Streaming Algorithms via Precision Sampling*

Alexandr Andoni†
Microsoft Research
Mountain View, CA, USA
andoni@microsoft.com

Robert Krauthgamer‡
Weizmann Institute of Sciences
Rehovot, Israel
robi@weizmann.ac.il

Krzysztof Onak§
CMU
Pittsburgh, PA, USA
konak@cs.cmu.edu

Abstract—A technique introduced by Indyk and Woodruff (STOC 2005) has inspired several recent advances in data-stream algorithms. We show that a number of these results follow easily from the application of a single probabilistic method called *Precision Sampling*. Using this method, we obtain simple data-stream algorithms that maintain a randomized sketch of an input vector $x = (x_1, x_2, \ldots, x_n)$, which is useful for the following applications:

- Estimating the F_k-moment of x, for $k > 2$.
- Estimating the ℓ_p-norm of x, for $p \in [1, 2]$, with small update time.
- Estimating cascaded norms $\ell_p(\ell_q)$ for all $p, q > 0$.
- ℓ_1 sampling, where the goal is to produce an element i with probability (approximately) $|x_i|/\|x\|_1$. It extends to similarly defined ℓ_p-sampling, for $p \in [1, 2]$.

For all these applications the algorithm is essentially the same: scale the vector x entry-wise by a well-chosen random vector, and run a heavy-hitter estimation algorithm on the resulting vector. Our sketch is a linear function of x, thereby allowing general updates to the vector x.

Precision Sampling itself addresses the problem of estimating a sum $\sum_{i=1}^{n} a_i$ from weak estimates of each real $a_i \in [0, 1]$. More precisely, the estimator first chooses a desired precision $u_i \in (0, 1]$ for each $i \in [n]$, and then it receives an estimate of every a_i within additive u_i. Its goal is to provide a good approximation to $\sum a_i$ while keeping a tab on the "approximation cost" $\sum_i (1/u_i)$. Here we refine previous work (Andoni, Krauthgamer, and Onak, FOCS 2010) which shows that as long as $\sum a_i = \Omega(1)$, a good multiplicative approximation can be achieved using total precision of only $O(n \log n)$.

Keywords-streaming, sampling, moments, cascaded norms

1. Introduction

A number of recent developments in algorithms for data streams have been inspired, at least in part, by a technique devised by Indyk and Woodruff [21] to obtain near-optimal space bounds for estimating F_k moments, for $k > 2$. Indeed, refinements and modifications of that technique were used for designing better or new algorithms for applications such as: F_k moments [6] (with better bounds than [21]), entropy estimation [5], cascaded norms [18], [23], Earth-mover Distance [2], ℓ_1 sampling algorithm [29], distance to independence of two random variables [7], and even, more generically, a characterization of "sketchable" functions of frequencies [9]. While clearly very powerful, the Indyk-Woodruff technique is somewhat technically involved, and hence tends to be cumbersome to work with.

In this paper, we show an alternative design for the Indyk-Woodruff technique, resulting in a simplified algorithm for several of the above applications. Our key ingredient, dubbed the *Precision Sampling Lemma (PSL)*, is a probabilistic method, concerned with estimating the sum of a number of real quantities. The PSL was introduced in [3, Lemma 3.12], in an unrelated context, of *query-efficient* algorithms (in the sense of property testing) for estimating the edit distance.

Our overall contribution here is providing a generic approach that leads to simplification and unification of a family of data-stream algorithms. Along the way we obtain new and improved bounds for some applications. We also give a slightly improved version of the PSL.

In fact, all our algorithms comprise of the following two simple steps: multiply the stream by well-chosen random numbers (given by PSL), and then solve a certain heavy-hitters problem. Interestingly, each of the two steps (separately) either has connections to or is a well-studied problem in the literature of data streams. Namely, our implementation of the first step is somewhat similar to *Priority Sampling* [16], as discussed in Section 1.3. The second step, the heavy-hitters problem, is a natural streaming primitive, studied at least since the work of Misra and Gries [28]. It would be hard to list all the relevant literature for this problem concisely; instead we refer the reader, for example, to the survey by Muthukrishnan [30] and the CountMin wiki site [13] and the references therein.

1.1. Streaming Applications

We now describe the relevant streaming applications in detail. In most cases, the input is a vector $x \in \mathbb{R}^n$, which we maintain under stream updates. An update has the form (i, δ), which means that $\delta \in \mathbb{R}$ is added to x_i, the ith

*The full paper is available at http://arxiv.org/abs/1011.1263.

†Work done in part while the author was a postdoctoral researcher at Princeton University/CCI, supported by NSF CCF 0832797.

‡Supported in part by The Israel Science Foundation (grant #452/08), and by a Minerva grant.

§Supported in part by a Simons Postdoctoral Fellowship and NSF grants 0732334 and 0728645.

coordinate of x.[1] The goal is to maintain a sketch of x of small size (much smaller than n), such that, at the end of the stream, the algorithm outputs some function of x, depending on the actual problem in mind. Besides the space usage, another important complexity measure is the update time — how much time it takes to modify the sketch to reflect an update (i, δ).

We study the following problems.[2] For all these problems, the algorithm is essentially the same (see the beginning of Section 3). All space bounds are in terms of words, each having $O(\log n)$ bits.

- **$\mathbf{F_k}$ moment estimation, for $\mathbf{k > 2}$:** The goal is to produce a $(1+\epsilon)$ factor approximation to the k-th moment of x, i.e., $\|x\|_k^k = \sum_{i=1}^n |x_i|^k$. The first sublinear-space algorithm for $k > 2$, due to [1], gave a space bound $n^{1-1/k} \cdot (\epsilon^{-1} \log n)^{O(1)}$, and further showed the first polynomial lower bound for k sufficiently large. A lower bound of $\Omega(n^{1-2/k})$ was shown in [10], [4], and it was (nearly) matched by Indyk and Woodruff [21], who gave an algorithm using space $n^{1-2/k} \cdot (\epsilon^{-1} \log n)^{O(1)}$. Further research reduced the space bound to essentially $O(n^{1-2/k} \cdot \epsilon^{-2-4/k} \log^2 n)$ [6], [29] (see [29] for multi-pass bounds). Independently of our work, this bound was improved by a roughly $O(\log n)$ factor in [8].
  Our algorithm for this problem appears in Section 3.1, and improves the space usage over these bounds. Very recently, following the framework introduced here, [17] reports a further improvement in space for a certain regime of parameters.
- **$\ell_\mathbf{p}$-norm estimation, for $\mathbf{p} \in [\mathbf{1}, \mathbf{2}]$:** The goal is to produce a $1+\epsilon$ factor approximation to $\|x\|_p$, just like in the previous problem.[3] The case $p = 2$, i.e., ℓ_2-norm estimation was solved in [1], which gives a space bound of $O(\epsilon^{-2} \log n)$. It was later shown in [20] how to estimate ℓ_p norm for all $p \in (0, 2]$, using p-stable distributions, in $O(\epsilon^{-2} \log n)$ space. Further research aimed to get a tight bound and to reduce the update time (for small ϵ) from $\Omega(\epsilon^{-2})$ to $\log^{O(1)} n$ (or even $O(1)$ for $p = 2$), see, e.g., [31], [26], [27], [19] and references therein.
  Our algorithm for this problem appears in Section 3.2 for $p = 1$ and Section 4.1 for all $p \in [1, 2]$. The algorithm has an improved update time, over that of [19], for $p \in (1, 2]$, and uses comparable space, $O(\epsilon^{-2-p} \log^2 n)$. We note that, for $p = 1$, our space bound is worse than that of [31]. Independently of our work, fast space-optimal algorithms for all $p \in (0, 2)$ were recently obtained in [25].
- **Mixed/cascaded norms:** The input is a matrix $x \in \mathbb{R}^{n \times n}$, and the goal is to estimate the $\ell_p(\ell_q)$ norm, defined as $\|x\|_{p,q} = \left(\sum_{i \in [n]} (\sum_{j \in [n]} |x_{i,j}|^q)^{p/q}\right)^{1/p}$, for $p, q \geq 0$. Introduced in [15], this problem generalizes the ℓ_p-norm/F_k-moment estimation questions, and for various values of p and q, it has particular useful interpretations, see [15] for examples. Perhaps the first algorithm, applicable to some regime of parameters, appeared in [18]. Further progress on the problem was accomplished in [23], which obtains near-optimal bounds for a large range of values of $p, q \geq 0$ (see also [29] and [18]).
  We give in Section 4.2 algorithms for all parameters $p, q > 0$, and obtain bounds that are tight up to $(\epsilon^{-1} \log n)^{O(1)}$ factors. In particular, we obtain the first algorithm for the regime $q > p > 2$ — no such (efficient) algorithm was previously known. We show that the space complexity is controlled by a metric property, which is a generalization of the *p-type constant* of ℓ_q. Our space bounds fall out directly from bounds on this property.
- **$\ell_\mathbf{p}$-sampling, for $\mathbf{p} \in [\mathbf{1}, \mathbf{2}]$:** Here, the goal of the algorithm is to produce an index $i \in [n]$ sampled from a distribution D_x that depends on x, as opposed to producing a fixed function of x. In particular, the (idealized) goal is to produce an index $i \in [n]$ where each i is returned with probability $|x_i|^p / \|x\|_p^p$. We meet this goal in an approximate fashion: there exists some approximating distribution D'_x on $[n]$, where $D'_x(i) = (1 \pm \epsilon)|x_i|/\|x\|_1 \pm 1/n^2$ (the exponent 2 here is arbitrary), such that the algorithm outputs i drawn from the distribution D'_x. Note that the problem would be simple if the stream had only insertions (i.e., $\delta \geq 0$ always); so the challenge is to be able to support both positive and negative updates to the vector x.
  The ℓ_p-sampling problem was introduced in [29], where it is shown that the ℓ_p-sampling problem is a useful building block for other streaming problems, including cascaded norms, heavy hitters, and moment estimation. The algorithm in [29] uses $(\epsilon^{-1} \log n)^{O(1)}$ space.
  Our algorithm for the ℓ_p-sampling problem, for $p \in [1, 2]$, appears in the full paper. It improves the space to $O(\epsilon^{-p} \log^3 n)$. Very recently, following the framework introduced here, [24] further improve the space bound to a *near-optimal* bound, and extend the algorithm to $p \in [0, 1]$.

