cap B) = w(A) + w(B) - w(A \cup B)$. But this estimator does not perform well (see Section 7).

in each subspace. Consider a subspace where the number of samples of keys from $U \setminus \{i\}$ until we get k distinct keys from $U \setminus \{i\}$ is ℓ. (Notice that $\ell \geq k$.) The estimator $\rho(i)$ in this subspace is $\frac{c(i,s)}{c(i,s)+\ell-1}$. This is because if we do not sample i by the time we get k distinct keys from $U \setminus \{i\}$ then $c(i,s) = 0$ as well as $\rho(i)$, and otherwise $c(U,s) = c(i,s) + \ell - 1$ and therefore $\rho(i) = \frac{c(i,s)}{c(i,s)+\ell-1}$.

The number of times i is sampled between two samples from $U \setminus \{i\}$ is geometrically distributed with parameter $p = w(i)/w(U)$. Therefore we need to show that

$$\sum_{i_1=0}^{\infty} \cdots \sum_{i_\ell=0}^{\infty} p^{\sum_{j=1}^{\ell} i_j} (1-p)^{\ell} \frac{\sum_{j=1}^{\ell} i_j}{\ell - 1 + \sum_{j=1}^{\ell} i_j} = p \ . \qquad (3)$$

By combining together terms in which $\sum_{j=1}^{\ell} i_j = t$ in the left side of (3) we obtain that

$$\begin{aligned}
&\sum_{t=1}^{\infty} \binom{t+\ell-1}{\ell-1} p^t (1-p)^{\ell} \frac{t}{t+\ell-1} \\
&= (1-p)^{\ell} \sum_{t=1}^{\infty} \binom{t+\ell-2}{\ell-1} p^t = (1-p)^{\ell} p \sum_{t=1}^{\infty} \binom{t+\ell-2}{\ell-1} p^{t-1} \\
&= (1-p)^{\ell} p \sum_{t=0}^{\infty} \binom{t+\ell-1}{\ell-1} p^t = (1-p)^{\ell} p (1+p+p^2+\cdots)^{\ell} \\
&= p
\end{aligned}$$

WSRC**:** For subsets $A \in S$ such that $i \in A$ consider the occurrence of the kth distinct key from $A \setminus \{i\}$ and for subsets such that $i \notin A$ consider the occurrence of the $(k+1)$st distinct key from A. Fix ℓ and consider the subspace of the probability space where the total number of samples until and including the first among these occurrences. If i is sampled at least once, then there are $\ell - 1$ samples from $U \setminus \{i\}$ in the WSRC sketch. The number of times i is sampled between two samples from $U \setminus \{i\}$ is geometrically distributed with parameter $w(i)/w(U)$. The proof proceeds as the proof for WSRD sketches. □

LEMMA 8. *For $M \in \{\text{WSR}, \text{WSRD}, \text{WSRC}\}$:*

- *For $i \neq j \in U$, $\text{COV}[\rho_1^{(M)}(i,s), \rho_1^{(M)}(j,s)] \leq 0$.*
- *$\sum_{i \in U} \rho_1^{(M)}(i,s) = 1$.*

LEMMA 9. *For $M \in \{\text{WSR}, \text{WSRD}, \text{WSRC}\}$ and $J \subset U$:*

$$\text{VAR}[\rho_1^{(\text{WSRC})}(J,s)] \leq \text{VAR}[\rho_1^{(\text{WSRC})}(J,s)] \leq \text{VAR}[\rho_1^{(\text{WSR})}(J,s)] \ .$$

For a bottom-k sketch when all keys have equal weights, Broder observed [6, 39] that the fraction of keys in the sketch of the union $A_1 \cup A_2$ that are contained in $A_1 \cap A_2$ is an unbiased estimator of the Jaccard coefficient. More generally, adjusted selectivities of $\rho^{(\text{WS})}(i) = 1/k$ are correct for WS sketches when the keys have equal weights. This is not true anymore if keys have different weights.

Unbiased selectivity estimators for WS bottom-k sketches can be obtained via a *mimicking process* [17]. The mimicking process is a randomized algorithm that inputs a WS sketch and output a sequence of "emulated" weighted samples with replacement. We can also apply mimicking to an SCS of WS bottom-k sketches by arranging the keys in the SCS by increasing rank values and using this as an input to the process.

If we stop the process when k keys (not necessarily distinct) are drawn, we obtain a WSR-sketch. If we continue until we see the $(k+1)$st distinct key, which exhausts the "information" in the WS sketch, we obtain a WSRD sketch. If applied to an SCS until the information is exhausted, we obtain a WSRC *sketch*.

