# Sketching Probabilistic Data Streams

Graham Cormode
AT&T Labs—Research
180 Park Avenue, Florham Park NJ
graham@research.att.com

Minos Garofalakis*
Yahoo! Research and UC Berkeley
2821 Mission College Blvd, Santa Clara CA
minos@yahoo-inc.com

## ABSTRACT

The management of uncertain, probabilistic data has recently emerged as a useful paradigm for dealing with the inherent unreliabilities of several real-world application domains, including data cleaning, information integration, and pervasive, multi-sensor computing. Unlike conventional data sets, a set of probabilistic tuples defines a probability distribution over an exponential number of *possible worlds* (i.e., "grounded", deterministic databases). This "possible worlds" interpretation allows for clean query semantics but also raises hard computational problems for probabilistic database query processors. To further complicate matters, in many scenarios (e.g., large-scale process and environmental monitoring using multiple sensor modalities), probabilistic data tuples arrive and need to be processed in a streaming fashion; that is, using limited memory and CPU resources and without the benefit of multiple passes over a static probabilistic database. Such probabilistic data streams raise a host of new research challenges for stream-processing engines that, to date, remain largely unaddressed.

In this paper, we propose the first space- and time-efficient algorithms for approximating complex aggregate queries (including, the number of distinct values and join/self-join sizes) over probabilistic data streams. Following the possible-worlds semantics, such aggregates essentially define probability distributions over the space of possible aggregation results, and our goal is to characterize such distributions through efficient approximations of their key moments (such as expectation and variance). Our algorithms offer strong randomized estimation guarantees while using only sublinear space in the size of the stream(s), and rely on novel, concise streaming sketch synopses that extend conventional sketching ideas to the probabilistic streams setting. Our experimental results verify the effectiveness of our approach.

**Categories and Subject Descriptors:**
E.1 [**Data**]: Data Structures; F.2 [**Theory**]: Analysis of Algorithms

**General Terms:** Algorithms, Performance, Reliability

**Keywords:** Data Streams, Uncertain Data.

*Work done while at Intel Research, Berkeley


*SIGMOD'07,* June 11–14, 2007, Beijing, China.


## 1. INTRODUCTION

Conventional database systems and query processing tools are designed around the idea of *static* collections of *exact* data tuples. Unfortunately, the data generated by a diverse set of real-world applications is often *uncertain and imprecise*. For instance, data integration and record linkage tools can generate distinct degrees of confidence for output data tuples (based on the quality of the match for the underlying entities) [8]; structured information extractors typically assign different confidences to rules for identifying meaningful patterns in their (unstructured) input [17]; and, pervasive multi-sensor computing applications need to routinely handle noisy sensor/RFID readings [19]. Motivated by these new application requirements, recent research efforts on *probabilistic data management* aim to incorporate uncertainty and probabilistic information as "first-class citizens" of the database system.

Among the different approaches for managing uncertainty inside the database, *tuple-level uncertainty models* (that, essentially, associate independent existence probabilities with individual tuples) have seen wide adoption both in research papers as well as system prototypes. This is due to both their simplicity of representation in current relational systems (i.e., just adding an "existence probability" column), as well as their simple and intuitive query semantics. In a nutshell, a probabilistic database is a concise representation for a probability distribution over an exponentially-large collection of *possible worlds*, each representing a possible "grounded" (deterministic) instance of the database (e.g., by flipping appropriately-biased independent coins to select each uncertain tuple). This "possible-worlds" semantics also implies clean semantics for queries over a probabilistic database — essentially, the result of a probabilistic query defines a probability distribution over the space of possible query results across all possible worlds [8].

Unfortunately, despite its simple, intuitive semantics, the paradigm shift towards tuple-level uncertainty also appears to imply a huge jump in the computational complexity of simple query processing operations: As demonstrated by Dalvi and Suciu [8], correctly evaluating the resulting tuple probabilities for duplicate eliminating projections over a simple three-way join can give rise to problems with $\#P$-hard data complexity (i.e., exponential in the number of database tuples). Query-processing efficiency issues are, of course, further exacerbated by the *streaming nature* of several target applications for probabilistic data-management tools. For example, large-scale process and environmental monitoring tools rely on the continuous collection and processing of large amounts of noisy, uncertain readings from numerous sensor modalities, and recent work has already suggested attaching tuple-level probabilities as explicit indicators of data quality [19]. Due to the continuous, online nature of these applications and the large data volumes involved, these probabilistic data tuples arrive continuously and need to be pro-

cessed in a streaming fashion; that is, query results must be available in real-time, using only limited memory and CPU resources, and without the benefit of several passes over a static probabilistic database. Efficient stream-processing algorithms and system architectures for deterministic tuple streams have formed a very active research area in recent years. As in the static database case, however, such *probabilistic data streams* raise a host of new, difficult research challenges for stream-processing engines that mandate novel algorithmic approaches.