All our algorithms maintain a linear sketch $L : \mathbb{R}^n \to \mathbb{R}^S$ (i.e., L is a linear function), where S is the space bound (in words, or $O(S \log n)$ in bits). Hence, all the updates may be

[1] We make a standard discretization assumption that all numbers have a finite precision, and in particular, $\delta \in \{-M, -M+1, \ldots, M-1, M\}$, for $M = n^{O(1)}$.

[2] Since we work in the general update framework, we will not be presenting the literature that is concerned with restricted types of updates, such as positive updates $\delta > 0$.

[3] The difference in notation (p vs. k) is partly due to historical reasons: the ℓ_p norm for $p \in [1, 2]$ has been usually studied separately from the F_k moment for $k > 2$, having generally involved somewhat different techniques and space bounds.

implemented using the linearity: $L(x+\delta e_i) = Lx + \delta \cdot Le_i$, where e_i is the ith standard basis vector.

1.2. Precision Sampling

We now describe the key primitive used in all our algorithms, the Precision Sampling Lemma (PSL). It has originally appeared in [3]. The present version is improved in two respects: it has better bounds and is streaming-friendly.

PSL addresses a variant of the standard sum estimation problem, where the goal is to estimate the sum $\sigma \stackrel{\text{def}}{=} \sum_i a_i$ of n unknown quantities $a_i \in [0,1]$. In the standard sampling approach, one randomly samples a set of indices $I \subset [n]$, and uses these a_i's to compute an estimate such as $\frac{n}{|I|}\sum_{i\in I} a_i$. *Precision sampling* considers a different scenario, where the estimation algorithm chooses a sequence of *precisions* $u_i \in (0,1]$ (without knowing the a_i's), and then obtains a sequence of estimates $\hat{a}_i$ that satisfy $|\hat{a}_i - a_i| \le u_i$, and it has to report an estimate for the sum $\sigma = \sum_i a_i$. As it turns out from applications, producing an estimate with additive error u_i (for a single a_i) incurs cost $1/u_i$, hence the goal is to achieve a good approximation to σ while keeping tabs on the total cost (total precision) $\sum_i (1/u_i)$.[4]

To illustrate the concept, consider the case where $10 \le \sigma \le 20$, and one desires a 1.1 multiplicative approximation to σ. How should one choose the precisions u_i? One approach is to employ the aforementioned sampling approach: choose a random set of indices $I \subset [n]$ and assign to them a high precision, say $u_i = 1/n$, and assign trivial precision $u_i = 1$ to the rest of indices; then report the estimate $\hat{\sigma} = \frac{n}{|I|}\sum_{i\in I} \hat{a}_i$. This way, the error due to the adversary's response is at most $\frac{n}{|I|}\sum_{i\in I} |\hat{a}_i - a_i| \le 1$, and standard sampling (concentration) bounds prescribe setting $|I| = \Theta(n)$. The total precision becomes $\Theta(n \cdot |I|) = \Theta(n^2)$, which is no better than naively setting all precisions $u_i = 1/n$, which achieves total additive error 1 using total precision n^2. Note that in the restricted case where all $a_i \le 40/n$, the sampling approach is better, because setting $|I| = O(1)$ suffices; however, in another restricted case where all $a_i \in \{0,1\}$, the naive approach could fare better, if we set all $u_i = 1/2$. Thus, total precision $O(n)$ is possible in both cases, but by a different method. We previously proved in [3] that one can always choose w_i randomly such that $\sum w_i \le O(n \log n)$ with constant probability.

In this paper, we provide a more efficient version of PSL (see Section 2 for details). To state the lemma, we need a definition that accommodates both additive and multiplicative errors.

Definition 1.1 (Approximator)**.** *Let $\rho > 0$ and $f \in [1,2]$. A (ρ, f)-*approximator *to $\tau > 0$ is any quantity $\hat{\tau}$ satisfying $\tau/f - \rho \le \hat{\tau} \le f\tau + \rho$. (Without loss of generality, $\hat{\tau} \ge 0$.)*

[4]Naturally, in other application, other notions of cost may make more sense, and are worth investigating.

The following lemma is stated in a rather general form. Due to historical reasons, the lemma refers to precisions as $w_i \in [1,\infty)$, which is identical to our description above via $w_i = 1/u_i$. Upon first reading, it may be instructive to consider the special case $f = 1$, and let $\rho = \epsilon > 0$ be an absolute constant (say 0.1 to match our discussion above).

Lemma 1.2 (Precision Sampling Lemma)**.** *Fix an integer $n \ge 2$, a multiplicative error $\epsilon \in [1/n, 1/3]$, and an additive error $\rho \in [1/n, 1]$. Then there exist a distribution $\mathcal{W}$ on the real interval $[1,\infty)$ and a reconstruction algorithm R, with the following two properties.*

- **Accuracy:** *Consider arbitrary $a_1, \ldots, a_n \in [0,1]$ and $f \in [1, 1.5]$. Let $w_1, \ldots, w_n$ be chosen at random from $\mathcal{W}$ pairwise independently.*[5] *Then with probability at least $2/3$, when algorithm R is given $\{w_i\}_{i\in[n]}$ and $\{\hat{a}_i\}_{i\in[n]}$ such that each $\hat{a}_i$ is an arbitrary $(1/w_i, f)$-approximator of a_i, it produces $\hat{\sigma} \ge 0$ which is a $(\rho, f \cdot e^{\epsilon})$-approximator to $\sigma \stackrel{\text{def}}{=} \sum_{i=1}^n a_i$.*
- **Cost:** *There is $k = O(1/\rho\epsilon^2)$ such that the conditional expectation $\mathbb{E}_{w\in\mathcal{W}}[w \mid M] \le O(k \log n)$ for some event $M = M(w)$ occurring with high probability. For every fixed $\alpha \in (0,1)$, we have $\mathbb{E}_{w\in\mathcal{W}}[w^\alpha] \le O(k^\alpha)$. The distribution $\mathcal{W} = \mathcal{W}(k)$ depends only on k.*

We emphasize that the probability $2/3$ above is over the choice of $\{w_i\}_{i\in[n]}$ and holds (separately) for every fixed setting of $\{a_i\}_{i\in[n]}$. In the case where R is randomized, the probability $2/3$ is also over the coins of R. Note also that the precisions w_i are chosen without knowing a_i, but the estimators $\hat{a}_i$ are adversarial — each might depend on the entire $\{a_i\}_{i\in[n]}$ and $\{w_i\}_{i\in[n]}$, and their errors might be correlated.

In our implementation, it turns out that the reconstruction algorithm uses only $\hat{a}_i$'s which are (retrospectively) good approximation to a_i — namely $\hat{a}_i \gg 1/w_i$ — hence the adversarial effect is limited. For completeness, we also mention that, for $k = 1$, the distribution $\mathcal{W} = \mathcal{W}(1)$ is simply $1/u$ for a random $u \in [0,1]$. We present the complete proof of the lemma in Section 2.

It is natural to ask whether the above lemma is tight. In the full paper, we show a lower bound on $\mathbb{E}_{w\in\mathcal{W}}[w]$ in the considered setting, which matches our PSL bound up to a factor of $1/\epsilon$. We leave it as an open question what is the best achievable bound for PSL.

1.3. Connection to Priority *Sampling*

We remark that (our implementation of) Precision Sampling has some similarity to *Priority Sampling* [16], which is a scheme for the following problem.[6] We are given a vector $x \in \mathbb{R}^n_+$ of positive weights (coordinates), and we

[5]That is, for all $i < j$, the pair (w_i, w_j) is distributed as $\mathcal{W}^2$.

[6]The similarity is at the more technical level of applying the PSL in streaming algorithms, hence the foregoing discussion actually refers to Sections 2 and 3.

want to maintain a sample of k weights in order to be able to estimate sums of weights for an arbitrary subset of coordinates, i.e., $\sum_{i\in I} x_i$ for arbitrary sets $I \subseteq [n]$. Priority Sampling has been shown to attain an essentially best possible variance for a sampling scheme [32].

The similarity between the two sampling schemes is the following. In our main approach, similarly to the approach in Priority Sampling, we take the vector $x \in \mathbb{R}^n$, and consider a vector y where $y_i = x_i/u_i$, for u_i chosen at random from $[0,1]$. We are then interested in heavy hitters of the vector y (in ℓ_1 norm). We obtain these using the CountSketch/CountMin sketch [11], [14]. In Priority Sampling, one similarly extracts a set of k heaviest coordinates of y. However, one important difference is that in Priority Sampling the weights (and updates) are positive, thus making it possible to use Reservoir sampling-type techniques to obtain the desired heavy hitters. In contrast, in our setting the weights (and updates) may be negative, and we need to extract the heavy hitters approximately and hence post-process them differently.