Mimicking allows us to carry over unbiased estimators applicable to WSR, WSRD, and WSRC sketches to WS sketches and SCS combinations.

Tighter estimators. The adjusted selectivities $\rho_1^{(M)}$ $(i \in s)$ have the desirable qualities of (i) non-positive covariances between different keys and (ii) adjusted selectivities sum up to one. (See [19, 13, 49] for a discussion of these qualities.)

We obtain tighter estimators than $\rho_1^{(M)}$, that share these qualities but have a lower sum of per-key variances.

Mimicking is a random process and therefore, each WS sketch or SCS corresponds to a probability distribution D over WSR and WSRD (for SCS also WSRC) sketches. Tighter estimators are obtained by taking the expectation of $\rho_1^{(M)}$ (or average over multiple draws) over D. We can get even tighter estimators by looking at the expectation of this estimator over equivalence classes of WS sketches (or SCS combinations). Equivalence class can include all sketches/combinations with same rank ordering of keys, obtained by redrawing the ranks of keys, or (if total weight is available) containing the same set of keys [17, 19]. One interesting corollary is the following:

LEMMA 10. *If all weights are equal, then $\rho^{\text{SCS}}(i) = 1/\ell$ for all $i \in \text{SCS}_k(\mathcal{S}, r)$, where $\ell = |\text{SCS}_k(\mathcal{S}, r)|$, are correct adjusted selectivities.*

PROOF. Redrawing the rank values of the first ℓ keys in U does not change the SCS. The resulting distribution is symmetric for all ℓ keys and therefore the expectation of $\rho_1(i)$ is the same. □

The adjusted selectivities $\rho^{\text{SCS}}(i) = 1/\ell$ are superior to $\rho^{\text{WS}}(i) = 1/k$ (and in particular, improve over classic union-sketch resemblance estimator [6, 39]). Both estimators have symmetric nonnegative covariances and the adjusted selectivities sum up to 1. However, $\text{VAR}[\rho^{\text{WS}}(i)] = N/k - 1 \geq \text{VAR}[\rho^{\text{SCS}}(i)] = N/k' - 1$, where N is the total number of keys and $k' = 1/\text{E}[1/\ell]$, where $\ell \geq k$ is the number of keys in the SCS. (Let p_ℓ be the probability that the SCS contains ℓ keys. We have $\text{VAR}[\rho^{\text{SCS}}(i)] = \sum_\ell p_\ell (N/\ell - 1) = N/k' - 1$.) We typically have $k' \approx \text{E}[\ell]$. Section 7 includes an evaluation of this estimator relative to the classic union-sketch estimators for Jaccard coefficient.

7. EMPIRICAL EVALUATION

We compare our combination estimators to state of the art estimators applied to the union sketch. As a point of reference, we also include k-mins estimators applied to sketch of the union of k-mins sketches. We measure the benefit of combination estimators by their *improvement factor*, which is the ratio of average relative error of (the best) union-sketch estimator to that of the combination estimator.

Datasets. We used synthetic data designed to quantify and demonstrate how the quality and relative performance of the estimators depends on different parameters of the data, such as the number of relevant sets in the selection predicate and the relation between these sets. We also used the following real-life datasets that demonstrate example applications:

- Two IP packet traces of about 9×10^6 packets from gateway routers (*peering* and *campus*). These traces were partitioned

into 5 consecutive time periods and we produced coordinated sketches of the set of destination IP addresses in each time period. The campus data had 3196, 2636, 2656, 2175, 2105 distinct addresses in each time period and 6830 distinct addresses overall. The peering data had 14158, 14564, 14281, 14705, 14483 distinct addresses in each time period and 37574 distinct IP addresses overall.

• The Netflix Prize [40] Data, that consists of about 1×10^8 reviews by 5×10^5 users of 17770 movies. We consider the set of reviewers of each movie as a "set," and produced coordinated sketches for these sets.

