Practical Algorithms for Tracking Database Join Sizes

Sumit Ganguly, Deepanjan Kesh, and Chandan Saha

Indian Institute of Technology, Kanpur

Abstract. We present novel algorithms for estimating the size of the natural join of two data streams that have efficient update processing times and provide excellent quality of estimates.

1 Introduction

The problem of accurately estimating the size of the natural join of two database tables is a classical problem[15, 13, 1, 11, 12], with fundamental applications to database query optimization and approximate query answering. Prior work in the '80s through the mid '90s largely focussed on the *stored data* model, where, the joining relations are either disk or memory-resident. *Sampling* emerged as a popular solution technique in this model [14, 15, 13].

The *streaming data model* [6, 5, 7, 4] was proposed in the late '90's as a model for a class of monitoring applications, such as network management, RF-id based applications, sensor networks, etc. These applications are characterized by high volumes of rapidly and continuously arriving records. The monitoring applications can often tolerate approximate answers, provided, (a) the error probability and the approximation ratio are both guaranteed to be low, (b) the rate of processing is able to keep pace with the fast arrival rates without significantly degrading the quality of answers, and, (c) the space consumed is significantly smaller than that needed for exact computation. Existing streaming algorithms satisfy a majority of the above properties, and in addition, process the stream in an online fashion, (i.e., look once only).

Data Stream Model and Notation. A data stream is viewed as a sequence of updates of the form (i, v), where, i takes values from the domain $\mathcal{D} = \{0, 1, \ldots, N-1\}$, and v is the change in the frequency of the items. If $v > 0$, then we can think of the tuple (i, v) as representing v insertions of i; correspondingly, if $v < 0$, then, (i, v) can be thought of as representing v deletions of i. The frequency of i, denoted by f_i, is the sum of the changes to the frequency of i since the inception of the stream, that is, $f_i = \sum_{(i,v) \text{ appears in stream}} v$. We denote by m_R the sum of the frequencies of the items in a stream R, that is, $m_R = \sum_{i \in \mathcal{D}} f_i$. In this paper, we consider the insert-only model of data streams (i.e., $v > 0$ for all updates) and the general update model of data streams (i.e., $v > 0$ or $v < 0$).

The self-join [2, 3, 1] of a stream R is denoted by $\text{SJ}(R)$ and is defined as $\text{SJ}(R) = \sum_{i \in \mathcal{D}} f_i^2$. For $r = 1, 2, \ldots, N$, let rank(r) be a (ranking) function that returns an item whose frequency is the r^{th} largest frequency in f (ties are broken arbitrarily). The *residual self-join* [8] of a stream R, denoted by $\text{SJ}^{res}(R, k)$ is defined as the self-join of

R after the top-k ranked frequencies are removed, that is, $\text{SJ}^{res}(R,k) = \sum_{r>k} f^2_{\text{rank}(r)}$. It is easily shown that $\text{SJ}^{res}(R,k) \le \frac{m_R^2}{4k}$.

In this paper, we consider two data streams R and S, and denote the frequencies of an item i in streams R and S by f_i and g_i respectively. The size J of the natural join of R and S is defined as $J = |R \bowtie S| = \sum_{i \in \mathcal{D}} f_i \cdot g_i$. Following standard convention, we let $0 < \epsilon \le 1$ and $0 < \delta < 1$ denote user-specified accuracy and confidence parameters respectively. When referring to the join of R and S, we use m to denote $m_R + m_S$, SJ to denote $\text{SJ}_R + \text{SJ}_S$, and $\text{SJ}^{res}(k)$ to denote $\text{SJ}^{res}(R,k) + \text{SJ}^{res}(S,k)$.

Previous work. The seminal work in [1–3] presents the product of sketches technique that estimates the join size using space $O(s \cdot (\log(mN)) \cdot \log \frac{1}{\delta})$ bits with additive error of $O(\frac{(\text{SJ}(R)\text{SJ}(S))^{1/2}}{\epsilon\sqrt{s}})$. The work in [1] also presents a space lower bound of $s = \Omega(\frac{m^2}{J})$ for approximating the join size J to within a constant confidence over general data streams. The product of sketches algorithm does not match the space lower bound for the problem, and, the time taken to process each stream update can be large ($O(s \cdot \log \frac{1}{\delta})$). The *Fast-AGMS* algorithm[10] is a time-efficient variant of the product of sketches technique, processing stream updates in time $O(\log \frac{1}{\delta})$, while providing the same space versus accuracy guarantees of the product of sketches algorithm.

COUNT-MIN sketches[9] presents an elegant technique for estimating the join size using space $O(s(\log N + \log m)\log \frac{1}{\delta})$ bits, time $O(\log \frac{1}{\delta})$ for processing each stream update and with additive estimation error of $O(\frac{m^2}{s})$. The cross-sampling algorithm [1] has similar properties; however, it is not applicable to streams with deletion operations and is known to be generally outperformed by sketch-based methods in practice. The skimmed-sketches algorithm [12] estimates the join size using space $O(s(\log N)\log(m \cdot N) \cdot \log \frac{(m \log N)}{\delta})$ bits, time $O(\log \frac{1}{\delta})$ for processing each stream update and with additive error of $O(\frac{m^2}{\epsilon s})$. The COUNT-MIN sketch and skimmed-sketch techniques match the worst-case lower bound for the problem. Their main drawback is that they often perform poorly in comparison with the simple product of sketches algorithm, since, the complexity term m^2 of [12] is in practice, much larger than the self-join sizes.

Contributions. In this paper, we present two novel, space-time efficient algorithms called REDSKETCH and REDSKETCH-A for estimating the size of the natural join of two data streams. The REDSKETCH algorithm estimates the join size using $O(s \cdot \log(mN) \cdot \log \frac{m}{\delta})$ bits , with additive error $= O(\frac{m \cdot (\text{SJ}^{res}(s))^{1/2}}{\sqrt{s}})$. The REDSKETCH-A algorithm estimates the join size using space $O(s \cdot \log(mN) \cdot \log \frac{m}{\delta})$ bits and with additive estimation error of $O\left(\frac{J^{2/3} \cdot (\text{SJ})^{1/6} \cdot (\text{SJ}^{res}(s))^{1/6}}{s^{1/6}}\right)$. Both algorithms process each stream update in time $O(\log \frac{m}{\delta})$ and match the space lower bound of [1] (up to logarithmic factors). Our algorithms are practically effective, since, the bounds are in terms of SJ and SJ^{res}_s, which are significantly less than m^2 and $\frac{m^2}{s}$, respectively, in practice.

Organization. The rest of the paper is organized as follows. In Section 2, we review basic data stream algorithms that we use later. Sections 3 and 4 present the REDSKETCH and the REDSKETCH-A algorithms respectively. We conclude in Section 5.

2 Review

In this section, we review sketches [2, 3], the algorithm *C*ountSketch [8] for approximately finding the top-k frequent items over R and the FAST-AGMS algorithm [10] for estimating binary join sizes.