**Prior Work.** Several efficient algorithms have been developed for processing different classes of complex queries over massive data streams; examples include computing quantiles [15], estimating distinct-value counts [4], counting frequent elements (i.e., "heavy hitters") [7, 20], approximating large Haar-wavelet coefficients [14], and estimating join sizes and stream norms [1, 2, 9]. None of these works consider the issues raised by uncertain streaming tuples.

Recent work on probabilistic database systems has focused on different aspects of managing uncertain data tuples in relational DBMS architectures, including the complexity and algorithmic problems of query evaluation [8], data modeling issues [23], and managing lineage for probabilistic query results [5]. The issue of efficient aggregate query evaluation under the stringent constraints of the streaming model was not considered in any of these papers. Khoussainova *et al.* [19] propose an architecture for cleaning sensor readings by attaching explicit "correctness" probabilities — their techniques are essentially complementary to our work. More recently, Jayram *et al.* [16] have studied the problem of evaluating simple aggregate functions (focusing, in particular, on AVERAGE) over streams of uncertain data. Their model of uncertainty is actually richer than ours, and can capture *value-level uncertainty* as well as the probability of existence. Still, even for very simple aggregates like AVERAGE, their techniques and analyses rely on sophisticated mathematical tools (e.g., generating functions), and it is very unclear if they can be extended to the broader, more complex class of aggregate queries considered here. Independently and concurrent with this work, Jayram *et al.* [18] showed some results for the expectation versions of $F_0$, $F_2$ and quantiles problems studied here, and improve their results for AVERAGE.

**Our Contributions.** In this paper, we initiate the study of space- and time-efficient techniques for approximating a broad class of complex aggregate queries over continuous probabilistic tuple streams. By possible-worlds semantics, such *probabilistic aggregates* define probability distributions over an (exponentially-large) space of possible results; thus, our goal is to characterize such distributions through efficient, streaming approximations of their key *moments* (such as expectation and variance) over the possible-worlds collection. We propose novel, randomized sketching synopses and estimation algorithms for probabilistic data streams; as with conventional data streaming, our methods employ only sublinear space (in both the size and domain of the probabilistic stream) and offer strong randomized estimation guarantees for the key moments of the underlying complex aggregate. More concretely, our contributions are summarized as follows.

• **Generic Streaming Probabilistic Aggregate Estimator based on Possible-Worlds Sampling.** We present a universal aggregate estimation algorithm for probabilistic data streams based on the intuitive idea of *sampling* possible-worlds (i.e., deterministic streams) from the input, and running conventional streaming estimators over the sampled streams. The obvious appeal of such a scheme is that it allows us to directly leverage existing space-efficient algorithms for deterministic data streams in the probabilistic setting. Unfortunately, as our analysis and experimental results show, this approach has severe limitations when it comes to estimating the moments of complex aggregates over streaming probabilistic data.

• **Probabilistic FM (pFM) Sketch Synopses and Estimators for Probabilistic Count-Distinct Aggregates.** While count-based aggregates (e.g., expected tuple counts or heavy-hitters) allow for simple solutions (due to their *linearity*), that is *not* the case for more complex aggregate queries over the probabilistic stream. We initially focus on the class of *count-distinct* queries and introduce a novel, randomized estimation algorithm for probabilistic data streams. Our algorithm relies on a new, hash-based *sketch synopsis* structure for streaming probabilistic data (termed *probabilistic FM (pFM) sketch*), inspired by the well-known Flajolet-Martin (FM) sketch [11] for counting distinct deterministic values. We introduce and analyze pFM-sketch-based estimators for probabilistic count-distinct, and demonstrate strong error guarantees while using only small space to sketch a large probabilistic data stream.

• **Streaming Estimators for Probabilistic Self-Join/Join Sizes and Higher Frequency Moments based on AMS Techniques.** For the second moment of a probabilistic data stream, corresponding to the expected self-join size, and the related quantity of the join size of two independent probabilistic streams, we develop new insights based on expressing the expectation and variance in terms of the *cumulants* of appropriate distributions. These cumulants can be computed easily for each component of the stream, and we show how to represent them all compactly, and manipulate them, using variations of the Alon-Matias-Szegedy (AMS) sketch data structure [2]. We show the accuracy and power of this approach, as we are able to track yet higher moments using cumulant analysis.