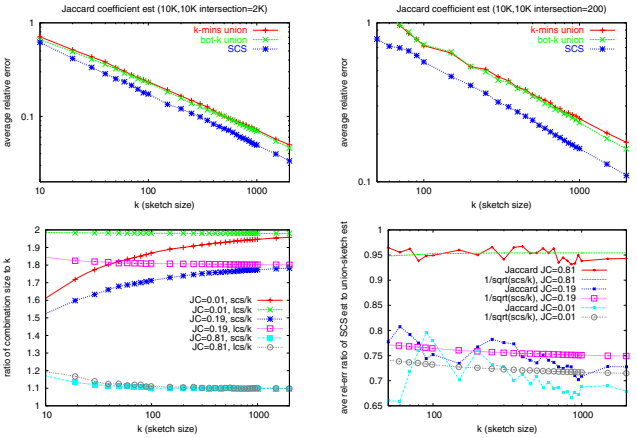

Figure 4: Top: Averaged relative error of different estimators for the Jaccard coefficient of two sets, each containing 10000 (uniformly weighted) keys. The size of the intersection is 2000 (left) and 200 (right). Bottom: Left: Combination sizes for 2 sets containing 10000 keys. Right: Ratio of averaged relative error of SCS to union-sketch estimators of Jaccard coefficient (with corresponding square root of the ratio of k and SCS size.)

Predicates with 2 relevant sets. We first consider basic pairwise aggregates: The union size, intersection size, Jaccard coefficient, and Hamming distance (the difference of the sizes of union and intersection).

We use two sets A_1 and A_2 of equal size $(10,000)$ and varied the number of common keys $|A_1 \cap A_2| \in \{200, 2000, 9000\}$ (respective Jaccard coefficients 0.81, 0.19, 0.01). We applied the RC union, k-mins union, and our RC SCS and RC LCS estimators for the size of the union. We applied the RC union, k-mins union, and our RC SCS for the size of the intersection. The intersection estimator based on inclusion exclusion and the RC LCS estimate of the union $w(A_1)+w(A_2)-\tilde{w}(A_1 \cup A_2)$ was also evaluated but it performed considerably worse than other estimators and is not shown. Hamming distance is estimated as the difference of union and intersection estimators (as the difference of unbiased estimators, this estimator is unbiased. It is also easy to show from the derivation that the estimate is always nonnegative). For the Jaccard coefficient, we applied the classic k-mins and bottom-k union estimators of Broder [7, 6] and our SCS combination selectivity estimator (Section 6). Figure 4 shows the average relative error, over 1000 runs, of Jaccard coefficient estimators. For uniform weights and for k small relative to number of keys, the relative error of the union-sketch estimators decreases proportionally to $\sqrt{k}$ and there was a proportional decrease also for the combination estimators.

The improvement factor of combination estimators is larger when the Jaccard coefficient is smaller. The intuitive reason is that smaller Jaccard coefficient means less overlap between the sets, hence less overlap between sketches, and more distinct keys in the combination that are available to the combination estimators. We relate the improvement factor to the size of the combination. Figure 4 (bottom, left) shows the ratio ℓ/k, where ℓ is the average size of the combination (SCS and LCS) and k is the size of the union-sketch. The figure demonstrates that the combination size is larger when the Jaccard coefficient is smaller. Figure 4 (bottom, right) shows the improvement factor and the respective $\sqrt{k/\ell}$ for our Jaccard coefficient estimators. In agreement with an analytic approximation (Section 6), we can see that the improvement factor is approximated well by $\sqrt{\ell/k}$, where ℓ is the combination size.

In particular, our combination estimator for Jaccard coefficient has about half the variance of union-sketch estimators [7, 6] when the two sets are almost disjoint. For the applications of identifying all similar pairs [7, 31], and on typical corpuses, with only a small fraction of pairs being similar, our estimator significantly decreases "false positives."

Performance dependence on the number of relevant sets. We next consider a synthetic distribution where all sets share 1000 common keys and each set has its own 5000 unique keys. This collection of sets allows us to study how the benefit of combination estimators increases with the number of sets. Figure 6 (top) shows the average relative error for estimating the size of the union of multiple (2,3,4, and 5) sets using the RC union, RC SCS, and the RC LCS estimators. The average relative error of union-size estimator applied to the sketch of the union is about $\sqrt{2/(\pi k)}$ and is about $\sqrt{2/(\pi \ell)}$ for the combination estimators. Figure 6 (bottom) shows combination size ratio to k. A simple calculations shows that the LCS size with i sets is about $\ell = 0.2k + 0.8ik$. The SCS size ratio varies with k and approaches the LCS size ratio as k increases. Figure 6 also demonstrates that improvement factors are approximated well by $\sqrt{\ell/k}$, where ℓ is the size of the combination.

Figure 5 shows the improvement factor of SCS RC and LCS RC estimators on the destination IP addresses data sets. We estimate the total number of distinct destination IP addresses (union) and the number of common destination IP addresses (intersection) of the first $i \in \{2, 3, 4, 5\}$ time periods. The figure shows how the improvement factor of the SCS and (in particular) the LCS increases with the number of sets. The improvement factor is again approximated well by $\sqrt{\ell/k}$ (not shown).