Sketches and estimating self-join sizes. A sketch[2, 3] X of the stream R is a random integer defined as $X = \sum_{i \in \mathcal{D}} f_i \cdot x_i$, where, for each $i \in \mathcal{D}$, x_i is chosen randomly from the set $\{-1, +1\}$ such that the family of random variables $\{x_i\}_{i \in \mathcal{D}}$ are four-wise independent. The family $\{x_i\}_{i \in \mathcal{D}}$ is called the *sketch basis*. Corresponding to a stream update of the form (i, v), the sketch is updated in time $O(1)$ as follows: X **:=** $X + x_i \cdot v$. It can be shown that $\mathrm{E}\big[X^2\big] = \mathrm{SJ}$ and $\mathrm{Var}\big[X^2\big] = O(\mathrm{SJ}^2)$. An ϵ-accurate estimate of the self-join is obtained by taking the average of $O(\frac{1}{\epsilon^2})$ independent sketches. The confidence of the estimate is boosted to $1 - \delta$ by using the standard technique of returning the median of $O(\log \frac{1}{\delta})$ independently computed averages.

Algorithm C*ountSketch* [8]. Sketches are used in [8] to design the *C*ountSketch algorithm for finding the top-k frequent items in a data stream. The data structure called CSK consists of a collection of s hash tables, $T[1], \ldots, T[s]$, each consisting of A buckets. A pair-wise independent hash function $h_t : \mathcal{D} \rightarrow \{0, 1, \ldots, A-1\}$ and a pair-wise independent sketch basis $\{x_{t,i}\}_{i \in \mathcal{D}}$ are associated with each hash table, $1 \leq t \leq s$. Each bucket, $T[t, b]$ keeps the sketch $X_{t,b} = \sum_{h_t(i)=b} f_i \cdot x_{t,i}$, of the sub-stream of the items that map to this bucket. In addition, an array capable of storing A pairs of the form $(i, \hat{f}_i)$ is kept and organized as a classical min-heap data structure. Corresponding to a stream update (i, v), the structure CSK is updated in time $O(s)$ as follows.

UPDATE$_{\mathrm{CSK}}(i, v)$: **for** t **:=** 1 **to** s **do** $X_{t,h_t(i)}$ **:=** $X_{t,h_t(i)} + v \cdot x_{t,i}$ **endfor**

Once all the hash tables are updated, the frequency f_i is estimated as

$$\hat{f}_i = \mathrm{median}_{t=1}^{s} X_{t,h_t(i)} \cdot x_{t,i} \quad . \tag{1}$$

If $\hat{f}_i$ exceeds the lowest value estimate in the heap H, then, the latter value is evicted and replaced by the pair $(i, \hat{f}_i)$. The estimation guarantees of the *C*ountSketch algorithm are stated as a function Δ of the residual self-join and is summarized below.

$$\Delta(s, A) = 8 \left(\frac{\mathrm{SJ}^{res}(s)}{A} \right)^{1/2} \tag{2}$$

Theorem 1 ([8]). *Let $s = O(\log \frac{m}{\delta})$, $A \geq 8 \cdot k$, and let $\Delta = \Delta(\frac{A}{8}, A)$. Then, for every item i,* Pr $\left\{ |\hat{f}_i - f_i| \leq \Delta \right\} \geq 1 - \frac{\delta}{2 \cdot m}$. *The space complexity is $O(k \cdot \log \frac{m}{\delta} \cdot (\log(m \cdot N))$ bits, and the time taken to process a stream update is $O(\log \frac{m}{\delta})$.* □

The FAST-AMS *[16] and* FAST-AGMS *algorithms* [10]. The FAST-AGMS algorithm is a time-efficient variant of the product of sketches technique for estimating join sizes. The *C*ountSketch based second moment estimator presented in [16] applies a similar optimization for reducing the processing time for estimating self-joins. The algorithm uses a pair of set of hash tables, $T_1, T_2, \ldots, T_s$ and $U_1, U_2, \ldots, U_s$ for streams R and S respectively, such that, each hash table consists of A buckets. The T and U hash tables are *parallel* in the sense that for $1 \le t \le s$, the tables T_t and U_t use the same random pair-wise independent hash function $h_t : \mathcal{D} \rightarrow \{0, 1, \ldots, A-1\}$ and the same four-wise independent sketch basis $\{x_{t,i}\}$. The random bits used for different hash table indices are independent of each other. For $1 \le t \le s$ and $0 \le b \le A-1$, each bucket $T_t[b]$ (resp. $U_t[b]$), contains a single sketch $X_{t,b}$ (resp. $Y_{t,b}$) of the sub-stream of items that hash to this bucket, that is, $X_{t,b} = \sum_{h_t(i)=b} f_i \cdot x_{t,i}$ (resp. $Y_{t,b} = \sum_{h_t(i)=b} g_i \cdot x_{t,i}$). Updates to the stream R or S are propagated to the corresponding data structure T or U appropriately, similar to the UPDATE$_{\text{CSK}}$ sub-routine given in Section 2. For each hash table index t, $1 \le t \le s$, an estimate $\hat{J}_t$ is obtained as follows: $\hat{J}_t = \sum_{b=0}^{A-1} X_{t,b} \cdot Y_{t,b}$. Finally, the median of these estimates is returned as the estimate of the join size, that is, $\hat{J} = \text{median}_{t=1}^{s} \hat{J}_t$. Lemma 1 summarizes the basic property of this algorithm.

Lemma 1 ([10, 16]). $\mathrm{E}\big[\hat{J}_t\big] = J$ *and* $\mathrm{Var}\big[\hat{J}_t\big] \le \frac{1}{A}\left(\mathrm{SJ}(R)\cdot\mathrm{SJ}(S) + J^2\right)$. *In particular, if* $R = S$*, then,* $\mathrm{E}\big[\hat{J}_t\big] = \mathrm{SJ}(R)$ *and* $\mathrm{Var}\big[\hat{J}\big] < \frac{2(\mathrm{SJ}(R))^2}{A}$. □

3 Algorithm REDSKETCH for join size estimation

In this section, we present the algorithm REDSKETCH for estimating the size of the join of data streams R and S for the insert-only stream model. The algorithm can be extended to insert-delete streams by using a variant of the *C*ountSketch algorithm that can handle deletions.

The data structure used by the algorithm is a pair of *parallel* *C*ountSketch structures denoted by CSK_R and CSK_S, for streams R and S respectively. The structures CSK_R and CSK_S use a pair of *parallel* hash table sets, $T[1], \ldots, T[s]$ for CSK_R and $U[1], \ldots, U[s]$ for CSK_S, respectively, each consisting of A buckets. The hash table sets in the sense that T_t and U_t use the same random pair-wise independent hash function h_t and the same four-wise independent sketch basis $x_{t,i}$. The updates to the structure are done as in the *C*ountSketch algorithm.

A join value i from stream R (resp. S) is said to be *frequent* in R (resp. S) provided its estimate $\hat{f}_i$ obtained using the frequency estimation procedure of *C*ountSketch(resp. $\hat{g}_i$) is among the top-k estimated frequencies in the stream R (resp. S).