See also [12] and the references therein for streaming-friendly versions of Priority Sampling and other related sampling procedures.

2. Proof of the Precision Sampling Lemma

In this section we prove the Precision Sampling Lemma (Lemma 1.2). Compared to our previous version of PSL from [3], this version has the following improvements: a better bound on $\mathbb{E}_{w\in\mathcal{W}}[w]$ (hence better total precision), it requires the w_i's to be only pairwise independent (hence streaming-friendly), and a slightly simpler construction and analysis via its inverse $u = 1/w$. In the full paper we show a lower bound for the total precision.

The probability distribution $\mathcal{W}$. Fix $k = \zeta/\rho\epsilon^2$ for sufficiently large constant $\zeta > 0$. The distribution $\mathcal{W}$ takes a random value $w \in [1,\infty)$ as follows: pick i.i.d. samples $u_1,\ldots,u_k$ from the uniform distribution $U(0,1)$, and set $w \stackrel{\text{def}}{=} \max_{j\in[k]} 1/u_j$. Note that $\mathcal{W}$ depends on k only.

The reconstruction algorithms. The randomized reconstruction algorithm R' gets as input $\{w_i\}_{i\in[n]}$ and $\{\hat{a}_i\}_{i\in[n]}$ and works as follows. For each $i \in [n]$, sample k i.i.d. random variables, $u_{i,j} \in U(0,1)$ for $j \in [k]$, conditioned on the event $\{w_i = \max_{j\in[k]} 1/u_{i,j}\}$. Now define the "indicators" $s_{i,j} \in \{0, 1/k\}$, for each $i \in [n], j \in [k]$, by setting

$$s_{i,j} \stackrel{\text{def}}{=} \begin{cases} 1/k & \text{if } u_{i,j} \le \hat{a}_i/t \text{ for } t \stackrel{\text{def}}{=} 4/\epsilon; \\ 0 & \text{otherwise.} \end{cases}$$

Finally, algorithm R' sets $s \stackrel{\text{def}}{=} \sum_{i\in[n],j\in[k]} s_{i,j}$ and reports $\hat{\sigma} \stackrel{\text{def}}{=} s\,t$ as an estimate for $\sigma = \sum_i a_i$. A key observation is that altogether, i.e., when we consider both the coins involved in the choice of w_i from $\mathcal{W}$ as well as those used by algorithm R', we can think of $u_{i,1},\ldots,u_{i,k}$ as being chosen i.i.d. from $U(0,1)$. Observe also that whenever $\hat{a}_i$ is a $(1/w_i, f)$-approximator to a_i, it is also a $(u_{i,j}, f)$-approximator to a_i for *all* $j \in [k]$.

We now build a more efficient deterministic algorithm R that performs at least as well as R'. Specifically, R does not generate the $u_{i,j}$'s (from the given w_i's), but rather sets $s_i \stackrel{\text{def}}{=} \mathbb{E}\left[\sum_{j\in[k]} s_{i,j} \mid \min_{j\in[k]} u_{i,j} = 1/w_i\right]$ and $s \stackrel{\text{def}}{=} \sum_{i\in[n]} s_i$. A simple calculation yields an explicit formula, which is easy to compute algorithmically:

$$s_i = \begin{cases} \frac{1}{k} + \frac{k-1}{k}\cdot\frac{\hat{a}_i w_i/t - 1}{w_i - 1}; & \text{if } \hat{a}_i w_i/t \ge 1 \\ 0 & \text{otherwise.} \end{cases}$$

We proceed to the analysis of this construction. We will first consider the randomized algorithm R', and then show that derandomization can only decrease the error.

Proof of Lemma 1.2: We first give bounds on the moments of the distribution $\mathcal{W}$. Indeed, recall that by definition $w \stackrel{\text{def}}{=} \max_{j\in[k]} \frac{1}{u_j}$. We define the event M to be that $w \le n^5$; note that $\Pr[M] \ge 1 - k\cdot n^{-5} \ge 1 - O(n^{-2})$. Conditioned on M, each $u_j \in U(n^{-5}, 1)$, and we have $\mathbb{E}\left[\frac{1}{u_j}\right] = \frac{1}{1-n^{-5}}\int_{n^{-5}}^1 \frac{1}{x}\,\mathrm{dx} = \frac{\ln(n^5)}{1-n^{-5}}$. Thus

$$\mathbb{E}_{w\in\mathcal{W}}[w \mid M] \le \mathbb{E}\left[\sum_{j\in[k]} \frac{1}{u_j} \mid M\right] \le O(k\log n).$$

Now fix $\alpha \in (0,1)$. It is immediate that $\mathbb{E}[1/u^\alpha] = O(1/(1-\alpha))$. We can similarly prove that $\mathbb{E}_{w\in\mathcal{W}}[w^\alpha] \le O(k^\alpha/(1-\alpha))$, but the calculation is technical, and we include its proof in Appendix A.

We now need to prove that $\hat{\sigma}$ is an approximator to σ, with probability at least $2/3$. The plan is to first compute the expectation of $s_{i,j}$, for each $i \in [n], j \in [k]$. This expectation depends on the approximator values $\hat{a}_i$, which itself may depend (adversarially) on w_i, so instead we give upper and lower bounds on the expectation $\mathbb{E}[s_{i,j}] \approx \frac{a_i}{tk}$. Then, we wish to apply a concentration bound on the sum of $s_{i,j}$, but again the $s_{i,j}$ might depend on the random values w_i, so we actually apply the concentration bound on the upper/lower bounds of $s_{i,j}$, and thereby derive bounds on $s = \sum s_{i,j}$.

Formally, we define random variables $\overline{s}_{i,j}, \underline{s}_{i,j} \in \{0, 1/k\}$. We set $\overline{s}_{i,j} = 1/k$ iff $u_{i,j} \le f a_i/(t-1)$, and 0 otherwise. Similarly, we set $\underline{s}_{i,j} = 1/k$ iff $u_{i,j} \le a_i/f(t+1)$, and 0 otherwise. We now claim that

$$\underline{s}_{i,j} \le s_{i,j} \le \overline{s}_{i,j}. \tag{1}$$

Indeed, if $s_{i,j} = 1/k$ then $u_{i,j} \le \hat{a}_i/t$, and hence, using the fact that $\hat{a}_i$ is a $(u_{i,j}, f)$-approximator to a_i, we have $u_{i,j} \le f a_i/(t-1)$, or $\overline{s}_{i,j} = 1/k$. Similarly, if $s_{i,j} = 0$, then $u_{i,j} > \hat{a}_i/t$, and hence $u_{i,j} > a_i/f(t+1)$, or $\underline{s}_{i,j} = 0$. Notice for later use that each of $\{\overline{s}_{i,j}\}$ and $\{\underline{s}_{i,j}\}$ is a collection of nk pairwise independent random variables. For ease of notation, define $\underline{\hat{\sigma}} = t\sum_{i,j}\underline{s}_{i,j}$ and $\overline{\hat{\sigma}} = t\sum_{i,j}\overline{s}_{i,j}$, and observe that $\underline{\hat{\sigma}} \le \hat{\sigma} \le \overline{\hat{\sigma}}$.

We now bound $\mathbb{E}[\overline{s}_{i,j}]$ and $\mathbb{E}\left[\underline{s}_{i,j}\right]$. For this, it suffices to compute the probability that $\overline{s}_{i,j}$ and $\underline{s}_{i,j}$ are $1/k$. For the first quantity, we have:

$$\Pr\left[\overline{s}_{i,j} = \tfrac{1}{k}\right] = \Pr\left[u_{i,j} \le \tfrac{fa_i}{t-1}\right] = \tfrac{fa_i}{t-1} \le e^{\epsilon/2} f \cdot \tfrac{a_i}{t}, \quad (2)$$

where we used the fact that $t - 1 \ge e^{-\epsilon/2}t$. Similarly, for the second quantity, we have:

$$\Pr\left[\underline{s}_{i,j} = \tfrac{1}{k}\right] = \Pr\left[u_{i,j} \le \tfrac{a_i}{f(t+1)}\right] = \tfrac{a_i}{f(t+1)} \ge e^{-\epsilon/2} f^{-1} \cdot \tfrac{a_i}{t}. \quad (3)$$

Finally, using Eqn. (1) and the fact that $\mathbb{E}[s] = \sum_{i,j} \mathbb{E}[s_{i,j}]$, we can bound the expectation and variance of $\hat{\sigma} = st$ as follows:

$$e^{-\epsilon/2} f^{-1} \cdot \sigma \le t \sum_{i,j} \mathbb{E}\left[\underline{s}_{i,j}\right] \le \mathbb{E}[\hat{\sigma}] \le t \sum_{i,j} \mathbb{E}[\overline{s}_{i,j}] \le e^{\epsilon/2} f \cdot \sigma, \quad (4)$$

and, using pairwise independence, $\mathbf{Var}[\underline{\hat{\sigma}}], \mathbf{Var}\left[\overline{\hat{\sigma}}\right] \le t^2 \cdot \sum_{i,j} k^{-2} \cdot e^{\epsilon/2} \cdot \frac{fa_i}{t} \le 4tk^{-1}\sigma$. Recall that we want to bound the probability that $\underline{\hat{\sigma}}$ and $\overline{\hat{\sigma}}$ deviate (additively) from their expectation by roughly $\epsilon\sigma + \rho$, which is larger than their standard deviation $O(\sqrt{tk^{-1}\sigma}) = O(\sqrt{\rho\epsilon\sigma})$.