• **Experimental Results Validating our Approach.** We perform a thorough evaluation of our techniques on a mixture of real and synthetic data. We observe that our methods can be highly practical: they typically obtain a small ($< 10\%$) error on streams of millions of items, using only tens of kilobytes and taking only seconds of CPU time. We show that simple techniques often suffice for estimating expectations, but to understand the higher moments our more involved algorithms are a necessity.

Due to space constraints, several proofs and detailed technical arguments are omitted; complete details are deferred to the full version.

## 2. PROBABILISTIC DATA STREAM MODEL

We consider a simple model of probabilistic data streams, where each stream renders a *multi-set of relational tuples* with *independent tuple-level uncertainties* (i.e., existence probabilities). As mentioned earlier, such independent probabilistic tuples form the basis of several recent studies on probabilistic data management (e.g., [8]), since they can be naturally represented in relational systems, and allow for clean and intuitive query semantics; furthermore, even such simple models of uncertainty raise intractable computational problems for probabilistic query processing [8].

More formally, we define a probabilistic data stream as a sequence of $N$ uncertain tuples $\langle t, p\rangle$, with the semantics that tuple $t$ occurs in an instance of the database with probability $p \in (0, 1]$ (independently of all other tuples in the database).[1] Tuples $t$ are drawn from a finite domain of size $M$ that, without loss of generality, is assumed to be the integer domain $[M] = \{1, \ldots, M\}$. As is typical in data stream analysis, we focus on the case when $N$ and $M$ are "large", and our stream query processor can observe the streaming tuples *only once* (in the fixed order of arrival) and can

[1]For simplicity, we assume that all arithmetic is exact, i.e., we do not address issues of precision, and instead assume that all probabilities can be represented exactly within a small, constant number of machine words.

only use space/time that is *sublinear in both $N$ and $M$* to maintain a concise *synopsis* of the probabilistic data stream. As our results show, relatively simple techniques can be used in the case where $O(\min\{N, M\})$ space is available, but more advanced tools are needed under sublinear space constraints. This model is a special case of the more general model where each tuple encodes a compact probability distribution (PDF); we comment that many of our results immediately apply to this more general setting, but for simplicity we concentrate our discussion on the simpler case.

**Aggregate Estimation over Probabilistic Streams.** Following all earlier work on probabilistic databases, we view a probabilistic data stream as defining a probability distribution over a collection of *possible worlds*: Implicitly, a probabilistic stream encodes exponentially many (up to $2^N$) conventional, deterministic data streams, each occurring with some probability (determined by the individual existence probabilities of its constituent tuples). We refer to these possible deterministic instantiations of a probabilistic data stream $\mathcal{S}$ as its *grounded streams* (denoted by $\mathrm{grnd}(\mathcal{S})$). Specifically, consider a probabilistic stream $\mathcal{S} = (\langle t_i, p_i\rangle : 1 \leq i \leq N)$ and let $G(I)$ denote one possible outcome of $\mathcal{S}$ comprising the sequence of all tuples with index in some subset $I \subseteq [N]$ (i.e., $G(I) = (t_i : i \in I)$); then, by tuple independence, it is easy to see that the probability of this possible outcome $G(I)$ can be computed simply as $\Pr[G(I)] = \prod_{i\in I} p_i \cdot \prod_{j\notin I}(1 - p_j)$. Note, of course, that, since the tuples $t_i$ are not necessarily distinct (i.e., a stream renders a bag of tuples), several distinct index subsets $I$ can in fact map to the *same* grounded stream $G \in \mathrm{grnd}(\mathcal{S})$; thus, $\mathrm{grnd}(\mathcal{S}) \leq 2^N$ and the probability of a grounded stream instance $G \in \mathrm{grnd}(\mathcal{S})$ is defined as $\Pr[G] = \sum_{I:G(I)=G} \Pr[G(I)]$. (It is straightforward to show that $\Pr[G]$ defines a valid probability distribution over $\mathrm{grnd}(\mathcal{S})$.)