Performance dependence on the relation between the sets. When sets have fewer common keys, combinations contain more keys, and combination estimators have larger improvement factors. We demonstrate this using two collections $\mathcal{S}_1$ and $\mathcal{S}_2$ of 5 sets each. Both collections have the same size union (49530 keys). $\mathcal{S}_1$ contains 5 disjoint sets of 9906 keys. $\mathcal{S}_2$ contains sets of size 29718 with 24765 keys common to all sets and 4953 exclusive keys for each set. The LCS of $\mathcal{S}_1$ contains about $5k$ keys. The LCS of $\mathcal{S}_2$ contains about $5k/3$ keys (5/6 of the keys in each sketch are common to all 5 sets). Figure 7 shows corresponding improvement factors of $\sqrt{5}$ for $\mathcal{S}_1$ and $\sqrt{5/3}$ for $\mathcal{S}_2$.

SCS versus LCS. Figures 5,6,7 show comparable perfor-

mance factors (reflecting similar sizes) for the SCS and the LCS. When can we expect the SCS to be large? For $A \in \mathcal{S}$, keys in the sketch of $A \in \mathcal{S}$ are included in the SCS only if they have rank smaller than $r_{k+1}(\mathcal{S})$. Thus, when $r_{k+1}(\mathcal{S}) = \min_{B \in \mathcal{S}} r_{k+1}(B)$ is close to $r_{k+1}(A)$, most keys are included in the SCS. The SCS is large when sets have closely related distributions of $r_{k+1}(A)$ (sets have similar weights) *and* when $|\mathcal{S}|$ is smaller (see Figure 6).

Figure 8 shows the performance of estimators for two queries on the Netflix data set: "the number of users with at least one rating of a National Geographic title" and "the number of users with at least one rating of a movie released on or before 1930." These are estimates on the size of the union of sets. The corresponding sets of movie titles where larger (more sets than in previous datasets) and heterogeneous (high variability in number of reviewers of different titles). For the first query, there were 45 National Geographic titles with 19708 ratings by 12351 distinct reviewers. The number of ratings for each NG title varied between 93 and 1170 (mean is about 438). For the second query, there were 120 titles with release year on or before 1930. There were 117617 ratings with 53774 distinct reviewers. The number of ratings per title ranged between 54 and 12054 (mean is 980). We observe improvement factor of 3-4 of the RC LCS estimator over RC union but we also see a ratio of 1.5-2 between the relative errors of the RC SCS and the RC LCS estimates, reflecting a much smaller SCS than LCS.

Lastly, we consider the incremental effectiveness of combination samples. SCS samples (that are not included in the union-sketch) are always as effective as additional samples from the union of the sets. LCS samples (that are not included in the SCS) can be as effective, but the effectiveness decreases with heterogeneity of $\mathcal{S}$. Intuitively, consider two sets, one much larger than the other, and each contributing k samples. Then samples from the smaller set (that are mostly excluded from the SCS) are much less useful to estimate properties of the union of the sets. On the other hand, if we have multiple homogeneous sets, the SCS is smaller than LCS due to the "variance" of the $k+1$st rank, but LCS samples are as effective.

k-mins versus bottom-k. "Without replacement" (bottom-k) estimators dominate "with replacement" (k-mins) estimator, but the gain is negligible with uniform weights (see Figure 4). Gain can be significant only when keys are likely to be sampled repeatedly under "with replacement" sampling [24, 17]. With uniform weights, union-sketch k-mins estimators performs similarly to respective union-sketch bottom-k estimators *but* combination estimators typically outperform union-sketch estimators. Since combination estimators are *not applicable* to k-mins sketches, this suggests the use of bottom-k sketches also with uniform weights.

Weighted keys. Improvement factor, as a function of ℓ/k, is larger when keys are weighted. This is because variance decrease with sample size is *at least* $1/k$ (relative error decrease is *at least* $1/\sqrt{k}$), with uniform weights exhibiting the "worst-case" decrease.

Restricted predicates. The demonstrated performance factor on unions and intersections of sets carries over when adding attribute based conditions to the predicate. This is because also with added conditions, the combination contains proportionally more keys than the union sketch. Examples of attribute-based conditions (on IP addresses) is to restrict the query to blacklisted addresses or addresses that belongs to a particular Autonomous System or (on Netflix-like data) to reviewers from a certain gender or zip-code.