Let F denote the set of join values that are frequent in either R or S. We decompose the join size J into two components as follows.

$$J_0 = \textstyle\sum_{i \in F} f_i \cdot g_i, \quad \text{and} \quad J_1 = \textstyle\sum_{i \notin F} f_i \cdot g_i.$$

The estimate $\hat{J}_0$ is obtained as $\hat{J}_0 = \sum_{i \in F} \hat{f}_i \cdot \hat{g}_i$. Next, we *reduce* the hash tables by deleting the estimated contribution of each frequent item $i \in F$ from the sketches contained in those buckets to which the item i hashes to.

$$X_{t,h_t(i)} := X_{t,h_t(i)} - \hat{f}_i \cdot x_{t,i} \; ; \; Y_{t,h_t(i)} := Y_{t,h_t(i)} - \hat{g}_i \cdot x_{t,i} \;\; \text{for } i \in F, 1 \le t \le s$$

We then multiply the corresponding buckets of the reduced hash table pair T_t and U_t and obtain an estimate for J_1 as the median of averages.

$$J'_t = \sum_{b=0}^{A-1} X_{t,b} \cdot Y_{t,b}, \quad \text{for } t = 1, 2, \ldots, s, \quad \text{and } \hat{J}_1 = \text{median}_{t=1}^{s} J'_t \ .$$

The join size is estimated as $\hat{J} = \hat{J}_0 + \hat{J}_1$. Theorem 2 presents the accuracy versus space guarantees of the algorithm.

Theorem 2. *For any $0 < \delta < 1$, $A = 64k$, and $s = O(\log \frac{m}{\delta})$, $\mathbf{Pr}\{|\hat{J} - J| \leq E\} \geq 1 - \delta$, where, $E = \frac{4}{\sqrt{k}}(m_R \cdot (\text{SJ}^{res}(S,k))^{1/2} + m_S \cdot (\text{SJ}^{res}(R,k))^{1/2} + \frac{J}{4\sqrt{k}}$.* □

If $A = 64k$, then, the space used by the algorithm is $O(k \cdot \log m \log \frac{m}{\delta})$ bits. The time taken to process each stream update is $O(\log \frac{m}{\delta})$ operations. We now prove Theorem 2.

Analysis. Let $\Delta_R = \Delta_R\left(\frac{A}{8}, A\right) = 8\left(\frac{(\text{SJ}^{res}(R,\frac{A}{8})}{A}\right)^{1/2}$ and $\Delta_S = 8\left(\frac{\text{SJ}^{res}(S,\frac{A}{8})}{A}\right)^{1/2}$. Let $\Gamma = (m_R(\text{SJ}^{res}(S,k)^{1/2} + m_S(\text{SJ}^{res}(S,k))^{1/2})$.

Lemma 2. *Let $A \geq 64k$. Then, (i) $\;(m_R\Delta_S + m_S\Delta_R) \leq \frac{2\Gamma}{\sqrt{k}}$,*
(ii) $\;(\text{SJ}^{res}(R,k))^{1/2}(\text{SJ}^{res}(S,k))^{1/2} \leq \frac{\Gamma}{8\sqrt{2k}}$ and (iii) $\;k\Delta_R\Delta_S \leq \frac{\Gamma}{8\sqrt{2k}}$.

Proof. We use the property that $\text{SJ}^{res}(R,k) \leq \frac{m_R^2}{4k}$.

(*i*) $m_R\Delta_R \leq \frac{8m_R(\text{SJ}^{res}(R,\frac{A}{8}))^{1/2}}{\sqrt{A}} \leq \frac{m_R(\text{SJ}^{res}(R,k))^{1/2}}{\sqrt{k}}$, since, $A \geq 64k$. Similarly $m_S\Delta_S \leq \frac{m_S\text{SJ}^{res}(S,k)}{\sqrt{k}}$). Adding, we obtain part (i).

(*ii*) $(\text{SJ}^{res}(R,k))^{1/2}(\text{SJ}^{res}(S,k))^{1/2} \leq \frac{m_R}{4\sqrt{2k}}(\text{SJ}^{res}(S,k))^{1/2}$. Similarly, $(\text{SJ}^{res}(R,k))^{1/2}(\text{SJ}^{res}(S,k))^{1/2} \leq (\text{SJ}^{res}(R,k))^{1/2}\frac{m_S}{4\sqrt{2k}}$). Therefore, adding, we have, $2(\text{SJ}^{res}(R,k))^{1/2}(\text{SJ}^{res}(S,k))^{1/2} \leq \frac{\Gamma}{4\sqrt{2k}}$.

(*iii*) Since, $k \leq \frac{A}{64} < \frac{A}{8}$, $\text{SJ}^{res}(R,\frac{A}{8}) \leq \text{SJ}^{res}(R,k)$ and $\text{SJ}^{res}(S,\frac{A}{8}) \leq \text{SJ}^{res}(S,k)$. Thus, $k\Delta_R\Delta_S \leq \frac{64k}{A}\left(\text{SJ}^{res}(R,k)\text{SJ}^{res}(S,k)\right)^{1/2} \leq \frac{\Gamma}{8\sqrt{2k}}$, by part(ii). □

Lemma 3. *Let $A = 64k$. Then, $|\hat{J}_0 - J_0| \leq (2 + \frac{1}{4\sqrt{2}})\frac{\Gamma}{\sqrt{k}}$ with probability $1 - \frac{\delta}{4}$.*

Proof. By Theorem 1, it follows that $|\hat{f}_i - f_i| \leq \Delta_R$, and $|\hat{g}_i - g_i| \leq \Delta_S$, with probability $1 - \frac{\delta}{8m}$. Since, $|F| \leq k + k = 2k$, therefore,

$$\begin{aligned}
|\hat{J}_0 - J_0| &\leq \sum_{i \in F} |\hat{f}_i\hat{g}_i - f_ig_i| \leq \sum_{i \in F}\left((f_i + \Delta_R)(g_i + \Delta_S) - f_ig_i\right) \\
&= \sum_{i \in F}\left(f_i\Delta_S + g_i\Delta_R + \Delta_R\Delta_S\right) \leq m_R\Delta_S + m_S\Delta_R + |F|\Delta_R\Delta_S \\
&\leq m_R\Delta_S + m_S\Delta_R + 2k\Delta_R\Delta_S \leq \left(\frac{2\Gamma}{\sqrt{k}} + \frac{\Gamma}{4\sqrt{2}\sqrt{k}}\right)
\end{aligned}$$

by Lemma 2, parts (i) and (iii). By union bound, the error probability is bounded by $\frac{\delta|F|}{8m} \leq \frac{\delta}{4}$. □

Defining the reduced frequency vector f' as follows.

$$f'_i = \begin{cases} f_i & \text{if } i \notin F \text{ (i.e., } i \text{ is not a frequent item)} \\ f_i - \hat{f}_i & \text{otherwise.} \end{cases} \tag{3}$$

Lemma 4. *Let $A = 64k$. Then, $|\mathrm{E}\big[J'_t\big] - J_1| \le \frac{\Gamma}{4\sqrt{2k}}$, with probability $1 - \frac{\delta}{4}$.*

Proof. By Lemma 1, $\mathrm{E}\big[J'_t\big] = \sum_{i\in\mathcal{D}} f'_i g'_i$. Thus,

$$\big|\mathrm{E}\big[J'_t\big] - J_1\big| = \big|\sum_{i\in\mathcal{D}} f'_i g'_i - \sum_{i\notin F} f_i g_i\big| \le \sum_{i\in F} |f_i - \hat{f}_i||g_i - \hat{g}_i| \le 2k\Delta_R\Delta_S \le \frac{\Gamma}{4\sqrt{2k}}$$

by Lemma 2, part(iii). The total error probability is bounded by $\frac{\delta|F|}{8m} \le \frac{\delta}{4}$. □

We now present an upper bound on the self-join size of the reduced frequencies. Let H denote the set of top-k items of a stream (say R) in terms of estimated frequencies.