Formally, to bound the quantity $\hat{\sigma}$ itself, we distinguish two cases. First, consider $\sigma > \rho/\epsilon$. Then for our parameters $k = \zeta/\rho\epsilon^2$ and $t = 4/\epsilon$,

$$\Pr\left[\overline{\hat{\sigma}} > e^{\epsilon/2} f\sigma \cdot (1 + \epsilon/2)\right] \le \Pr\left[\overline{\hat{\sigma}} - \mathbb{E}\left[\overline{\hat{\sigma}}\right] > \epsilon/2 \cdot e^{\epsilon} f\sigma\right]$$

$$\le \tfrac{\mathbf{Var}\left[\overline{\hat{\sigma}}\right]}{(\epsilon/2 \cdot e^{\epsilon} f\sigma)^2} \le \tfrac{4tk^{-1}\sigma}{\epsilon^2\sigma^2/4} \le \tfrac{O(\rho/\epsilon\zeta)}{\sigma} \le 0.1$$

for sufficiently large ζ. Similarly, $\Pr[\underline{\hat{\sigma}} < f^{-1}e^{-\epsilon/2}\sigma \cdot e^{-\epsilon/2}] \le 0.1$.

Now consider the second case, when $\sigma \le \rho/\epsilon$. It holds

$$\Pr\left[\overline{\hat{\sigma}} > fe^{\epsilon/2}\sigma + \rho\right] \le \Pr\left[\overline{\hat{\sigma}} - \mathbb{E}\left[\overline{\hat{\sigma}}\right] > \rho\right]$$

$$\le \tfrac{\mathbf{Var}\left[\overline{\hat{\sigma}}\right]}{\rho^2} \le \tfrac{4tk^{-1} \cdot \rho/\epsilon}{\rho^2} \le 0.1.$$

Similarly, we have $\Pr[\underline{\hat{\sigma}} < f^{-1}e^{-\epsilon/2}\sigma - \rho] \le 0.1$. This completes the proof that $\hat{\sigma}$ is a (ρ, fe^{ϵ})-approximator to σ, with probability at least $2/3$.

Finally, we argue that switching to the deterministic algorithm R only decreases the variances without affecting the expectations, and hence the same concentration bounds hold. Formally, denote our replacement for s_i by $s'_i = \mathbb{E}_{u_{i,j}}\left[\sum_{j \in [k]} s_{i,j} \mid \max_{j \in [k]} 1/u_{i,j} = w_i\right]$, and note it is a random variable (because of w_i). Define $\overline{s}'_i = \mathbb{E}\left[\sum_{j \in [k]} \overline{s}_{i,j} \mid \max_{j \in [k]} 1/u_{i,j} = w_i\right]$, and by applying conditional expectation to Eqn. (1), we have $s_i \le \overline{s}'_i$. We now wish to bound the variance of $\sum_i \overline{s}'_i$. By the law of total variance, and using the shorthand $\vec{w} = \{w_i\}_i$,

$$\mathbf{Var}\left[\textstyle\sum_i \overline{s}_i\right] = \mathbb{E}\left[\mathbf{Var}\left[\textstyle\sum_i \overline{s}_i \mid \vec{w}\right]\right] + \mathbf{Var}\left[\mathbb{E}\left[\textstyle\sum_i \overline{s}_i \mid \vec{w}\right]\right]. \quad (5)$$

We now do a similar calculation for $\sum_i \overline{s}'_i$, but since each $\overline{s}'_i$ is completely determined from the known $\vec{w}$, the first summand is just 0 and in the second summand we can change each $\overline{s}'_i$ to $\overline{s}_i$, formally

$$\mathbf{Var}\left[\textstyle\sum_i \overline{s}'_i\right] = \mathbb{E}\left[\mathbf{Var}\left[\textstyle\sum_i \overline{s}'_i \mid \vec{w}\right]\right] + \mathbf{Var}\left[\mathbb{E}\left[\textstyle\sum_i \overline{s}'_i \mid \vec{w}\right]\right]$$

$$= \mathbf{Var}\left[\mathbb{E}\left[\textstyle\sum_i \overline{s}_i \mid \vec{w}\right]\right]. \quad (6)$$

Eqns. (5) and (6) imply that in the deterministic algorithm the variance (of the upper bound) can indeed only decrease. The analysis for the lower bound is analogous, using $\underline{s}'_i$. As before, using the fact that the $\overline{s}'_i$ are pairwise independent (because the w_i are) we apply Chebyshev's inequality to bound deviation for the algorithm $R's$ actual estimate $\hat{\sigma} = t\sum_i s'_i$. ∎

3. Applications I: Warm-Up

We now describe our streaming algorithms that use the Precision Sampling Lemma (PSL) as the core primitive. We first outline two generic procedures that are used by several of our applications. The current description leaves some parameters unspecified: they will be fixed by the particular applications. These two procedures are also given in pseudocode as Alg. 1 and Alg. 2.

As previously mentioned, our sketch function is a linear function $L : \mathbb{R}^n \to \mathbb{R}^S$ mapping an input vector $x \in \mathbb{R}^n$ into $\mathbb{R}^S$, where S is the space (in words). The algorithm is simply a fusion of PSL with a heavy hitters algorithm [11], [14]. We use a parameter $p \ge 1$, which one should think of as the p in the ℓ_p-norm estimation problem, and $p = k$ in the F_k moment estimation. Other parameters are: $\rho \in (0, 1)$ (additive error), $\epsilon \in (0, 1/3)$ (multiplicative error), and $m \in \mathbb{N}$ (a factor in the space usage).

The sketching algorithm is as follows. We start by initializing a vector of w_i's using Lemma 1.2: specifically we draw w_i's from $\mathcal{W} = \mathcal{W}(k)$ for $k = \frac{\zeta}{\rho\epsilon^2}$. We use $l = O(\log n)$ hash tables $\{H_j\}_{j \in [l]}$, each of size m. For each hash table H_j, choose a random hash function $h_j : [n] \to [m]$, and Rademacher random variables $g_j : [n] \to \{-1, +1\}$. Then the sketch Lx is obtained by repeating the following for every hash table $j \in [l]$ and index $i \in [n]$: hash index $i \in [n]$ to find its cell $h_j(i)$, and add to this cell's contents the quantity $g_j(i) \cdot x_i w_i^{1/p}$. Overall, $S = lm$.

The estimation algorithm E proceeds as follows. First normalize the sketch Lx by scaling it down by an input parameter $r \in \mathbb{R}_+$. Now for each $i \in [n]$, compute the median, over the l hash tables, of the pth power of cells where i falls into. Namely, let $\hat{x}_i$ be the median of $|H_j(h_j(i))|^p / r w_i$ over all $j \in [l]$. Then run the PSL reconstruction algorithm R on the vectors $\{\hat{x}_i\}_{i \in [n]}$ and $\{w_i\}_{i \in [n]}$, to obtain an estimate $\hat{\sigma} = \hat{\sigma}(r)$. The final output is $r \cdot \hat{\sigma}(r)$.

We note that it will always suffice to use pairwise independence for each set of random variables $\{w_i\}_i$, $\{g_j(i)\}_i$,

and $\{h_j(i)\}_i$ for each $j \in [l]$. For instance, it suffices to draw each hash function h_j from a universal hash family.

Finally, we remark that, while the reconstruction Alg. 2 takes time $\Omega(n)$, one can reduce this to time $m \cdot (\epsilon^{-1} \log n)^{O(1)}$ by using a more refined heavy hitter sketch. We discuss this issue later in this section.

Algorithm 1: Sketching algorithm for norm estimation. Input is a vector $x \in \mathbb{R}^n$. Parameters p, ϵ, ρ, and m are specified later.

1 Generate $\{w_i\}_{i\in[n]}$ as prescribed by PSL, using $\mathcal{W} = \mathcal{W}(k)$ for $k = \zeta\rho^{-1}\epsilon^{-2}$.
2 Initialize $l = O(\log n)$ hash tables $H_1, \ldots, H_l$, each of size m. For each table H_j, choose a random hash function $h_j : [n] \to [m]$ and a random $g_j : [n] \to \{-1, +1\}$.
3 **for** *each* $j \in [l]$ **do**
4 Multiply x coordinate-wise with the vectors $\{w_i^{1/p}\}_{i\in[n]}$ and g_j, and hash the resulting vector into the hash table H_j. Formally, $H_j(z) \triangleq \sum_{i:h_j(i)=z} g_j(i) \cdot w_i^{1/p} \cdot x_i$.

Algorithm 2: Reconstruction algorithm for norm estimation. Input consists of l hash tables H_j, precisions w_i for $i \in [n]$, and a real $r > 0$. Other parameters, p, ϵ, ρ, m, are as in Alg. 1.

1 For each $i \in [n]$, compute $\hat{x}_i = \text{median}_{j\in[l]} \left\{ \frac{|H_j(h_j(i))/r|^p}{w_i} \right\}$.
2 Apply PSL reconstruction algorithm R to vector $(\hat{x}_1, \ldots \hat{x}_n)$ and $(w_1, \ldots w_n)$, and let $\hat{\sigma}$ be its output. Explicitly, for each $i \in [n]$, if $\hat{x}_i w_i \geq t \triangleq 4/\epsilon$, then set $s_i \triangleq \frac{1}{k} + \frac{k-1}{k} \cdot \frac{\hat{x}_i w_i/t - 1}{w_i - 1}$ (recall $k = \zeta\rho^{-1}\epsilon^{-2}$ from PSL), otherwise $s_i \triangleq 0$; then, let $\hat{\sigma} = t \sum_i s_i$.
3 Output $r \cdot \hat{\sigma}$.