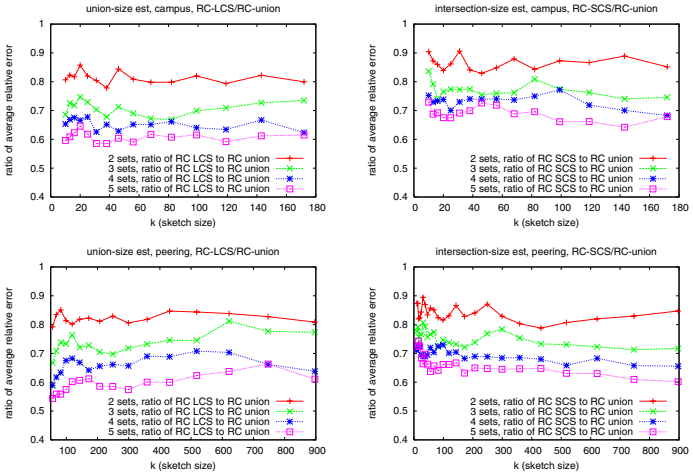

Figure 5: Ratio of the average relative error to that of the RC union estimator (inverse improvement factor) as a function of k. Applied to sketches of destination IP addresses in $i \in \{2,3,4,5\}$ consecutive time periods (Top: *campus* data set, Bottom: *peering* data set). Left : RC LCS estimate of the union of the first i time periods. Right: RC SCS estimate of the intersection of the first i periods.

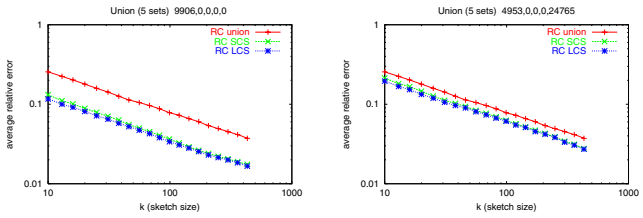

Figure 7: Relative error of RC union, RC SCS, and RC LCS estimators on the size of the union of 5 sets. Size of the union is 49530. Left: 5 disjoint sets of size 9906. Right: 5 sets with 24765 common keys to all 5 and 4953 exclusive keys in each set.

8. RELATED WORK

Sample-based coordinated sketches. Coordinated samples of multiple sets, based on keys "retaining" the same "random draw" across sets, are extensively used as a way to maximize or minimize sample overlap [5, 41, 43, 44] or to facilitate (approximate) aggregations over distinct keys [12, 27, 28, 7, 16, 17, 19]. Sample-based coordinated sketches where used with size-k samples with replacement (k-mins sketches) [12, 7, 16], size-k samples without replacement (bottom-k/order samples) [41, 43, 12, 17, 19], and Poisson sampling [5, 27, 28].

Multiple-set aggregates. The union-sketch reduction was used with both k-mins and bottom-k sketches [12, 7, 6, 2, 23]. Bottom-k sketches dominate k-mins sketches in terms of estimation quality [17], and in particular, this dominance extends to union-sketch based estimates. Moreover, in contrast to bottom-k sketches, we are not aware of a better estimator over k-mins sketches than through the union-sketch