Lemma 5. *Let $s_3 = O(\log \frac{m}{\delta})$. Then, $\sum_{i\notin H} f_i^2 \le \mathrm{SJ}^{res}(k)\left(1 + 32\left(\frac{k}{A}\right)^{1/2} + 256\frac{k}{A}\right)$, with probability at least $1 - \frac{\delta}{16}$.*

Proof. Let P be the set of the top-k items in terms of their true frequencies. Since P and H are sets of k values each, therefore, $|P - H| = |H - P|$ and we can map each value i of $P - H$ to a unique value i' of $H - P$ (arbitrarily). For any $i \in P - H$, $f_i \ge f_{i'}$ and $\hat{f}_i \le \hat{f}_{i'}$. Therefore, for any $i \in P - H$,

$$0 \le f_i - f_{i'} = (\hat{f}_{i'} - f_{i'}) + (f_i - \hat{f}_i) + (\hat{f}_i - \hat{f}_{i'}) \le (\hat{f}_{i'} - f_{i'}) + (\hat{f}_i - f_i) \ .$$

Taking absolute values, $|f_i - f_{i'}| \le |\hat{f}_{i'} - f_{i'}| + |\hat{f}_i - f_i| \le \Delta + \Delta = 2\Delta$, by Theorem 1 (with probability $1 - \frac{\delta}{8m}$ each). We therefore have,

$$\begin{aligned}
\sum_{i\notin H} f_i^2 &= \sum_{i\in P-H} f_i^2 + \sum_{i\notin(P\cup H)} f_i^2 \le \sum_{i'\in(H-P)} (f_{i'} + 2\cdot\Delta)^2 + \sum_{i\notin(P\cup H)} f_i^2 \\
&= \sum_{j\notin P} f_j^2 + 4\Delta \sum_{i'\in(H-P)} f_{i'} + 4\cdot|H-P|\cdot\Delta^2 \\
&= \mathrm{SJ}^{res}(k) + 4\Delta|H-P|^{1/2} \sum_{i'\in H-P} f_{i'}^2 + 4k\Delta^2 \\
&\le \mathrm{SJ}^{res}(k) + 4k^{1/2}\Delta(\mathrm{SJ}^{res}(k))^{1/2} + 4k\Delta^2 \\
&< \mathrm{SJ}^{res}(k)\left(1 + 32\left(\frac{k}{A}\right)^{1/2} + 256\frac{k}{A}\right) \square
\end{aligned}$$

Lemma 6. *Let $A = 64k$. Then, $\sum_{i\in\mathcal{D}} f_i'^2 < \frac{37}{4}\mathrm{SJ}^{res}(R,k)$ and $\sum_{i\in\mathcal{D}} g_i'^2 < \frac{37}{4}\mathrm{SJ}^{res}(S,k)$ with probability $1 - \frac{\delta}{16}$.*

Proof. Let F_R denote the top-k items in R in terms of estimated frequencies. Then,

$$\begin{aligned}\textstyle\sum_{i\in\mathcal{D}} f_i'^2 &= \textstyle\sum_{i\in F_R}(f_i-\hat{f}_i)^2+\sum_{i\notin F_R} f_i^2 \\ &\le k\Delta_R^2+\mathrm{SJ}^{res}(R,k)\left(1+\tfrac{32\sqrt{k}}{\sqrt{A}}+\tfrac{256k}{A}\right), \text{ by Lemma 5} \\ &= \tfrac{1}{4}\mathrm{SJ}^{res}(R,k)+\mathrm{SJ}^{res}(R,k)(1+\tfrac{32\sqrt{k}}{\sqrt{64k}}+\tfrac{256k}{64k})=\tfrac{37}{4}\mathrm{SJ}^{res}(R,k)\ .\square\end{aligned}$$

Lemma 7. *Let* $A=64k$. *Then,* $|\hat{J}_1-J_1|\le\frac{\Gamma}{\sqrt{k}}+\frac{J_1}{4\sqrt{k}}$ *with probability* $1-\frac{\delta}{4}$.

Proof. By Lemma 1, $\mathrm{Var}\big[J_t'\big]\le\frac{1}{A}\big((\sum_{i\in\mathcal{D}} f_i'^2)(\sum_{i\in\mathcal{D}} g_i'^2)+\frac{1}{A}(\mathrm{E}\big[J_t'\big])^2\big)$. Substituting from Lemma 6, we obtain that

$$\mathrm{Var}\big[J_t'\big]\le\frac{(37)^2}{16A}\mathrm{SJ}^{res}(R,k)\mathrm{SJ}^{res}(S,k)+\frac{1}{A}(\mathrm{E}\big[J_t'\big])^2)\le\frac{(37)^2\Gamma^2}{(16)(64)(128)k}+\frac{(\mathrm{E}\big[J_t'\big])^2}{64k}$$

by Lemma 2, part(ii) and substituting $A=64k$. Therefore, $(\mathrm{Var}\big[J_t'\big])^{1/2}\le\frac{37\Gamma}{256\sqrt{2k}}+\frac{\mathrm{E}\big[J_t'\big]}{8\sqrt{k}}$. By Lemma 4, $\mathrm{E}\big[J_t'\big]\le J_1+\frac{\Gamma}{4\sqrt{2k}}$. Adding, we have, $(\mathrm{Var}\big[J_t'\big])^{1/2}<\frac{37\Gamma}{256\sqrt{2k}}+\frac{\Gamma}{32k\sqrt{2}}+\frac{J_1}{8\sqrt{k}}$. By Chebychev's inequality $\Pr\left\{|J_t'-\mathrm{E}\big[J_t'\big]|\le 2(\mathrm{Var}\big[J_t'\big])^{1/2}\right\}\ge\frac{3}{4}$, or that $\Pr\left\{|J_t'-J_1|\right\}\le 2(\mathrm{Var}\big[J_t'\big])^{1/2}+\frac{\Gamma}{4\sqrt{2k}}$, with probability $\frac{3}{4}$. By a standard argument of boosting the confidence of taking medians, we obtain the statement of the lemma. $\square$

Proof (Of Theorem 2.). Adding the errors given by Lemmas 3 and 7 and the error probabilities , we obtain that $|\hat{J}-J|\le(2+\frac{1}{4\sqrt{2}})\frac{\Gamma}{\sqrt{k}}+\frac{\Gamma}{\sqrt{k}}+\frac{J_1}{4\sqrt{k}}<\frac{4\Gamma}{\sqrt{k}}+\frac{J}{4\sqrt{k}}$ with probability $1-\frac{\delta}{2}$. $\square$

4 Algorithm REDSKETCH-A

In this section, we present a variant of the REDSKETCH algorithm for estimating join sizes. The data structure used by the REDSKETCH-A algorithm is identical to that of the REDSKETCH algorithm; hence the space and the time complexity of algorithm REDSKETCH-A is the same as that of the REDSKETCH algorithm. Additionally, the REDSKETCH-A algorithm uses an estimator for the residual self-join size $\mathrm{SJ}^{res}(R,k)$ for any stream R which is presented below.