3.1. Estimating F_k Moments for $k > 2$

We now present the algorithm for estimating F_k moments for $k > 2$, using the PSL Lemma 1.2. To reduce the clash of parameters, we refer to the problem as "F_p moment estimation".

Theorem 3.1. *Fix $n \geq 8$, $p > 2$, and $0 < \epsilon < 1/3$. There is a randomized linear function $L : \mathbb{R}^n \to \mathbb{R}^S$, with $S = O(n^{1-2/p} \cdot p^2 \epsilon^{-2-4/p} \log n)$, and a deterministic estimation algorithm $E : \mathbb{R}^S \to \mathbb{R}$, such that for every $x \in \mathbb{R}^n$, with probability at least 0.51, its output $E(L(x))$ approximates $\|x\|_p^p$ within factor $1 + \epsilon$.*

Proof of Theorem 3.1: Our linear sketch L is Alg. 1, and the estimation algorithm E is Alg. 2, with the following choice of parameters. Let $\rho = \frac{\epsilon/4}{n^{p/2-1}}$. Let $\mathcal{W} = \mathcal{W}(k)$, for $k = \zeta\rho^{-1}\epsilon^{-2}$, be from PSL Lemma 1.2. Define $\omega = 9\mathbb{E}_{w\in\mathcal{W}}\left[w^{2/p}\right]$, and note that $\omega \leq O(\rho^{-2/p}\epsilon^{-4/p})$ by Lemma 1.2. Finally we set $m = \alpha \cdot O(\rho^{-2/p}\epsilon^{-4/p})$ so that $m \geq \alpha\omega$, where $\alpha = \alpha(p, \epsilon) > 1$ will be determined later.

In Alg. 2, we set r to be a factor $1 - 1/p$ approximation to $\|x\|_2$, i.e., $(1-1/p)\|x\|_2 \leq r \leq \|x\|_2$. Note that such r is easy to compute (with high probability) using, say, the AMS linear sketch [1], with $O(p^2 \log n)$ additional space. Thus, for the rest, we will just assume that $\|x\|_2 \in [1 - 1/p, 1]$ and set $r = 1$.

The plan is to apply PSL Lemma 1.2 where each unknown value a_i is given by $|x_i|^p$, and each estimate $\hat{a}_i$ is given by $\hat{x}_i$. For this purpose, we need to prove that the $\hat{x}_i$'s are good approximators. We thus let $F_2 = \sum_{i=1}^n (x_i w_i^{1/p})^2$. Note that $\mathbb{E}[F_2] = \|x\|_2^2 \cdot \mathbb{E}_{w\in\mathcal{W}}\left[w^{2/p}\right] \leq \omega/9$, and hence by Markov's inequality, with probability at least $8/9$ we have $F_2 \leq \omega$.

Claim 3.2. *Assume that $F_2 \leq \omega$. Then with high probability (say $\geq 1 - 1/n^2$) over the choice of the hash tables, for every $i \in [n]$ the value $\hat{x}_i$ is a $(1/w_i, e^\epsilon)$-approximator to $|x_i|^p$.*

Proof: We shall prove that for each $i \in [n]$ and $j \in [l]$, with probability $\geq 8/9$ over the choice of h_j and g_j, the value $\frac{|H_j(h_j(i))|^p}{w_i}$ is a $(1/w_i, e^\epsilon)$-approximator to $|x_i|^p$. Recall that each $\hat{x}_i$ is the median of $|H_j(h_j(i))|^p/w_i$ over $l = O(\log n)$ values of j, we get by applying a Chernoff bound that with high probability it is a $(1/w_i, e^\epsilon)$-approximator to $|x_i|^p$. The claim then follows by a union bound over all $i \in [n]$.

Fix $i \in [n]$ and $j \in [l]$, let $Y \triangleq H_j(h_j(i))$. For $f \in [n]$, define $y_f = g_j(f) \cdot x_f w_f^{1/p}$ if $h_j(f) = h_j(i)$ and 0 otherwise. Then $Y = y_i + \delta$ where $\delta \triangleq \sum_{f\neq i} y_f$. Ideally, we would like that $|Y|^p \approx |y_i|^p = |x_i|^p w_i$, i.e., the effect of the error δ is small. Indeed, $\mathbb{E}\left[\delta^2\right] = \mathbb{E}\left[(\sum_{f\neq i} y_f)^2\right] = \frac{1}{m}\sum_{f\neq i}(x_f w_f^{1/p})^2 \leq F_2/m$. Hence, by Markov's inequality, $|\delta| \leq \sqrt{9F_2/m} \leq 3/\sqrt{\alpha}$ with probability at least $8/9$.

We now argue that if this event $|\delta| \leq 3/\sqrt{\alpha}$ occurs, then $\frac{|H_j(h_j(i))|^p}{w_i} = \frac{|Y|^p}{w_i} = \left|g_j(i)x_i + \delta/w_i^{1/p}\right|^p$ is a good approximator to $|x_i|^p$. Indeed, if $|\delta|/w_i^{1/p} \leq \frac{\epsilon}{2p}|x_i|$, then clearly $\frac{|Y|^p}{w_i} = (1 \pm \frac{\epsilon}{2p})^p |x_i|^p$. Otherwise, since $|\delta| \leq 3/\sqrt{\alpha}$, we have that

$$\begin{aligned}
\left| |Y|^p - |x_i w_i^{1/p}|^p \right| &\leq (|x_i w_i^{1/p}| + |\delta|)^p - |x_i w_i^{1/p}|^p \\
&\leq (\tfrac{2p}{\epsilon}|\delta| + |\delta|)^p - (\tfrac{2p}{\epsilon}|\delta|)^p \\
&\leq |\delta|^p \cdot (2p/\epsilon)^p \cdot \left((1 + \tfrac{\epsilon}{2p})^p - 1\right) \\
&\leq (6p)^p \cdot \epsilon^{1-p}/\alpha^{p/2}.
\end{aligned}$$

If we set $\alpha = (6p)^2/\epsilon^{2-2/p}$, then we obtain that $\frac{|Y|^p}{w_i}$ is a $(1/w_i, e^\epsilon)$-approximator to $|x_i|^p$, with probability at least

$8/9$. We now take median over $O(\log n)$ hash tables and apply a union bound to reach the desired conclusion. ■

We can now complete the proof of Theorem 3.1. Apply PSL (Lemma 1.2) with $a_i = |x_i|^p$ and $\hat{a}_i = \hat{x}_i$'s. By Hölder's inequality for $p/2$ and the normalization $r = 1$, we have $\|x\|_p^p \geq \|x\|_2^p / n^{p/2-1} \geq \rho/\epsilon$, and thus additive error ρ transforms to multiplicative error $1+\epsilon$. It remains to bound the space: $S \leq O(m \log n) = O(\alpha \rho^{-2/p} \epsilon^{-4/p} \log n) = O(p^2/\epsilon^{2-2/p} \cdot \epsilon^{-6/p} n^{1-2/p} \cdot \log n) = O(p^2 n^{1-2/p} \cdot \epsilon^{-2-4/p} \cdot \log n)$. ■

3.2. Estimating ℓ_1 Norm

To further illustrate the use of the Alg. 1 and 2, we now show how to use them for estimating the ℓ_1 norm. In a later section, we obtain similar results for all ℓ_p, $p \in [1, 2]$, except that the analysis is more involved.

We obtain the following theorem. For clarity of presentation, the efficiency (space and runtime bounds) are discussed separately below.

Theorem 3.3. *Fix $n \geq 8$ and $8/n < \epsilon < 1/8$. There is a randomized linear function $L : \mathbb{R}^n \to \mathbb{R}^S$, with $S = O(\epsilon^{-3} \log^2 n)$, and a deterministic estimation algorithm $E : \mathbb{R}^S \to \mathbb{R}$, such that for every $x \in \mathbb{R}^n$, with probability at least 0.51, its output $E(L(x))$ approximates $\|x\|_1$ within factor $1 + \epsilon$.*

Proof: The sketch function L is given by Alg. 1, with parameters $p = 1$, $\rho = \epsilon/8$, and $m = C\epsilon^{-3} \log n$ for a constant $C > 0$ defined shortly. Let $\mathcal{W} = \mathcal{W}(k)$ for $k = \zeta \rho^{-1} \epsilon^{-2}$ be obtained from the PSL Lemma 1.2. Define $\omega = 10 \mathbb{E}_{w \in \mathcal{W}} [w \mid M]$, where event $M = M(w)$ satisfies $\Pr[M] \geq 1 - O(n^{-2})$. Note that $\omega \leq O(\epsilon^{-3} \log n)$. We set constant C such that $m \geq 3\omega$.

The estimation procedure is just several invocations of Alg. 2 for different values of r. For the time being, assume we hold an overestimate of $\|x\|_1$, which we call $r \geq \|x\|_1$. Then algorithm E works by applying Alg. 2 with this parameter r.

Let $F_1 = \sum_{i=1}^n |x_i w_i|/r$. Note that $\mathbb{E}[F_1 \mid \cap_i M(w_i)] = \|x\|_1/r \cdot \mathbb{E}_{w \in \mathcal{W}} [w \mid M(w)] \leq \omega/10$, and hence by Markov's inequality, $F_1 \leq \omega \leq m/3$ with probability at least $9/10 - O(n/n^2) \geq 8/9$. Call this event $\mathcal{E}_r$, and assume henceforth it indeed occurs.