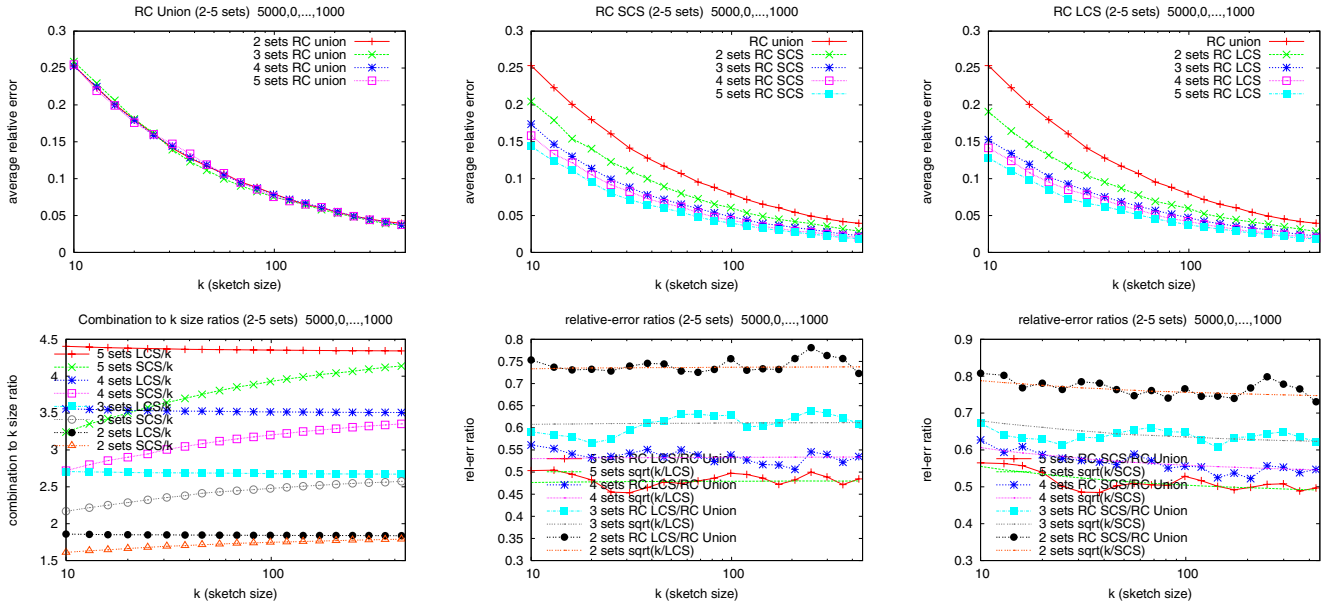

Figure 6: Top: Relative error of RC union, RC SCS, and RC LCS estimators on the size of the union of 2,3,4, and 5 sets of size 5000 each with intersection of size 1000. Bottom: size ratios of combinations to k (left) and relative error ratios for LCS (middle) and SCS (right).

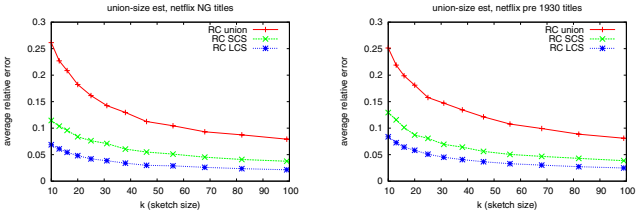

Figure 8: Relative error of RC union, RC SCS, and RC LCS estimators on "number of distinct reviewers of National Geographic titles" and "number of distinct reviewers of titles released on or before 1930." (Netflix data set)

reduction. Multiple-set aggregates over coordinated Poisson samples can be approximated by computing a Poisson sample of the union of the sets (see [27, 28] for uniform sampling). More generally, over Poisson samples with *expected* size k, it is possible to derive "LCS-like" estimators that use all sampled keys and "SCS-like" estimators that use all keys below the lowest sampling rate. These estimators perform similarly to combination estimators over bottom-k sketches. Poisson samples, however, have the disadvantage of variable sample size.

The only previous work we are aware of that leveraged combinations of bottom-k sketches is [35], but they only derive ML estimators that are biased and applicable only to WS bottom-k sketches.

Coordinated sketches that are not sample based. A strength of sampling-based coordinated sketches is the generality of the selection predicates combined with a tunable and potentially very small summary size. Methods that are not sample-based include bloom filters [4] and variants [25] that have the drawback that summary size grows linearly with the size of the corpus. Other methods, such as Charikar's simhash [10], produce tunable small-size summaries [36, 26, 10, 11, 45, 22, 31, 34, 37]. These methods are very effective for some tailored goals, such as pairwise similarity measures between sets [31], but have inherent limitations: Since the summary does not retain keys' identifiers or meta-data, there is no support for predicates with attribute-based conditions. For example, in a market basket data set, where baskets are "keys" and goods are "sets," we can estimate the association "purchase of beer implies purchase of diapers", using the ratio of the number of baskets with beer and diapers (size of the intersection) and the number of baskets with beer. The more refined query where the selection is restricted to consumer/basket segments (such as "female consumer," "basket contains at most 12 goods," or "paid in cash."), however, can not be supported. Furthermore, only a limited set of membership-based conditions is supported and inherently these methods do not provide a "representative sample" of keys that satisfy the predicate.

A recent sampling scheme, *varopt*, minimizes the sum of variances of sets of any fixed size [13]. We do not know how to apply it to produce coordinated sketches. This paper expands a previous 6-page exposition [18].

9. CONCLUSION

Sketches based on coordinated random samples are a classic summarization method for datasets modeled as a collection of sets over a ground set of keys. The sketch of each set is a weighted sample of the keys with some auxiliary information. This powerful model covers a wide range of applications that require scalability in computing the sketches, estimation quality, and flexibility in terms of supported approximate aggregates.