EXAMPLE 2.1. *Consider the simple probabilistic stream $\mathcal{S} = (\langle x, \frac{1}{2}\rangle, \langle y, \frac{1}{4}\rangle, \langle y, \frac{1}{3}\rangle)$. This encodes $2^3 = 8$ possible outcomes, covering 6 distinct grounded stream instances:*

$$\mathrm{grnd}(\mathcal{S}) = \{(x); (y); (x, y); (y, y); (x, y, y); \phi\}$$

*(where $\phi$ denotes the empty stream). The probabilities for each possible distinct grounded stream can be easily computed as:*

| *Grounded Stream G* | *(x)* | *(y)* | *(x,y)* | *(y,y)* | *(x,y,y)* | $\phi$ |
|---|---|---|---|---|---|---|
| $\Pr[G]$ | $\frac{1}{4}$ | $\frac{5}{24}$ | $\frac{5}{24}$ | $\frac{1}{24}$ | $\frac{1}{24}$ | $\frac{1}{4}$ |

Our focus here is on computing complex aggregates over probabilistic data streams. As mentioned earlier, such aggregates essentially define a *probability distribution* over the (exponentially large) space of aggregation results for all possible worlds. Given the potentially enormous size and complexity of such distributions, our goal is instead to effectively characterize these probabilistic data aggregates through efficient streaming approximations of their key *distribution moments*, such as *expectation* and *variance* over the possible-worlds collection. More formally, consider a probabilistic data stream $\mathcal{S}$, and let $F(\mathcal{S})$ denote the result of evaluating an aggregate function $F()$ over $\mathcal{S}$. Then, $F(\mathcal{S})$ is a *random variable* ranging over all possible grounded streams $G \in \mathrm{grnd}(\mathcal{S})$ with expectation and variance naturally defined as

$$\begin{aligned}
\mathsf{E}_G[F(\mathcal{S})] &= \sum_{G\in\mathrm{grnd}(\mathcal{S})} \Pr[G] \cdot F(G) \qquad \text{and} \\
\mathsf{Var}_G[F(\mathcal{S})] &= \mathsf{E}_G[F^2(\mathcal{S})] - \mathsf{E}_G^2[F(\mathcal{S})] \\
&= \sum_{G\in\mathrm{grnd}(\mathcal{S})} \Pr[G] \cdot F^2(G) \; - \; \mathsf{E}_G^2[F(\mathcal{S})],
\end{aligned}$$

where we use the "$G$" subscript to denote the underlying probability space of all possible worlds $G \in \mathrm{grnd}(\mathcal{S})$.

Naturally, a "naive" way to compute $\mathsf{E}_G[F(\mathcal{S})]$ and $\mathsf{Var}_G[F(\mathcal{S})]$ in $\Omega(2^N)$ time and space, is by explicitly grounding $\mathcal{S}$ and computing $\Pr[G]$ and the (deterministic) aggregate value $F(G)$ for every $G \in \mathrm{grnd}(\mathcal{S})$. However, our goal is *much* stronger: we aim to perform these computations in time exponentially less than this (essentially, *in a single pass* over the probabilistic tuples), and with space significantly sublinear (e.g., poly-logarithmic) in $N$ and $M$. These are the typical requirements for efficient query processing algorithms in the (deterministic) streaming model [3, 13].

The specific class of aggregate queries of interest in this work are the *frequency moments* of a stream (as well as closely related streaming problems). Such frequency moments have formed the basis of most algorithmic studies of (non-probabilistic) data streams, and the techniques developed for their computation are at the heart of many other algorithms for data-stream query processing [1, 2, 3, 9, 7, 13]. Formally, consider a grounded stream $G$ and let $f_t$ denote the number of occurrences of element $t \in [M]$ in $G$. (We use "tuple" and "element" interchangeably in the remainder of the paper.) The $k^{th}$ *frequency moment* of $G$, $F_k(G)$ ($k \geq 0$), is defined as $F_k(G) = \sum_{t=1}^{M} f_t^k$, where $k \geq 0$. We treat $0^0 = 0$, which implies that $F_0(G)$ is the number of domain elements $t \in [M]$ such that $f_t$ is non-zero, i.e., the *number of distinct tuples* in the stream $G$. Similarly, $F_1(G)$ is simply the total number of tuples in $G$ (i.e., $F_1(G) = N$), and $F_2(G)$ is the size of the *self-join* of $G$ (over the attributes of the streaming tuples) [2].

The $k^{th}$ frequency moment $F_k(\mathcal{S})$ of a probabilistic stream $\mathcal{S}$ can now be defined naturally as a random variable over $G \in \mathrm{grnd}(\mathcal{S})$, as earlier. Of course, it is important to note here the distinction between the $F_k(\mathcal{S})$ frequency moments, and the corresponding *distribution moments*, such as $\mathsf{E}_G[F_k(\mathcal{S})]$ and $\mathsf{Var}_G[F_k(\mathcal{S})]$ over $\mathrm{grnd}(\mathcal{S})$ — our goal in this work is to devise streaming algorithms for estimating such distribution moments in a single pass over $\mathcal{S}$. As with most work on data-stream algorithmics, our approach is based on the design of efficient *randomized schemes* that, ideally, guarantee $(\epsilon, \delta)$-*approximation bounds* [2, 11]; that is, given a quantity $X$ to be estimated over a stream, we aim to produce an estimate $\hat{X}$ of $X$ such that $\Pr[|\hat{X} - X| < \epsilon X] > 1 - \delta$.