To apply the PSL, we need to prove that each $\hat{x}_i$ in Alg. 2 is a good approximator to x_i. Fix $i \in [n]$ and $j \in [l]$. We claim that, conditioned on $\mathcal{E}_r$, the with probability at least $2/3$, $\frac{|H_j(h_j(i))|}{r w_i}$ is a $(1/w_i, 1)$-approximator of $|x_i|$. Indeed, $\frac{H_j(h_j(i))}{r w_i} = \frac{1}{r} g_j(i) x_i + \frac{1}{r w_i} \sum_{f \neq i : h_j(f) = h_j(i)} g_j(f) w_f x_f$, and thus,

$$\mathbb{E}\left[\left|\frac{|H_j(h_j(x))|}{r w_i} - \frac{|x_i|}{r}\right|\right] \leq \frac{1}{r w_i} \sum_{f \neq i} \frac{1}{m} |x_f w_f| \leq \frac{F_1}{m w_i} \leq \frac{1}{3 w_i}.$$

Hence, by Markov's inequality, $\frac{|H_j(h_j(x))|}{r w_i}$ is a $(1/w_i, 1)$-approximator of $|x_i|/r$ with probability at least $2/3$. By a Chernoff bound, their median $\hat{x}_i = \mathrm{median}_{j \in [l]} \left\{ \frac{|H_j(h_j(i))|}{r w_i} \right\}$ is a $(1/w_i, 1)$-approximator to $|x_i|/r$ with probability at least $1 - n^{-2}$. Taking a union bound over all $i \in [n]$ and applying the PSL (Lemma 1.2), we obtain that the PSL output, $\hat{\sigma} = \hat{\sigma}(r)$ is an $(\epsilon/8, e^\epsilon)$-approximator to $\|x\|_1/r$, with probability at least $2/3 - 1/9 - 1/n^2 \geq 0.6$.

Now, if we had $r \leq 4\|x\|_1$, then we would be done as $r\hat{\sigma}$ would be a $(\epsilon \|x\|_1/2, e^\epsilon)$-approximator to $\|x\|_1$, and hence a $1+2\epsilon$ multiplicative approximator (and this easily transforms to factor $1+\epsilon$ by suitable scaling of ϵ). Without such a good estimate r, we try all possible values r that are powers of 2, from high to low, until we make the right guess. Notice that it is easy to verify that the current guess r is sufficiently large that we can safely decrease it. Specifically, if $r > 4\|x\|_1$ then $r\hat{\sigma} < e^\epsilon \|x\|_1 + \epsilon r/8 \leq (r/4) \cdot [1 + 3\epsilon/2 + \epsilon/2] = (1 + 2\epsilon) r/4$. However, if $r \leq 2\|x\|_1$ then $r\hat{\sigma} \geq e^{-\epsilon} \|x\|_1 - \epsilon r/8 \geq (r/2) \cdot [1 - \epsilon - \epsilon/4] > (1 + 2\epsilon) r/4$. We also remark that, while we repeat Alg. 2 for $O(\log n)$ times (starting from $r = n^{O(1)}$ suffices), there is no need to increase the probability of success as the relevant events $\mathcal{E}_r = \{\sum_i |x_i w_i| \leq rm/3\}$ are nested and contain the last one, where $r/\|x\|_1 \in [1, 4]$. ■

3.3. The Running Times

We now briefly discuss the runtimes of our algorithms: the update time of the sketching Alg. 1, and the reconstruction time of the Alg. 2.

It is immediate to note that the update time of our sketching algorithm is $O(\log n)$: one just has to update $O(\log n)$ hash tables. We also note that we can compute a particular w_i in $O(\log n)$ time, which is certainly doable as w_i may be generated directly from the seed used for the pairwise-independent distribution. Furthermore, we note that we can sample from the distribution $\mathcal{W} = \mathcal{W}(k)$ in $O(1)$ time (see, e.g., [22]).

Now we turn to the reconstruction time of Alg. 2. As currently described, this runtime is $O(n \log n)$. One can improve the runtime by using the CountMin heavy hitters (HH) sketch of [14], at the cost of a $O(\log(\frac{\log n}{\epsilon}))$ factor increase in the space and update time. This improvement is best illustrated in the case of ℓ_1 estimation. We construct the new sketch by just applying the $\Theta(t/m)$-HH sketch (Theorem 5 of [14]) to the vector $x \cdot w$ (entry-wise product). The HH procedure returns at most $O(m/t)$ coordinates i, together with $(1/w_i, e^\epsilon)$-approximators $\hat{x}_i$, for which it is possible that $\hat{x}_i w_i \geq t$ (note that, if the HH procedure does not return some index i, we can consider 0 as being its approximator). This is enough to run the estimation procedure E from PSL, which uses only i's for which $\hat{x}_i w_i \geq t$. Using the bounds from [14], we obtain the following guarantees. The total space is $O(\epsilon^{-1} \log n \log(\frac{\log n}{\epsilon}) \cdot m/t) = O(m \log n \cdot \log(\frac{\log n}{\epsilon})) = O(\epsilon^{-3} \log^2 n \cdot \log(\frac{\log n}{\epsilon}))$. The

update time is $O(\log n \cdot \log(\frac{\log n}{\epsilon}))$ and reconstruction time is $O(\log^2 n \cdot \log(\frac{\log n}{\epsilon}))$.

To obtain a similar improvement in reconstruction time for the F_k-moment problem, one uses an analogous approach, except that one has to use HH with respect to the ℓ_2 norm, instead of the ℓ_1 norm (considered in [14]).

4. Applications II: Bounds via p-Type Constant

In this section, we show further applications of the PSL to streaming algorithms. As in Section 3, our sketching algorithm will be linear, following the lines of the generic Alg. 1.

An important ingredient for our intended applications will be a variation of the notion of *p-type* of a Banach space (or, more specifically, the p-type constant). This notion will give a bound on the space usage of our algorithms, and hence we will bound it in various settings. Below we state the simplest such bound, which is a form of the Khintchine inequality.

Lemma 4.1. *Fix $p \in [1,2]$, $n \geq 1$ and $x \in \mathbb{R}^n$. Suppose that for each $i \in [n]$ we have two random variables, $g_i \in \{-1,+1\}$ chosen uniformly at random, and $\chi_i \in \{0,1\}$ chosen to be 1 with probability $\alpha \in (0,1)$ (and 0 otherwise). Then*

$$\mathbb{E}\left[\left|\sum_i g_i \chi_i x_i\right|^p\right] \leq \alpha \|x\|_p^p.$$

Furthermore, suppose each family of random variables $\{g_i\}_i$ and $\{\chi_i\}_i$ is only pairwise independent and the two families are independent of each other. Then, with probability at least 7/9, we have that

$$\left|\sum_i g_i \chi_i x_i\right|^p \leq 3^{2+p} \alpha \|x\|_p^p.$$

The proof of this lemma appears in the full paper.

4.1. ℓ_p-norm for $p \in [1,2]$

We now use Alg. 1 and 2 to estimate the ℓ_p norm for $p \in [1,2]$. We use Lemma 4.1 to bound the space usage.

Theorem 4.2. *Fix $p \in [1,2]$, $n \geq 6$, and $0 < \epsilon < 1/8$. There is a randomized linear function $L : \mathbb{R}^n \to \mathbb{R}^S$, with $S = O(\epsilon^{-2-p} \log^2 n)$, and a deterministic estimation algorithm E, such that for every $x \in \mathbb{R}^n$, with probability at least 0.51, $E(L(x))$ is a factor $1+\epsilon$ approximation to $\|x\|_p^p$.*

Proof: Our sketch function L is given by Alg. 1. We set $\rho = \epsilon/8$. Let $\mathcal{W} = \mathcal{W}(k)$ for $k = \zeta \rho^{-1} \epsilon^{-2}$ obtained from the PSL (Lemma 1.2). Define $\omega = 10 \mathbb{E}_{w \in \mathcal{W}}[w \mid M]$, where event $M = M(w)$ satisfies $\Pr[M] \geq 1 - O(n^{-2})$. Note that $\omega \leq O(\epsilon^{-3} \log n)$. We set $m = \alpha\omega$ for a constant $\alpha > 0$ to be determined later.

We now describe the exact reconstruction procedure, which will be just several invocations of the algorithm 2 for different values of r. As in Theorem 3.3, we guess $r > 0$ starting from the highest possible value and halving it each time, until we obtain a good estimate: $\|x\|_p \leq r \leq 4\|x\|_p$ (alternatively, one could prepare for all possible r's). To simplify the exposition, let us just assume in the sequel that $r = 1$ and thus $1/4 \leq \|x\|_p \leq 1$.

Let $F_p = \sum_{i=1}^n |x_i|^p w_i$. Note that $\mathbb{E}[F_p \mid \cap_i M(w_i)] = \|x\|_p^p \cdot \mathbb{E}_{w \in \mathcal{W}}[w \mid M(w)] \leq \omega/10$, and hence by Markov's inequality, $F_p \leq \omega$ with probability at least $8/9$. Call this event $\mathcal{E}$ and assume henceforth it occurs. To apply PSL, we need to prove that every $\hat{x}_i$ from Alg. 2 is a good approximator to x_i.

Claim 4.3. *Assume $F_p \leq \omega$ and fix $i \in [n]$. If $\alpha \geq 3^{2+p}\epsilon^{1-p}$, then with high probability, $\hat{x}_i$ is a $(1/w_i, e^\epsilon)$-approximator to $|x_i|^p$.*

Proof: Fix $j \in [l]$; we shall prove that $|H_j(h_j(i))|^p$ is a $(1, 1+\epsilon)$-approximator to $|x_i|^p w_i$, with probability at least $2/3$. Then we would be done by Chernoff bound, as $\hat{x}_i$ is a median over $l = O(\log n)$ independent trials $j \in [l]$.