We propose novel unbiased estimators for multiple-set weight and selectivity aggregates over coordinated bottom-k sketches. Our *combination* estimators outperform the existing union-sketch estimators by using more samples present in the sketches of the sets relevant to the query. We quantify the advantage of combination estimators over union-sketch estimators

through an extensive empirical evaluation. Our evaluation suggests that combination estimators applied when the combination has average size of ℓ (has ℓ distinct keys) perform *comparably* to estimators applied to a size-ℓ union-sketch (derived from coordinated bottom-ℓ sketches). In particular, we can expect ℓ/k factor reduction in variance ($\sqrt{\ell/k}$ reduction in estimation error) for uniform weights (distinct values count) and a larger factor for skewed distributions. The size ℓ of a combination is in $[k, t*k]$, where t is the number of relevant sets. Combination size is larger when there are more sets, when sets have fewer common keys, and when sets have homogeneous weights. Our evaluation, which includes natural queries on real data sets demonstrate typical 25%-75% reduction in estimation error.

10. REFERENCES

[1] R. Agrawal, T. Imielinski, and A. N. Swami. Mining association rules between sets of items in large databases. In *SIGMOD*, pages 207–216, 1993.

[2] K. S. Beyer, P. J. Haas, B. Reinwald, Y. Sismanis, and R. Gemulla. On synopses for distinct-value estimation under multiset operations. In *SIGMOD*, pages 199–210, 2007.

[3] K. Bharat and A. Z. Broder. Mirror, mirror on the web: A study of host pairs with replicated content. In *WWW*, pages 501–512, 1999.

[4] B. Bloom. Space/time tradeoffs in in hash coding with allowable errors. *Communications of the ACM*, 13:422–426, 1970.

[5] K. R. W. Brewer, L. J. Early, and S. F. Joyce. Selecting several samples from a single population. *Australian Journal of Statistics*, 14(3):231–239, 1972.

[6] A. Z. Broder. On the resemblance and containment of documents. In *Proceedings of the Compression and Complexity of Sequences*, pages 21–29. ACM, 1997.

[7] A. Z. Broder. Identifying and filtering near-duplicate documents. In *CPM*, pages 1–10, 2000.

[8] A. Z. Broder, M. Charikar, A. M. Frieze, and M. Mitzenmacher. Min-wise independent permutations. *J. Comput. System Sci.*, 60(3):630–659, 2000.

[9] M. Charikar, S. Chaudhuri, R. Motwani, and V. Narasayya. Towards estimation error guarantees for distinct values. In *PODS*, 2000.

[10] M. S. Charikar. Similarity estimation techniques from rounding algorithms. In *STOC*, 2002.

[11] A. Chowdhury, O. Frieder, D. Grossman, and M. C. McCabe. Collection statistics for fast duplicate document detection. *ACM Transactions on Information Systems*, 20(1):171–191, 2002.

[12] E. Cohen. Size-estimation framework with applications to transitive closure and reachability. *J. Comput. System Sci.*, 55:441–453, 1997.

[13] E. Cohen, N. Duffield, H. Kaplan, C. Lund, and M. Thorup. Stream sampling for variance-optimal estimation of subset sums. In *SODA*, 2009.

[14] E. Cohen, A. Fiat, and H. Kaplan. Associative search in Peer to Peer networks: Harnessing latent semantics. In *INFOCOM*, 2003.

[15] E. Cohen and H. Kaplan. Efficient estimation algorithms for neighborhood variance and other moments. In *SODA*, 2004.

[16] E. Cohen and H. Kaplan. Spatially-decaying aggregation over a network: model and algorithms. *J. Comput. System Sci.*, 73:265–288, 2007.

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

[18] E. Cohen and H. Kaplan. Estimating aggregates over multiple sets. In *ICDM*, 2008.

[19] E. Cohen and H. Kaplan. Tighter estimation using bottom-k sketches. In *VLDB*, 2008.

[20] E. Cohen and M. Strauss. Maintaining time-decaying stream aggregates. *J. Algorithms*, 59:19–36, 2006.

[21] E. Cohen, Y.-M. Wang, and G. Suri. When piecewise determinism is almost true. In *Proc. Pacific Rim International Symposium on Fault-Tolerant Systems*, pages 66–71, 1995.

[22] J. G. Conrad and C. P. Schriber. Constructing a text corpus for inexact duplicate detection. In *SIGIR*, pages 582–583, 2004.