# 3. WARM-UP: BASIC STREAM ESTIMATES

In this section, we start by describing a general aggregate estimation scheme for probabilistic data streams based on the idea of *sampling possible worlds*. Then, we briefly discuss a simple streaming estimator for the first frequency moment of a probabilistic stream $\mathcal{S}$ (i.e., $F_1(\mathcal{S})$), and its implications for other probabilistic aggregate queries (including, quantiles and heavy-hitters).

## 3.1 Universal Sampling-based Algorithm

Based on our possible-worlds interpretation of probabilistic stream aggregates, a natural *sampling-based scheme* emerges for obtaining streaming, estimators for $\mathsf{E}_G[F(\mathcal{S})]$ and $\mathsf{Var}_G[F(\mathcal{S})]$, for *any* aggregate $F()$ that can be computed (or, accurately approximated) over a *deterministic* data stream.

The idea is to randomly sample a small subset of possible worlds for an input probabilistic stream $\mathcal{S}$, where a grounded stream $G \in \mathrm{grnd}(\mathcal{S})$ is chosen with probability $\Pr[G]$. This is simple to implement: Given a target sample size $s$, initialize $G_j = \phi$ ($j = 1, \ldots, s$), and, for each incoming probabilistic tuple $\langle t_i, p_i\rangle$, simply perform $s$ independent biased coin flips (each with success probability $p_i$) setting $G_j = G_j \cup \{t_i\}$ if the $j^{th}$ coin flip succeeds. For our universal probabilistic-stream estimator for $F()$, we do not store the possible worlds $G_j$; instead, the tuples generated for these (grounded) streams are fed into $s$ parallel instances of streaming estimation algorithms for $F()$. Let $\{\hat{F}(G_j) : j = 1, \ldots, s\}$ de-

note the outputs of these streaming estimators; then, we estimate $\mathsf{E}_G[F(\mathcal{S})]$ and $\mathsf{Var}_G[F(\mathcal{S})]$ as the *sample mean* $\hat{\mu}$ and the *sample variance* $\hat{\sigma}$, respectively, where:

$$\hat{\mu} = \frac{1}{s}\sum_{j=1}^{s} \hat{F}(G_j) \quad \text{and} \quad \hat{\sigma}^2 = \sum_{j=1}^{s} \frac{\hat{F}(G_j)^2}{s-1} - \hat{\mu}^2.$$

The following theorem establishes the accuracy properties of our universal probabilistic stream estimator.

THEOREM 3.1. *If $\hat{F}(G)$ is an unbiased streaming estimator for $F(G)$, then $\hat{\mu}$ and $\hat{\sigma}$ are unbiased estimators for $\mathsf{E}_G[F(S)]$ and $\mathsf{Var}_G[F(S)]$, respectively. Moreover, using $s = O\left(\frac{1}{\epsilon^2}\frac{\mathsf{Var}_G[F(S)]}{\mathsf{E}^2_G[F(S)]}\right)$ samples, $\hat{\mu}$ provides an $(\epsilon, O(1))$-approximation of $\mathsf{E}_G[F(S)]$.* ∎

In other words, by sampling $s = O(\frac{1}{\epsilon^2}\frac{\mathsf{Var}_G[F(\mathcal{S})]}{\mathsf{E}^2_G[F(\mathcal{S})]})$ possible worlds and passing each to a separate instance of a streaming estimation algorithm for $F()$ over deterministic data streams, our sample-mean estimator can guarantee $\epsilon$-relative error with constant probability. We can amplify this to any user-defined success probability by simply repeating $O(\log(1/\delta))$ times and taking the median result; standard Chernoff-bounds arguments then guarantee that the probability of failure is less than $\delta$. By its sample-size requirement, it is easy to see that the estimation quality of our universal estimator depends crucially on the ratio of $\mathsf{Var}_G[F(\mathcal{S})] = \mathsf{E}_G[F^2(\mathcal{S})] - \mathsf{E}^2_G[F(\mathcal{S})]$ to $\mathsf{E}^2_G[F(\mathcal{S})]$. We will see subsequently that for the frequency moments we consider in this paper, $\mathsf{Var}_G[F_k(\mathcal{S})] \leq \mathsf{E}_G[F_k(\mathcal{S})]$ for $k = 0, 1$, and $\mathsf{Var}_G[F_2(\mathcal{S})] = O(\mathsf{E}^{3/2}_G[F_2(\mathcal{S})])$. So this ratio will be small, and decreasing as $\mathsf{E}_G[F_k(\mathcal{S})]$ increases.