For $f \in [n]$, define $y_f = g_j(f) \cdot x_i w_i^{1/p}$ if $h_j(f) = h_j(i)$ and $y_f = 0$ otherwise. Define $Y \triangleq H_j(h_j(i)) = y_i + \delta$, where $\delta = \sum_{f \neq i} y_f$. We apply Lemma 4.1 to conclude that $\mathbb{E}[|\delta|^p] \leq F_p/m$, and hence $|\delta|^p \leq 3\omega/m \leq 3/\alpha$ with probability at least $2/3$. Assume henceforth this is indeed the case.

Now we distinguish two cases. First, suppose $|x_i w_i^{1/p}| \geq \frac{2}{\epsilon} \cdot |\delta|$. Then $|Y|^p = (1 \pm \epsilon/2)|x_i|^p w_i$. Otherwise, $|x_i w_i^{1/p}| < \frac{2}{\epsilon} \cdot |\delta|$, and then

$$\begin{aligned}
\left||Y|^p - |x_i w_i^{1/p}|^p\right| &\leq (|x_i w_i^{1/p}| + |\delta|)^p - |x_i w_i^{1/p}|^p \\
&\leq |\delta|^p \cdot ((2/\epsilon + 1)^p - 2/\epsilon) \\
&\leq |\delta|^p \cdot (2/\epsilon)^p \cdot (1 + p\epsilon - 1) \\
&\leq p 2^p \cdot 3 \cdot \epsilon^{1-p}/\alpha.
\end{aligned}$$

Thus, if we set $\alpha \geq 3^{2+p}(1/\epsilon)^{p-1}$, then in both cases $|Y|^p$ is a $(1, e^\epsilon)$-approximator to $|x_i|^p w_i$ (under the event that occurs with probability at least $2/3$). ∎

We can now complete the proof of Theorem 4.2. Applying Lemma 1.2, we obtain that its output, $\hat{\sigma} = \hat{\sigma}(r)$, is a $(\epsilon/8, e^{2\epsilon})$-approximator to $\|x\|_p$, with probability at least $2/3 - 1/9 - 1/n^2 \geq 0.51$. ∎

4.2. Mixed and cascaded norms

We now show how to estimate mixed norms such as the $\ell_{p,q}$ norms. In the latter case, the input is a matrix $x \in \mathbb{R}^{n_1 \cdot n_2}$, and the $\ell_{p,q}$ norm is $\|x\|_{p,q} = (\sum_i \|x_i\|_q^p)^{1/p}$, where x_i is the ith row in the matrix.

We show a more general theorem, for the norm $\ell_p(X)$, which is defined similarly for a general Banach space X; the $\ell_{p,q}$ norms will be just particular cases. To state the general result, we need the following definition.

Definition 4.4. *Fix $p \geq 1$, $n, \kappa \in \mathbb{N}$, $\omega > 0$, $\delta \in [0,1)$, and let X be a finite dimensional Banach space. The* generalized p-type, *denoted $\alpha(X, p, n, \kappa, \omega, \delta)$, is the biggest*

constant $\alpha > 0$ satisfying the following: For each $i \in [n]$, let $g_i \in \{-1, +1\}$ be a random variable drawn uniformly at random, and let $\chi_i \in \{0, 1\}$ be a random variable that is equal 1 with probability $1/\alpha$ and 0 otherwise. Furthermore, each family $\{g_i\}_i$ and $\{\chi\}_i$ is κ-wise independent, and the two families are independent of each other. Then, for every $x_1, \dots x_n \in X$ satisfying $\sum_{i\in[n]} \|x_i\|_X^p \le \omega$,

$$\Pr\left[\left\| \sum_{i\in[n]} g_i \chi_i x_i \right\|_X^p \le 1\right] \ge 1 - \delta.$$

Theorem 4.5. *Fix $p \ge 1$, $n \ge 2$, and $0 < \epsilon < 1/3$. Let X be a Banach space admitting a linear sketch $L_X : X \to \mathbb{R}^{S_X}$, with space $S_X = S_X(\epsilon)$, and let $E_X : \mathbb{R}^{S_X} \to \mathbb{R}$ be its reconstruction procedure.*

Then there is a randomized linear function $L : X^n \to \mathbb{R}^S$, and an estimation algorithm E which, for any $x \in X^n$, given the sketch Lx, outputs a factor $1+\epsilon$ approximation to $\|x\|_{p,X}$, with probability at least 0.51.

Furthermore, $S \le S_X(\epsilon/2) \cdot \alpha(X, p, n, \kappa, O(p\epsilon^{-4}\log n), 2/3) \cdot O(\log n)$, where κ is such that each function g_j and h_j is κ-wise independent.

We note that the result for $\ell_{p,q}$ norms will follow by proving some particular bounds on the parameter α, the generalized p-type. We discuss these implications after the proof of the theorem.

Proof of Theorem 4.5: Our sketch function L is given by algorithm 1, with one notable modification. x_i's are now vectors from X and the hash table cells hold sketches given by sketching function L_X up to $1+\epsilon/2$ approximation. In particular, each cell of hash table $H_j(z) = \sum_{i:h_j(i)=z} g_j(i) \cdot w_i^{1/p} \cdot L_X x_i$. Furthermore, abusing notation, we use the notation $\|H_j(z)\|_q$ for some $z \in [m]$ to mean the result of the E-estimation algorithm on the sketch $H_j(z)$ (since it is a $1+\epsilon/2$ approximation, we can afford such additional multiplicative error).

We set $\rho = \epsilon/8$. Let $\mathcal{W} = \mathcal{W}(k)$ by for $k = \zeta\rho^{-1}\epsilon^{-2}$ obtained from the PSL Lemma 1.2. Define $\omega = 10\mathbb{E}_{w\in\mathcal{W}}[w \mid M]$, where event $M = M(w)$ satisfies $\Pr[M] \ge 1 - O(n^{-2})$. Note that $\omega \le O(\epsilon^{-3}\log n)$. We set m later.

We now describe the exact reconstruction procedure, which will be just several invocations of the algorithm 2 for different values of r. As in Theorem 3.3, we guess r starting from high and halving it each time, until we obtain a good estimate — $\|x\|_{p,X} \le r \le 4\|x\|_{p,X}$ (alternatively, one could prepare for all possible r's). For simplified exposition, we just assume that $1/4 \le \|x\|_{p,X} \le 1$ and $r = 1$ in the rest.

Let $F_{p,X} = \sum_{i=1}^n \|x_i w_i^{1/p}\|_X^p$. Note that $\mathbb{E}[F_{p,X} \mid \cap M(w_i)] = \|x\|_X^p \cdot \mathbb{E}_{w\in\mathcal{W}}[w \mid M(w)] \le \omega/10$, and hence $F_{p,X} \le \omega$ with probability at least $8/9$ by Markov's bound. Call this event $\mathcal{E}$. To apply PSL, we need to prove that $\hat{x}_i$'s from Alg. 2 are faithful approximators. For this, we prove that, for appropriate choice of $\alpha = \alpha(p, X, \epsilon, n)$, for each $j \in [l]$, $\|H_j(h_j(i))\|_X^p$ is a $(1, 1+\epsilon)$-approximator to $\|x_i\|_X^p w_i$, with probability at least $2/3$. This would imply that, since $\hat{x}_i$ is a median over $O(\log n)$ independent trials, $\hat{x}_i$ is a $(1/w_i, 1+\epsilon)$-approximator to $\|x_i\|_X^p$. Once we have such a claim, we apply Lemma 1.2, and conclude that the output, $\hat{\sigma} = \hat{\sigma}(r)$, is a $(\epsilon/8, 1+2\epsilon)$-approximator to $\|x\|_{p,X}$, with probability at least $2/3 - 1/9 - 1/n \ge 0.51$.

Claim 4.6. *Fix $p \ge 1$ and $\omega \in \mathbb{R}_+$. Let $m = \alpha(X, p, \kappa, 3p\omega/\epsilon, 2/3)$, the generalized p-type of X.*

Assume $F_{p,X} \le \omega$ and fix $i \in [n], j \in [l]$. Then $\|H_j(h_j(i))\|_X^p$ is a $(1, 1+\epsilon)$-approximator to $\|x_i\|_X^p w_i$ with probability at least 2/3.

Proof: For $f \in [n]$, define $y_f = g_j(f) \cdot x_i w^{1/p}$ if $h_j(f) = h_j(i)$ and $y_f = 0$ otherwise. Then, $a \triangleq \sum_{f\in[n]:h_j(f)=h_j(i)} g_j(i)x_i = y_i + \delta$, where $\delta = \sum_{f\neq i} y_f$. Then, by the definition of generalized p-type of X, whenever $m \ge \alpha(X, p, \kappa, \omega \cdot \frac{3p}{\epsilon}, 2/3)$, we have that $\|\delta\|_X \le \epsilon/3$, with probability at least $2/3$.