[23] T. Dasu, T. Johnson, S. Muthukrishnan, and V. Shkapenyuk. Mining database structure; or, how to build a data quality browser. In *SIGMOD*, pages 240–251, 2002.

[24] N. Duffield, M. Thorup, and C. Lund. Priority sampling for estimating arbitrary subset sums. *J. Assoc. Comput. Mach.*, 54(6), 2007.

[25] L. Fan, P. Cao, J. Almeida, and A. Z. Broder. Summary cache: a scalable wide-area Web cache sharing protocol. *IEEE/ACM Transactions on Networking*, 8(3):281–293, 2000.

[26] J. Feigenbaum, S. Kannan, M. Strauss, and M. Viswanathan. An approximate L1-difference algorithm for massive data streams. In *FOCS*, pages 501–511, 1999.

[27] P. Gibbons and S. Tirthapura. Estimating simple functions on the union of data streams. In *Proceedings of the 13th Annual ACM Symposium on Parallel Algorithms and Architectures*. ACM, 2001.

[28] P. B. Gibbons. Distinct sampling for highly-accurate answers to distinct values queries and event reports. In *VLDB*, pages 541–550, 2001.

[29] M. Hadjieleftheriou, X. Yu, N. Koudas, and D. Srivastava. Hashed samples: Selectivity estimators for set similarity selection queries. In *VLDB*, 2008.

[30] J. Hájek. *Sampling from a finite population*. Marcel Dekker, New York, 1981.

[31] M. R. Henzinger. Finding near-duplicate web pages: a large-scale evaluation of algorithms. In *SIGIR*, pages 284–291, 2006.

[32] D. G. Horvitz and D. J. Thompson. A generalization of sampling without replacement from a finite universe. *Journal of the American Statistical Association*, 47(260):663–685, 1952.

[33] H. Kaplan and M. Sharir. Randomized incremental constructions of three-dimensional convex hulls and planar voronoi diagrams, and approximate range counting. In *SODA*, pages 484–493, 2006.

[34] A. Kolcz, A. Chowdhury, and J. Alspector. Improved robustness of signature-based near-replica detection via lexicon randomization. In *SIGKDD*, pages 605–610, 2004.

[35] P. Li and K. W. Church. A sketch algorithm for estimating two-way and multi-way associations. *Computational Linguistics*, 33:305–354, 2007.

[36] U. Manber. Finding similar files in a large file system. In *Usenix*, pages 1–10, 1994.

[37] G. S. Manku, A. Jain, and A. D. Sarma. Detecting nearduplicates for web crawling. In *WWW*, 2007.

[38] D. Mosk-Aoyama and D. Shah. Computing separable functions via gossip. In *PODC*, 2006.

[39] R. Motwani, E. Cohen, M. Datar, S. Fujiware, A. Gronis, P. Indyk, J. Ullman, and C. Yang. Finding interesting associations without support pruning. *IEEE Transactions on Knowledge and Data Engineering*, 13:64–78, 2001.

[40] The Netflix Prize. `http://www.netflixprize.com/`.

[41] E. Ohlsson. Sequential poisson sampling. *J. Official Statistics*, 14(2):149–162, 1998.

[42] B. Rosén. Asymptotic theory for successive sampling with varying probabilities without replacement, I. *The Annals of Mathematical Statistics*, 43(2):373–397, 1972.

[43] B. Rosén. Asymptotic theory for order sampling. *J. Statistical Planning and Inference*, 62(2):135–158, 1997.

[44] B. Rosén. On sampling with probability proportional to size. *J. Statistical Planning and Inference*, 62(2):159–191, 1997.

[45] S. Schleimer, D. Wilkerson, and A. Aiken. Winnowing: local algorithms for document fingerprinting. In *SIGMOD*, 2003.

[46] N. T. Spring and D. Wetherall. A protocol-independent technique for eliminating redundant network traffic. In *SIGCOMM*, 2000.

[47] K. Sripanidkulchai, B. Maggs, and H. Zhang. Efficient content location using interest-based locality in peer-to-peer systems. In *INFOCOM*, 2003.

[48] M. Szegedy. The DLT priority sampling is essentially optimal. In *STOC*, 2006.

[49] M. Szegedy and M. Thorup. On the variance of subset sum estimation. In *ESA*, 2007.

[50] C. Tang, Z. Xu, and S. Dwarkadas. Peer-to-peer information retrieval using self-organizing semantic overlay networks. In *SIGCOMM*, 2003.