It is tempting to try to apply the same technique to derive bounds for the accuracy of $\hat{\sigma}^2$ for approximating the distribution variance $\mathsf{Var}_G[F(\mathcal{S})]$ of the probabilistic aggregate $F()$ — after all, the variance is just another expectation. Unfortunately, the bounds resulting from such an approach are considerably worse. Observe that the variance computation is equivalent to computing the difference of two values: $\sum_{j=1}^{s} \frac{\hat{F}(G_j)^2}{s-1}$ and $\hat{\mu}^2$. The first of these is an unbiased estimator for the quantity $\mathsf{Var}_G(\mathcal{S}) + \mathsf{E}^2_G[F(\mathcal{S})]$ and the second for the quantity $\mathsf{E}^2_G[F(\mathcal{S})]$. However, for the frequency moments that we study, $\mathsf{E}^2_G[F_k(\mathcal{S})]$ can be much larger than $\mathsf{Var}_G[F_k(\mathcal{S})]$: we stated above that $\mathsf{Var}_G[F_k(\mathcal{S})] \leq \mathsf{E}[F_k(\mathcal{S})]$ for $k = 0, 1$. Suppose we chose $s$ large enough to ensure that $\mathsf{E}_G[F_k(\mathcal{S})]$ is $(\epsilon', \delta)$ approximated, then $\mathsf{E}_G[F_k(\mathcal{S})]^2$ is $(3\epsilon', \delta)$ approximated. Because we the quantity we are estimating can be much smaller than this, we need a much more accurate approximation in order to $(\epsilon, \delta)$ approximate $\mathsf{Var}_G[F_k(\mathcal{S})]$. Thus we need to choose an $\epsilon'$ such that $\epsilon'\mathsf{E}_G[F_k(\mathcal{S})]^2 \leq \epsilon\mathsf{Var}_G[F_k(\mathcal{S})]$. Using the bound from Theorem 3.1, this means we need $\Omega(\mathsf{E}_G[F_k(\mathcal{S})])$ samples for $\mathsf{Var}_G[F_0(\mathcal{S})]$ and $\mathsf{Var}_G[F_1(\mathcal{S})]$, and $\Omega(\mathsf{E}^{1/2}_G[F_k(\mathcal{S})])$ samples for $\mathsf{Var}_G[F_2(\mathcal{S})]$. The bounds from this analysis are so weak that they are of little practical use. We shall also see experimentally that the estimations of variance using the universal sampling algorithm are very poor in practice.

In the remainder of the paper, we show several cases where more intelligent, "aggregate-aware" streaming estimation algorithms can be developed that perform significantly better than the above universal sampling-based strategy. We start by discussing solutions and applications for the simple case of the first frequency moment.

## 3.2 $F_1$: Counts, Quantiles, and Heavy Hitters

For a conventional, deterministic data stream $G$, computing the first frequency moment $F_1(G)$ is trivial: it is just the count of elements in the stream, which can be computed exactly using a single counter. Similarly, computing the (exact) distribution moments of $F_1(\mathcal{S})$ over a probabilistic stream $\mathcal{S}$ turns out to be quite straightforward. Specifically, consider the probabilistic stream $\mathcal{S} = (\langle t_i, p_i\rangle : 1 \leq i \leq N)$. For the purposes of first-moment estimation over $\mathcal{S}$, each streaming uncertain tuple $\langle t_i, p_i\rangle$ can be seen as a Bernoulli variable $B_i$ that takes the value 1 (i.e,. the tuple is in the possible world) with probability $p_i$, and 0 otherwise. Then, by linearity of expectation and the independence of probabilistic tuples in the stream:

$$\begin{aligned} \mathsf{E}_G[F_1(\mathcal{S})] &= \sum_{i=1}^{N} \mathsf{E}[X_i] = \sum_{i=1}^{N} p_i \ , \quad \text{and} \\ \mathsf{Var}_G[F_1(\mathcal{S})] &= \sum_{i=1}^{N} \mathsf{Var}[X_i] = \sum_{i=1}^{N} p_i(1-p_i), \end{aligned}$$

and hence $\mathsf{Var}_G[F_1(\mathcal{S})] \leq \mathsf{E}_G[F_1(\mathcal{S})]$. The above linearity immediately implies that we can compute the *exact* expectation and variance of $F_1(\mathcal{S})$ using only two variables. Earlier work on probabilistic data has already made similar observations for such linear aggregates, at least for the case of computing the expectation (e.g., see [16]). It is not difficult to see that, thanks to this linearity, we can also leverage known (deterministic) stream synopses and results to provide strong estimation guarantees for other important *count-based aggregates* over probabilistic data streams. We now summarize our key results in this setting.