4.1 Estimating $\mathrm{SJ}^{res}(k)$

The estimator for $\mathrm{SJ}^{res}(k)=\mathrm{SJ}^{res}(R,k)$ uses a CountSketch data structure CSK consisting of $s_3=O(\log\frac{m}{\delta})$ independent hash tables, $T[1],\ldots,T[s_3]$, each consisting of $A=O(\frac{k}{\epsilon^2})$ buckets, as explained in Section 2. Let H denote the set of the top-k items in terms of the estimated frequencies. First, the contributions of the top-k estimated frequencies are removed from the corresponding sketches contained in the hash tables, that is, $X_{t,h_t(i)}$:=$X_{t,h_t(i)}-\hat{f}_i\cdot x_{t,i}$, for every $i\in H$ and $1\le t\le s_3$. Next, we obtain an estimate Z_t from each hash table index t as follows: $Z_t=\sum_{b=0}^{A-1}X_{t,b}^2$. Finally, we return the estimate $\hat{\mathrm{SJ}}^{res}(k)$ as the median of the Z_t's, that is, $\hat{\mathrm{SJ}}^{res}(k)=$

median$_{t=1}^{s_3} Z_t$. The accuracy guarantees are given by Theorem 3. The algorithm uses space $O\left(\frac{k}{\epsilon^2} \cdot \log \frac{m}{\delta} \cdot \log m\right)$ bits and processes each stream update in time $O(\log \frac{m}{\delta})$.

Theorem 3. *If $\epsilon \leq \frac{1}{8}$, $A \geq \frac{1600k}{\epsilon^2}$ and $s_3 = O(\log \frac{m}{\delta})$ then, $|\hat{\text{SJ}}^{res}(R,k) - \text{SJ}^{res}(k)| \leq \epsilon \text{SJ}^{res}(k)$, with probability $1-\delta$.*

Proof. Let $f'_i = (f_i - \hat{f}_i)$, if $i \in H$, and $f'_i = f_i$, for $i \notin H$. Define $\text{SJ}^{\text{suffix}}(k) = \sum_i f_i'^2$. Note that the estimator $\hat{\text{SJ}}^{res}$ returns an approximation of $\text{SJ}^{\text{suffix}}(k)$ using the FAST-AMS algorithm. Let $\Delta = \Delta_R$. By property of CountSketch algorithm, $|\hat{f}_i - f_i| \leq \Delta$, with probability $1 - \frac{\delta}{8m}$.

$$\text{SJ}^{\text{suffix}}(k) = \sum_{i \in H}(f_i - \hat{f}_i)^2 + \sum_{i \notin H} f_i^2 \leq k \cdot \Delta^2 + \sum_{i \notin H} f_i^2$$

$$\leq \text{SJ}^{res}(k)\left(1 + \frac{32\sqrt{k}}{\sqrt{A}} + \frac{320k}{A}\right), \text{ by Lemma 5.}$$

Further, $\text{SJ}^{\text{suffix}} \geq \sum_{i \notin H} f_i^2 \geq \sum_{i \notin P} f_i^2 \geq \text{SJ}^{res}(k)$.

By Lemma 1, $\text{E}\left[Z_t\right] = \text{SJ}^{\text{suffix}}(k)$ and $\text{Var}\left[Z_t\right] \leq \frac{2}{A}(\text{SJ}^{\text{suffix}}(k))^2$. Therefore, Chebychev's inequality, $|Z_t - \text{SJ}^{\text{suffix}}(k)| \leq \frac{2}{\sqrt{A}}\text{SJ}^{\text{suffix}}(k)$ occurs with probability at least $\frac{3}{4}$. Therefore, by boosting the confidence by returning the median $\hat{\text{SJ}}^{res}(k)$ of the Z_t's, we have, $\hat{\text{SJ}}^{res}(k) \in (1 \pm \frac{2}{\sqrt{A}})\text{SJ}^{\text{suffix}}(k)$. Therefore, $\left(1 - \frac{2}{\sqrt{A}}\right)\text{SJ}^{res}(k) \leq \hat{\text{SJ}}^{res}(k) \leq \text{SJ}^{res}(k)\left(1 + \frac{32\sqrt{k}}{\sqrt{A}} + \frac{320k}{A}\right)(1 + \frac{2}{\sqrt{A}})\text{SJ}^{res}(k)$. Substituting $A \geq \frac{1600}{\epsilon^2}$ and $\epsilon \leq \frac{1}{8}$ gives $(1-\epsilon)\text{SJ}^{res}(k) \leq \hat{\text{SJ}}^{res}(k) \leq (1+\epsilon)\text{SJ}^{res}(k)$. □

4.2 Estimating join size using algorithm REDSKETCH-A

The REDSKETCH-A algorithm first estimates $\text{SJ}^{res}(R,k)$ and $\text{SJ}^{res}(S,k)$ as $\hat{\text{SJ}}^{res}(R,k)$ and $\hat{\text{SJ}}^{res}(S,k)$ respectively, to within factors of $1 \pm \frac{1}{8}$ with probability $1 - \frac{\delta}{32}$, each, using the algorithm given above. Let $\hat{\Delta}_R$ denote $8\left(\frac{\hat{\text{SJ}}^{res}(R, \frac{A}{8})}{A}\right)^{1/2}$ and $\hat{\Delta}_S$ denote $8\left(\frac{\hat{\text{SJ}}^{res}(S, \frac{A}{8})}{A}\right)^{1/2}$. The algorithm uses the following notion of frequent items.

Definition 1. *A join value i from the stream R (resp. S) is said to be frequent in R (resp. S), provided, (a) $\hat{f}_i \geq \gamma \hat{\Delta}_R$ (resp. $\hat{g}_i \geq \gamma \hat{\Delta}_S$), and, (b) $\hat{f}_i$ is among the top-k estimated frequencies in the stream R (resp. S), where, $\gamma = \frac{6}{5}\left(1 + \frac{2}{\epsilon}\right)$.* □

The value of ϵ used in Definition 1 is a parameter. Let F_R (resp. F_S) denote the set of join values that are frequent in R (resp. S) and let F denote $F_R \cup F_S$. Following the paradigm of the bifocal method [13], we decompose the join size J into four components, namely, $J = J_{d,d} + J_{d,s} + J_{s,d} + J_{s,s}$, where, $J_{d,d} = \sum_{i \in F_R \cap F_S} f_i g_i$, $J_{s,s} = \sum_{i \notin (F_R \cup F_S)} f_i g_i$, $J_{d,s} = \sum_{i \in F_R - F_S} f_i g_i$ and $J_{s,d} = \sum_{i \in F_S - F_R} f_i g_i$. The estimate $\hat{J}_{d,d}$ for $J_{d,d}$ is obtained as usual: $\hat{J}_{d,d} = \sum_{i \in F_R \cap F_S} \hat{f}_i \cdot \hat{g}_i$. Next, we *reduce* the hash table structure as follows. For every hash table index t, $1 \leq t \leq s_3$, we perform the following operations.