Now we distinguish two cases. First, suppose $\|x_i w_i^{1/p}\|_X \ge \frac{2p}{\epsilon} \cdot \|\delta\|_X$. Then $\|a\|_X^p \approx (1 \pm \epsilon)\|x_i\|_X^p w_i$. Otherwise, if $\|x_i w_i^{1/p}\|_X < \frac{2p}{\epsilon} \cdot \|\delta\|_X$, then

$$\|a\|_X^p \le \left(\|x_i w_i^{1/p}\|_X + \|\delta\|_X\right)^p \le (2p\|\delta\|_X/\epsilon + \|\delta\|_X)^p$$

$$\le \|\delta\|_X^p \cdot (2p/\epsilon + 1)^p \le 1.$$

Hence, we conclude that $\|a\|_X^p$ (and thus $\|H_j(h_j(i))\|_X^p$) is a $(1, 1+\epsilon)$-approximator to $\|x_i\|_X^p w_i$, with probability at least $2/3$. ∎

The claim concludes the proof of Theorem 4.5. Note that the space is $S = O(S_X(\epsilon/2) \cdot \alpha(X, p, \kappa, O(p\epsilon^{-4}\log n), 2/3) \cdot \log n)$. ∎

We now show the implications of the above theorem. For this, we present the following lemma, whose proof appears in the full paper.

Lemma 4.7. *Fix $n, m \in \mathbb{N}$, $\omega \in \mathbb{R}_+$, and a finite dimensional Banach space X. We have the following bounds on the generalized p-type:*

(a)) *if $0 < p \le q \le 2$, then $\alpha(\ell_q^m, p, n, 2, \omega, 2/3) \le O(\omega)$.*
(b)) *if $p, q \ge 2$, we have that $\alpha(\ell_q^m, p, n, 2q, \omega, 2/3) \le 9^2 q^{O(1)} \omega^{2/p} \cdot n^{1-2/p}$, and if $q \ge 2$ and $p \in (0, 2)$, then $\alpha(\ell_q^m, p, n, 2q, \omega, 2/3) \le 9^2 q^{O(1)} \omega^{2/p}$.*
(c)) *for $p \ge 1$, we have that $\alpha(X, p, n, 2, \omega, 2/3) \le O(n^{1-1/p}\omega^{1/p})$, and for $p \in (0, 1)$, we have that $\alpha(X, p, n, 2, \omega, 2/3) \le O(\omega^{1/p})$.*

Combining Theorem 4.5 and Lemma 4.7, also using Theorem 3.1, we obtain the following linear sketches for $\ell_{p,q}$ norms, which are optimal up to $(\epsilon^{-1}\log n)^{O(1)}$ factors (see, e.g., [23]).

Corollary 4.8. *There exist linear sketches for $\ell_p^{n_1}(\ell_q^{n_2})$, for $n_1, n_2 \leq n$ and $p, q \geq 1$, with the following space bounds S.*

For $0 < p \leq q \leq 2$, the bound is $S = (\epsilon^{-1} \log n)^{O(1)}$.
If $q \geq 2$, $p \in (0, 2)$, then $S = n_2^{1-2/q} \cdot (pq\epsilon^{-1} \log n)^{O(1)}$.
If $p, q \geq 2$, then $S = n_1^{1-2/p} n_2^{1-2/q} \cdot (pq\epsilon^{-1} \log n)^{O(1)}$.
If $p \geq 1$, $q \in (0, p)$, then $S = n_1^{1-1/p} \cdot (\epsilon^{-1} \log n)^{O(1)}$.
If $p \in (0, 1)$, $q \in (0, p)$, then $S = (\epsilon^{-1} \log n)^{O(1)}$.

ACKNOWLEDGMENTS

We would like to thank Piotr Indyk, Assaf Naor, and David Woodruff for helpful discussions about the F_k-moment estimation problem. We also thank Andrew McGregor for kindly giving an overview of the landscape of the heavy hitters problem.

REFERENCES

[1] N. Alon, Y. Matias, and M. Szegedy, "The space complexity of approximating the frequency moments," *J. Comp. Sys. Sci.*, vol. 58, pp. 137–147, 1999, previously appeared in STOC'96.

[2] A. Andoni, K. Do Ba, P. Indyk, and D. Woodruff, "Efficient sketches for Earth-Mover Distance, with applications," in *Proc. of FOCS*, 2009.

[3] A. Andoni, R. Krauthgamer, and K. Onak, "Polylogarithmic approximation for edit distance and the asymmetric query complexity," in *Proc. of FOCS*, 2010, a full version is available at `http://arxiv.org/abs/1005.4033`.

[4] Z. Bar-Yossef, T. S. Jayram, R. Kumar, and D. Sivakumar, "An information statistics approach to data stream and communication complexity," *J. Comput. Syst. Sci.*, vol. 68, no. 4, pp. 702–732, 2004.

[5] L. Bhuvanagiri and S. Ganguly, "Estimating entropy over data streams," in *Proc. of ESA*, 2006, pp. 148–159.

[6] L. Bhuvanagiri, S. Ganguly, D. Kesh, and C. Saha, "Simpler algorithm for estimating frequency moments of data streams," in *Proc. of SODA*, 2006, pp. 708–713.

[7] V. Braverman and R. Ostrovsky, "Measuring independence of datasets," in *Proc. of STOC*, 2010.

[8] ——, "Recursive sketching for frequency moments," *CoRR*, vol. abs/1011.2571, 2010.

[9] ——, "Zero-one frequency laws," in *Proc. of STOC*, 2010.

[10] A. Chakrabarti, S. Khot, and X. Sun, "Near-optimal lower bounds on the multi-party communication complexity of set disjointness," in *IEEE Conference on Computational Complexity*, 2003, pp. 107–117.

[11] M. Charikar, K. Chen, and M. Farach-Colton, "Finding frequent items in data streams," in *Proc. of ICALP*, 2002.

[12] E. Cohen, N. G. Duffield, H. Kaplan, C. Lund, and M. Thorup, "Stream sampling for variance-optimal estimation of subset sums," in *Proc. of SODA*, 2009, pp. 1255–1264.

[13] G. Cormode and M. Muthukrishnan, "Count-min sketch," 2010, `https://sites.google.com/site/countminsketch`.

[14] G. Cormode and S. Muthukrishnan, "An improved data stream summary: the count-min sketch and its applications," *J. Algorithms*, vol. 55, no. 1, pp. 58–75, 2005.

[15] ——, "Space efficient mining of multigraph streams," in *Proc. of PODS*, 2005.

[16] N. G. Duffield, C. Lund, and M. Thorup, "Priority sampling for estimation of arbitrary subset sums," *J. ACM*, vol. 54, no. 6, 2007.

[17] S. Ganguly, "Personal communication," April 2011.

[18] S. Ganguly, M. Bansal, and S. Dube, "Estimating hybrid frequency moments of data streams," in *Frontiers in Algorithmics*, 2008.

[19] S. Ganguly and G. Cormode, "On estimating frequency moments of data streams," in *Proc. of RANDOM*, 2007.

[20] P. Indyk, "Stable distributions, pseudorandom generators, embeddings and data stream computation," *J. ACM*, vol. 53, no. 3, pp. 307–323, 2006, previously appeared in FOCS'00.

[21] P. Indyk and D. Woodruff, "Optimal approximations of the frequency moments of data streams," *Proc. of STOC*, 2005.

[22] S. Ioffe, "Improved consistent sampling, weighted minhash and L1 sketching," in *International Conference on Data Mining*, 2010.

[23] T. Jayram and D. Woodruff, "The data stream space complexity of cascaded norms," in *Proc. of FOCS*, 2009.

[24] H. Jowhari, M. Saglam, and G. Tardos, "Tight bounds for L_p samplers, finding duplicates in streams, and related problems," *CoRR*, vol. abs/1012.4889, 2010.

[25] D. M. Kane, J. Nelson, E. Porat, and D. P. Woodruff, "Fast moment estimation in data streams in optimal space," in *Proc. of STOC*, 2011.

[26] D. M. Kane, J. Nelson, and D. P. Woodruff, "On the exact space complexity of sketching small norms," in *Proc. of SODA*, 2010.

[27] P. Li, "Estimators and tail bounds for dimension reduction in l_p $(0 < p \leq 2)$ using stable random projections," in *Proc. of SODA*, 2008.

[28] J. Misra and D. Gries, "Finding repeated elements," *Sci. Comput. Program.*, vol. 2, no. 2, pp. 143–152, 1982.

[29] M. Monemizadeh and D. Woodruff, "1-pass relative-error l_p-sampling with applications," in *Proc. of SODA*, 2010.

[30] M. Muthukrishnan, *Data Streams: Algorithms and Aplications*, ser. Foundations and Trends in Theoretical Computer Science. Now Publishers Inc, Jan. 2005.

[31] J. Nelson and D. Woodruff, "Fast Manhattan sketches in data streams," in *Proc. of PODS*, 2010.

[32] M. Szegedy, "The DLT priority sampling is essentially optimal," in *Proc. of STOC*, 2006, pp. 150–158.

APPENDIX

Claim A.1. *For $k \geq 1$, suppose u_j are drawn uniformly at random from $[0, 1]$. Then, for any $\alpha \in (0, 1)$, we have that $\mathbb{E}_{u_j}\left[(\max_j 1/u_j)^\alpha\right] \leq O\left(\frac{k^\alpha}{1-\alpha}\right)$.*

Proof: We compute the expectation directly:

$$\mathbb{E}_{u_j}\left[(\max_j 1/u_j)^\alpha\right] = \int_0^1 u^{-\alpha} \cdot k(1-u)^{k-1}\,\mathrm{du}$$

$$\leq \int_0^{1/k} k \cdot u^{-\alpha}\,\mathrm{du} + \int_{1/k}^1 k^\alpha \cdot k(1-u)^{k-1}\,\mathrm{du}$$

$$= k \cdot \left[\frac{u^{1-\alpha}}{1-\alpha}\right]_0^{1/k} + k^\alpha \Big[-(1-u)^k\Big]_{1/k}^1 \leq O(\tfrac{k^\alpha}{1-\alpha}).$$

■