Given a probabilistic data stream $\mathcal{S}$ and tuple $t \in [M]$, let $f_t$ denote the *point frequency* (i.e., number of occurrences) of $t$ in $\mathcal{S}$ — this is again a random variable over $\mathrm{grnd}(\mathcal{S})$. If $t \in [M]$ is given in advance, we can easily compute $\mathsf{E}_G[f_t]$ and $\mathsf{Var}_G[f_t]$ using two simple counters, as above. More interestingly, when $t$ is *not* known beforehand, it is possible to build small-space streaming synopses over $\mathcal{S}$ to estimate such point-frequency moments to within $\epsilon$ factors of the corresponding stream moments. (In other words, we can estimate point-frequencies accurately in small space as long as they are "large" with respect to the overall (expected) count of elements in the stream; these correspond to the typical streaming error guarantees for point estimation in the deterministic setting [7, 20].)

LEMMA 3.2. *Using space of only $O(\frac{1}{\epsilon})$, it is possible build a synopsis over a probabilistic data stream $\mathcal{S}$ that, given $t \in [M]$, can return estimates $\hat{\mu}_t$ and $\hat{\sigma}_t$ such that (with probability 1)*

$$\begin{aligned} \mathsf{E}_G[f_t] \leq \ \hat{\mu}_t \ &\leq \mathsf{E}_G[f_t] + \epsilon\, \mathsf{E}_G[F_1(\mathcal{S})] \quad \textit{and} \\ \mathsf{Var}_G[f_t] \leq \ \hat{\sigma}_t \ &\leq \mathsf{Var}_G[f_t] + \epsilon\, \mathsf{Var}_G[F_1(\mathcal{S})]. \end{aligned}$$

∎

*Proof:*By linearity of the aggregate, it is not difficult to see that estimating $\mathsf{E}_G[f_t]$ and $\mathsf{Var}_G[f_t]$ corresponds to a standard point query over a data stream containing fractional values — namely, the $p_i$ values for $\mathsf{E}_G[f_t]$ and the $p_i(1-p_i)$ values for $\mathsf{Var}_G[f_t]$. Thus, one can immediately apply known streaming synopses, such as the *Count-Min sketch* [7]. This provides estimates with errors proportional to $\epsilon\sum_{i=1}^{n} p_i = \epsilon\mathsf{E}_G[F_1(\mathcal{S})]$ and $\epsilon\mathsf{Var}_G[F_1(\mathcal{S})]$, for expectation and variance (respectively), but with only *probabilistic guarantees*. Well known techniques such as *Lossy Counting* [20] and Misra-Gries [22] do not naturally accommodate a stream of fractional values (they assume a stream of unitary updates). However, the algorithms of Suri *et al.* [24] and Metwally *et al.* [21] allow arbitrary update values, and can be applied to extract point estimates from a stream. They can be implemented so that each update takes expected time $O(1)$, and uses space $O(\frac{\log M}{\epsilon})$ [6] amd $O(\frac{1}{\epsilon})$ respectively. Thus, we can provide the required estimates $\hat{\mu}_t$ and $\hat{\sigma}_t$ in this time and space. ∎

Using standard streaming arguments, the above lemma directly implies space/time-efficient probabilistic-stream estimation algorithms for other interesting count-based aggregates, such as *approximate heavy hitters* and *approximate quantiles*. The development resembles that for the conventional streaming version of the problem — more details can be found, e.g., in [7, 24].

COROLLARY 3.3. *Assume a constant $\phi \in (0,1)$, and a desired accuracy guarantee $\epsilon < \phi$. We state the following results:*

*1.* Approximate $\phi$-Heavy Hitters: *To return all elements $t$ such that $\mathsf{E}_G[f_t] \geq (\phi+\epsilon)\mathsf{E}_G[F_1(\mathcal{S})]$, and no item $t'$ such that $\mathsf{E}_G[f_{t'}] \leq (\phi-\epsilon)\mathsf{E}_G[F_1(\mathcal{S})]$ can be done in space $O(\frac{1}{\epsilon})$.*

*2.* Approximate $\phi$-Quantiles: *To return all items $t$ such that $(\phi+\epsilon)\mathsf{E}_G[F_1(\mathcal{S})] \leq \mathsf{E}_G[\sum_{i=1}^{t} f_i] \leq (\phi+\epsilon)\mathsf{E}_G[F_1(\mathcal{S})]$ can be done in space $O(\frac{\log M}{\epsilon})$.*