$$\begin{aligned} X_{t,h_t(i)} &:= X_{t,h_t(i)} - \hat{f}_i \cdot x_{t,i}, \quad \text{for each } i \in F_R, \text{ and} \\ Y_{t,h_t(i)} &:= Y_{t,h_t(i)} - \hat{g}_i \cdot x_{t,i}, \quad \text{for each } i \in F_S \end{aligned}$$

We then obtain the estimates $\hat{J}_{d,s,t}$ and $\hat{J}_{s,d,t}$ from each hash table index t, $1 \le t \le s_3$, as follows.

$$\hat{J}_{d,s,t} = \sum_{b=0}^{A-1} Y_{t,b} \cdot \big(\sum_{i \in F_R : h_t(i) = b} \hat{f}_i \cdot x_{t,i} \big), \quad \hat{J}_{s,d,t} = \sum_{b=0}^{A-1} X_{t,b} \cdot \big(\sum_{i \in F_S : h_t(i) = b} \hat{g}_i \cdot x_{t,i} \big)$$

The estimates $\hat{J}_{d,s}$ and $\hat{J}_{s,d}$ are obtained as the medians of the estimates $\hat{J}_{d,s,t}$ and $\hat{J}_{s,d,t}$ respectively. That is,

$$\hat{J}_{d,s} = \text{median}_{t=1}^{s_3} \hat{J}_{d,s,t}, \qquad \text{and} \qquad \hat{J}_{s,d} = \text{median}_{t=1}^{s_3} \hat{J}_{s,d,t} \ .$$

The estimates $\hat{J}_{s,s,t}$, $1 \le t \le s_3$ and the median estimate $\hat{J}_{s,s}$ is obtained in a manner identical to J'_t and $\hat{J}_1$ in the REDSKETCH algorithm, as follows.

$$\hat{J}_{s,s,t} = \sum_{b=0}^{A-1} X_{t,b} \cdot Y_{t,b}, \;\; 1 \le t \le s_3, \quad \text{and} \quad \hat{J}_{s,s} = \text{median}_{t=1}^{s_3} \hat{J}_{s,s,t}$$

Finally, the estimate $\hat{J}$ for the join size is obtained as the sum of the estimates, that is, $\hat{J} = \hat{J}_{d,d} + \hat{J}_{d,s} + \hat{J}_{s,d} + \hat{J}_{s,s}$. The space versus accuracy properties of the algorithm is stated in Theorem 4 and proved below. $\Lambda = (\text{SJ}(R)\text{SJ}^{res}(S,k))^{1/2} + (\text{SJ}^{res}(R,k)\text{SJ}(S))^{1/2}$.

Theorem 4. *Let $A \ge 64k$. Then,* Pr $\left\{ |\hat{J} - J| \le E \right\} \ge 1 - \delta$, *where,* $E = \min\big(\frac{32\Lambda}{\sqrt{k}} + \frac{J}{2} + \frac{J}{\sqrt{k}}\big), \;\; 2J^{2/3}\big(\frac{2\Lambda}{\sqrt{k}}\big)^{1/3}\big)$. □

Analysis. Let $\gamma = \frac{6}{5}\left(1 + \frac{2}{\epsilon}\right)$ (as given by Definition 1), $\gamma_1 = \frac{5}{6}\gamma$ and $\gamma_2 = \frac{6}{5}\gamma$. Since, $\hat{\text{SJ}}^{res}(R,k) \ge \frac{3}{4}\text{SJ}^{res}(R,k)$, with probability $1 - \frac{\delta}{8m}$, therefore, $\left(\frac{3}{4}\right)^{1/2} \Delta(R,k) \le \hat{\Delta}(R,k) \le \left(\frac{4}{3}\right)^{1/2} \Delta(R,k)$, which implies that, $\gamma_1 \Delta(R,k) \le \hat{\Delta}(R,k) \le \gamma_2 \Delta(R,k)$. Similarly, $\gamma_1 \Delta(S,k) \le \hat{\Delta}(S,k) \le \gamma_2 \Delta(S,k)$, each with probability $1 - \frac{\delta}{8m}$.

Lemma 8. *Suppose i is a frequent item in R. Then, $f_i \ge (\gamma_1 - 1)\Delta_R$ and $|\hat{f}_i - f_i| \le \epsilon f_i$, with probability $1 - \frac{\delta}{8m}$. Otherwise, $f_i < (\gamma_2 + 1)\Delta(R,k)$, with probability $1 - \frac{\delta}{8m}$.*

Proof. By Definition 1, $\hat{f}_i \ge \gamma_1 \Delta_R$. Therefore, with probability $1 - \frac{\delta}{8m}$, $f_i \ge (\gamma_1 - 1)\Delta_R$. Further, $\frac{|\hat{f}_i - f_i|}{f_i} \le \frac{\Delta_R}{\gamma_1 - 1} \le \epsilon$. If $i \notin F_R$, then, $\hat{f}_i < \gamma_1 \hat{\Delta}(R,k) \le \gamma_2 \Delta(R,k)$. Therefore, with probability $1 - \frac{\delta}{8m}$, $f_i < (\gamma_2 + 1)\Delta(R,k)$. □

Lemma 9. *Let $\epsilon \le 1$. Then, $|\hat{J}_{d,d} - J_{d,d}| \le \frac{5\epsilon}{4} J_{d,d}$, with probability $1 - \frac{\delta}{8}$.*

Proof. $|\hat{J}_{d,d} - J_{d,d}| \le \sum_{i \in F_R \cap F_S} |\hat{f}_i \hat{g}_i - f_i g_i| \le \sum_{i \in F_R \cap F_S} f_i g_i ((1 + \frac{\epsilon}{2})^2 - 1) \le \frac{5\epsilon}{4} J_{d,d}$. Since, $|F_R \cap F_S| \le k$, the total error probability, is at most $\frac{\delta k}{8m} \le \frac{\delta}{8}$. □

The reduced frequencies are defined as before, namely: $f'_i = f_i$ if $i \notin F_R$, and $f'_i = f_i - \hat{f}_i$, otherwise; and analogously for S: $g'_i = g_i$ if $i \notin F_S$, and $g'_i = g_i - \hat{g}_i$, otherwise.

Lemma 10. $\left|\mathrm{E}\left[\hat{J}_{d,s,t}\right] - J_{d,s}\right| \le \frac{\epsilon}{2} J_{d,s} + \frac{9\epsilon}{16} J_{d,d}$ *and* $\left|\mathrm{E}\left[\hat{J}_{s,d,t}\right] - J_{s,d}\right| \le \frac{\epsilon}{2} J_{s,d} + \frac{9\epsilon}{16} J_{d,d}$*, each with probability* $1 - \frac{\delta}{8}$.

Proof. $J_{d,s} = \sum_{i \in F_R - F_S} f_i g_i$. By Lemma 1, $\mathrm{E}\left[\hat{J}_{d,s,t}\right] = \sum_{i \in F_R} \hat{f}_i g'_i$. Therefore,

$$|\mathrm{E}\left[\hat{J}_{d,s,t}\right] - J_{d,s}| = |\sum_{i \in F_R} \hat{f}_i g'_i - \sum_{i \in F_R - F_S} f_i g_i| = |\sum_{i \in F_R \cap F_S} \hat{f}_i g'_i + \sum_{i \in F_R - F_S} (\hat{f}_i - f_i) g_i|$$

If $i \in F_R \cap F_S$, then, $|\hat{f}_i - f_i| \le \frac{\epsilon f_i}{2}$, by Lemma 8, and $|g'_i| \le |\hat{g}_i - g_i| \le \frac{\epsilon g_i}{2}$, by Lemma 8. Adding, $|\sum_{i \in F_R \cap F_S} \hat{f}_i g'_i| \le \sum_{i \in F_R \cap F_S} (1 + \frac{\epsilon}{2}) \frac{\epsilon}{2} f_i g_i \le \frac{9\epsilon}{16} J_{d,d}$. If $i \in F_R - F_S$, then, $|\hat{f}_i - f_i| \le \frac{\epsilon f_i}{2}$, by Lemma 8. Therefore, $|\sum_{i \in F_R - F_S} (\hat{f}_i - f_i) g_i| \le \sum_{i \in F_R - F_S} \frac{\epsilon f_i}{2} g_i = \frac{\epsilon}{2} J_{d,s}$. Adding, we obtain the statement of the lemma. The proof for $J_{s,d}$ is analogous. □

Lemma 11. $\left|\mathrm{E}\left[\hat{J}_{s,s,t}\right] - J_{s,s}\right| \le \epsilon^2 J_{d,d} + \epsilon(J_{d,s} + J_{s,d})$*, with probability* $1 - \frac{\delta}{4}$.

Proof. $\left|\mathrm{E}\left[\hat{J}_{s,s,t}\right] - J_{s,s}\right| = \left|\sum_{i \in \mathcal{D}} f'_i g'_i - \sum_{i \notin (F_R \cup F_S)} f_i g_i\right|$

$\le \sum_{i \in F_R \cap F_S} |f_i - \hat{f}_i||g_i - \hat{g}_i| + \sum_{i \in F_R - F_S} |f_i - \hat{f}_i| g_i + \sum_{i \in F_S - F_R} f_i |g_i - \hat{g}_i|$

$\le \epsilon^2 J_{d,d} + \epsilon(J_{d,s} + J_{s,d})$. □

Lemma 12. *If* $A = 64k$*, then,* $\sum_{i \in F_R} \hat{f}_i^2 \le \frac{9}{4} \mathrm{SJ}(R)$ *and* $\sum_{i \in F_S} \hat{g}_i^2 \le \frac{9}{4} \mathrm{SJ}(S)$.

Proof. Using $(a+b)^2 \le 2(a^2 + b^2)$, we have,

$$\sum_{i \in F_R} \hat{f}_i^2 \le \sum_{i \in F_R} (f_i + \Delta_R)^2 \le 2 \sum_{i \in F_R} f_i^2 + 2k\Delta_R^2$$
$$\le 2\mathrm{SJ}(R) + \frac{16k}{A} \mathrm{SJ}^{res}(R, \frac{A}{8}) \le \frac{5}{2} \mathrm{SJ}(R). \qquad \square$$

Lemma 13. *If* $A \ge 64k$ *and* $\epsilon \le \frac{1}{4}$*, then,* $\sum_{i \in \mathcal{D}} f_i'^2 \le \frac{5}{4\epsilon^2} \mathrm{SJ}^{res}(R, k)$ *and* $\sum_{i \in \mathcal{D}} g_i'^2 \le \frac{5}{4\epsilon^2} \mathrm{SJ}^{res}(S, k)$*, with high probability* $(1 - \frac{\delta}{8})$.

Proof. Suppose that $|F_R| = l$. Consider the item whose rank is $l + 1$. This item must have frequency at most $\gamma \hat{\Delta}_R + \Delta_R \le (\gamma_2 + 1)\Delta_R$, otherwise, its estimate would have crossed the frequent item threshold $\gamma \hat{\Delta}_R$ (with probability $1 - \frac{\delta}{8m}$), and it, along with the l higher ranked items would all have been included in the frequent item set F_R. This would make $|F_R| \ge l + 1$. Thus,

$$\mathrm{SJ}^{res}(R, l) \le (k - l)((\gamma_2 + 1)\Delta_R)^2 + \mathrm{SJ}^{res}(R, k)$$
$$\le \frac{10(k-l)}{\epsilon^2} \Delta_R^2 + \mathrm{SJ}^{res}(R, k) \ \le \left(1 + \frac{5(k-l)}{4\epsilon^2 k}\right) \mathrm{SJ}^{res}(R, k)$$

$\sum_{i \in \mathcal{D}} f_i'^2 = \sum_{i \in F_R} (f_i - \hat{f}_i)^2 + \sum_{i \notin F_R} f_i^2 \le l\Delta_R^2 + \sum_{i \notin F_R} f_i^2 \le \frac{l}{8k} \mathrm{SJ}^{res}(R, k) + \sum_{i \notin F_R} f_i^2$, with probability at least $1 - \frac{l\delta}{8m}$. By Lemma 5, $\sum_{i \notin F_R} f_i^2 \le \mathrm{SJ}^{res}(R, l)\big(1 + \frac{32\sqrt{l}}{\sqrt{A}} + \frac{256l}{A}\big)$. Adding,

$$\sum_{i \in \mathcal{D}} f_i'^2 \le \mathrm{SJ}^{res}(R, k)\big(\frac{l}{8k} + \big(1 + \frac{5(k-l)}{4\epsilon^2 k}\big)\big(1 + \frac{32\sqrt{l}}{\sqrt{A}} + \frac{256l}{A}\big)\big) \le \frac{5}{4\epsilon^2} \mathrm{SJ}^{res}(R, k) \ .\square$$

Recall that $\Lambda = (\mathrm{SJ}(R)\mathrm{SJ}^{res}(S,k))^{1/2} + (\mathrm{SJ}^{res}(R,k)\mathrm{SJ}(S))^{1/2}$.

Proof (Of Theorem 4). By Lemma 1, $\mathrm{Var}\big[\hat{J}_{d,s,t}\big] \le \frac{1}{A}(\sum_{i \in F_R} \hat{f}_i^2)(\sum_{i \in \mathcal{D}} g_i'^2) + \frac{1}{A}(\mathrm{E}\big[J_{d,s,t}\big])^2$. By Lemmas 12 and 13, $\frac{1}{A}(\sum_{i \in F_R} \hat{f}_i^2)(\sum_{i \in \mathcal{D}} g_i'^2) \le \frac{45}{16\epsilon^2 A}\mathrm{SJ}(R) \cdot \mathrm{SJ}^{res}(S,k) \le \frac{\Lambda^2}{20\epsilon^2 k}$. By Lemma 10, $\mathrm{E}\big[\hat{J}_{d,s,t}\big] \le (J_{d,s} + \frac{9\epsilon}{16}(J_{d,d} + J_{d,s}))$. By Chebychev's inequality, $|\hat{J}_{d,s,t} - \mathrm{E}\big[J_{d,s,t}\big]| \;|\le 3(\mathrm{Var}\big[\hat{J}_{d,s,t}\big])^{1/2}$ with probability at least $\frac{8}{9}$. The median $\hat{J}_{d,s}$ satisfies the same relation with probability $1 - \frac{\delta}{4}$. Therefore, using triangle inequality,

$$\begin{aligned} |\hat{J}_{d,s} - J_{d,s}| &\le 3(\mathrm{Var}\big[\hat{J}_{d,s,t}\big])^{1/2} + |\mathrm{E}\big[\hat{J}_{d,s,t}\big] - J_{d,s}| \\ &\le \frac{3\Lambda}{\epsilon\sqrt{20}} + \frac{3J_{d,s}}{8\sqrt{k}} + \frac{9\epsilon}{16}(1 + \frac{3}{8\sqrt{k}})(J_{d,d} + J_{d,s}) \end{aligned}$$

Analogously, it can be shown that

$$|\hat{J}_{s,d} - J_{s,d}| \le \frac{3\Lambda}{\epsilon\sqrt{20}} + \frac{3J_{s,d}}{8\sqrt{k}} + \frac{9\epsilon}{16}(1 + \frac{3}{8\sqrt{k}})(J_{d,d} + J_{s,d}) \;\; .$$

By Lemma 1, $\mathrm{Var}\big[\hat{J}_{s,s,t}\big] \le \frac{1}{A}(\sum_{i \in \mathcal{D}} f'^2)(\sum_{j \in \mathcal{D}} g'^2) + \frac{1}{A}(\mathrm{E}\big[\hat{J}_{s,s,t}\big])^2$. Using Lemmas 13 and 11 and following a similar reasoning as above, it can be shown that $\mathrm{Var}\big[\hat{J}_{s,s,t}\big] \le \frac{\Lambda^2}{40\epsilon^4 k} + \frac{(\mathrm{E}\big[\hat{J}_{s,s,t}\big])^2}{64k}$, and therefore, the median $\hat{J}_{s,s}$ satisfies

$$|\hat{J}_{s,s} - J_{s,s}| \le \frac{\Lambda}{\sqrt{40}\epsilon^2 k} + (\epsilon^2 J_{d,d} + \epsilon(J_{d,s} + J_{s,d}))(1 + \frac{3}{8\sqrt{k}}) + \frac{2J_{s,s}}{8\sqrt{k}}$$

with probability $1 - \frac{\delta}{8}$. By Lemma 9, $|\hat{J}_{d,d} - J_{d,d}| \le \frac{5\epsilon}{4} J_{d,d} \le \frac{5\epsilon}{4} J$, with probability $1 - \frac{\delta}{8}$. Adding the errors and error probabilities, and using that $\epsilon \le \frac{1}{4}$, we have, $|\hat{J} - J| \le \frac{\Lambda}{2\epsilon^2\sqrt{k}} + (4\epsilon + \frac{2}{\sqrt{k}})J$, with probability $1 - \frac{\delta}{2}$.

The above property holds for all values of $\epsilon \le \frac{1}{4}$. Therefore, we can find the value of ϵ that minimizes the above function. Doing so, we obtain $\epsilon = \left(\frac{\Lambda}{4J\sqrt{k}}\right)^{1/3}$ and substituting this value yields the statement of the theorem. □

5 Conclusions

In this paper, we present novel, space and time efficient algorithms for estimating the join size of two data streams consisting of general insertion and deletion operations.

References

1. Noga Alon, Phillip B. Gibbons, Yossi Matias, and Mario Szegedy. "Tracking Join and Self-Join Sizes in Limited Storage". In *Proceedings of the Eighteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, Philadeplphia, Pennsylvania, May 1999.
2. Noga Alon, Yossi Matias, and Mario Szegedy. "The Space Complexity of Approximating the Frequency Moments". In *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing STOC, 1996*, pages 20–29, Philadelphia, Pennsylvania, May 1996.

3. Noga Alon, Yossi Matias, and Mario Szegedy. "The space complexity of approximating frequency moments". *Journal of Computer Systems and Sciences*, 58(1):137–147, 1998.
4. A. Arasu, B. Babcock, S. Babu, J. Cieslewicz, M. Datar, K. Ito, R Motwani, U. Srivastava, and J. Widom. "STREAM: The Stanford Data Stream Management System". In *Data Stream Management Processing High-Speed Data Streams Series: Data-Centric Systems and Applications,* Minos Garofalakis, Johannes Gehrke and Rajeev Rastogi *(Eds.) 2006, ISBN: 3-540-28607-1, Springer*.
5. Ron Avnur and Joseph M. Hellerstein. "Eddies: Continuously Adaptive Query Processing". In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, Dallas, Texas, USA, 2000.
6. Brian Babcock, Shivnath Babu, Mayur Datar, Rajeev Motwani, and Jennifer Widom. "Models and Issues in Data Stream Systems". In *Proceedings of the Twentysecond ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, Madison, Wisconsin, USA, 2002.
7. Donald Carney, Ugur Çetintemel, Mitch Cherniack, Christian Convey, Sangdon Lee, Greg Seidman, Michael Stonebraker, Nesime Tatbul, and Stanley B. Zdonik. "Monitoring Streams - A New Class of Data Management Applications". In *Proceedings of the 28th International Conference on Very Large Data Bases*, Hong Kong, China, 2002.
8. Moses Charikar, Kevin Chen, and Martin Farach-Colton. "Finding frequent items in data streams". In *Proceedings of the 29th International Colloquium on Automata Languages and Programming*, 2002.
9. G. Cormode and S. Muthukrishnan. "An improved data stream summary: The Count-Min sketch and its applications". In *Proceedings of the 6th Latin American Symposium on Informatics LATIN, Lecture Notes in Computer Science 2976 Springer 2004, ISBN 3-540-21258-2*, pages 29–38, Buenos Aires, Argentina, April 2004.
10. Graham Cormode and Minos Garofalakis. "Sketching Streams Through the Net: Distributed Approximate Query Tracking". In *Proceedings of the 31st International Conference on Very Large Data Bases*, September 2005.
11. Alin Dobra, Minos N. Garofalakis, Johannes Gehrke, and Rajeev Rastogi. "Processing complex aggregate queries over data streams". In *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data*, Madison, Wisconsin, USA, 2002.
12. Sumit Ganguly, Minos Garofalakis, and Rajeev Rastogi. "Processing Data Stream Join Aggregates using Skimmed Sketches". In *Proceedings of the Ninth International Conference on Extending Database Technology*, Herkailon, Crete, Greece, 2004.
13. Sumit Ganguly, Phil Gibbons, Yossi Matias, and Avi Silberschatz. "Bifocal Sampling for Skew-Resistant Join Size Estimation". In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, Montreal, Quebec, June 1996.
14. Wen-Chi Hou, Gultekin Ozsoyoglu, and Baldeo K. Taneja. "Statistical estimators for relational algebra expressions". In *Proceedings of the Seventh ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 276–287, Philadelphia, Pennsylvania, March 1988.
15. Richard Lipton, Jeffrey Naughton, and Donovan Schneider. "Practical Selectivity Estimation Through Adaptive Sampling". In *Proceedings of the 1990 ACM SIGMOD International Conference on Management of Data*, Atlantic City, NJ, 1990.
16. Mikkel Thorup and Yin Zhang. "Tabulation based 4-universal hashing with applications to second moment estimation". In *Proceedings of the Fifteenth ACM SIAM Symposium on Discrete Algorithms*, pages 615–624, New Orleans, Louisiana, USA, January 2004.