*The same estimation guarantees also hold for variance estimates (i.e., replacing $\mathsf{E}_G[]$ with $\mathsf{Var}_G[]$).*

# 4. ESTIMATING COUNT DISTINCT ($F_0(\mathcal{S})$)

As noted in Section 3, the case of of the first frequency moment $F_1$ and related count-based estimation problems was easy to reduce to prior streaming work, due to linearity of the aggregate. Unfortunately, other frequency moments are less straightforward. In this section, we consider the case of $F_0$, the number of distinct items over the probabilistic stream $\mathcal{S}$. It is important to note that computing $\mathsf{E}_G[F_0]$ is very different from computing the number of distinct tuples seen in $\mathcal{S}$; for instance, $\mathcal{S}$ could comprise many millions of tuples with distinct values, but if each $p_i$ is minuscule (say, of the order of $10^{-10}$), $\mathsf{E}_G[F_0(\mathcal{S})]$ could still be much less than one. Also, the linearity property that allowed us to simply sum probabilities in the case of $F_1$ is no longer valid — intuitively, this is because, for $F_0$, the number of occurrences of a distinct tuple $t \in [M]$ in a stream is immaterial as long as the tuple appears *at least once*. Thus, our streaming algorithms must track these probabilities of occurrence for distinct tuples in the domain. Thanks to tuple independence, this turns out to be quite simple to do if one assumes *linear* space ($O(M)$).

LEMMA 4.1. *Using space $O(M)$, a streaming algorithm can compute the exact values of $\mathsf{E}_G[F_0(\mathcal{S})]$ and $\mathsf{Var}_G[F_0(\mathcal{S})]$ over a probabilistic data stream $\mathcal{S}$.* ∎

*Proof:*Let $p_t$ denote the probability that element $t \in [M]$ is observed *at least once* in any grounded instance of the stream $\mathcal{S} = (\langle t_i, p_i\rangle : i = 1, \ldots, N)$; that is, $p_t = \sum_{t\in G, G\in \mathrm{grnd}(\mathcal{S})} \Pr[G]$. By tuple independence, it is not difficult to see that this probability can also be expressed as $p_t = 1 - \prod_{i:t_i=t}(1-p_i)$, i.e., one minus the probability that *none* of the instances of $t$ are actually materialized. This probability can be incrementally computed as tuples $\langle t_i, p_i\rangle$ from $\mathcal{S}$ are streaming by. Initially, set $p_t = 0$ for all $t = 1, \ldots, M$. Now, suppose the next tuple in $\mathcal{S}$ is $\langle t_i, p_i\rangle$, with $t_i = t$; then, we update $p_t \leftarrow p_t(1-p_i) + p_i$, to reflect the updated probability of seeing a $t$ tuple. It is easy to verify that this rule correctly maintains the probability of occurrence for $t$ as defined above. The expected number of distinct items is exactly $\mathsf{E}_G[F_0(S)] = \sum_{t=1}^{M} p_t$. For $\mathsf{Var}_G[F_0(\mathcal{S})]$, observe that, for count-distinct estimation, each $t \in [M]$ is just a Bernoulli random variable with parameter $p_t$, and, thus, its variance is simply $p_t(1-p_t)$. Since $F_0(\mathcal{S})$ is basically the summation of these Bernoulli random variables, and by tuple independence, we have $\mathsf{Var}_G[F_0(\mathcal{S})] = \sum_{t=1}^{M} p_t(1-p_t)$, and hence $\mathsf{Var}_G[F_0(\mathcal{S})] \leq \mathsf{E}_G[F_0(\mathcal{S})]$. ∎

EXAMPLE 4.2. *Consider our example probabilistic stream $\mathcal{S} = (\langle x, \frac{1}{2}\rangle, \langle y, \frac{1}{4}\rangle, \langle y, \frac{1}{3}\rangle)$. We have $p_x = \frac{1}{2}$, and $p_y = \frac{3}{4}\cdot\frac{1}{3} + \frac{1}{4} = \frac{1}{2}$, so $\mathsf{E}_G[F_0(\mathcal{S})] = 1$ and $\mathsf{Var}_G[F_0(\mathcal{S})] = \frac{1}{2}$. We can verify these results against the probabilities for the grounded streams $\Pr[G]$ (see Example 2.1). From these probabilities, we have $\mathsf{E}_G[F_0(\mathcal{S})] = 1(\Pr[(x)] + \Pr[(y)] + \Pr[(y,y)]) + 2(\Pr[(x,y)] + \Pr[(x,y,y)]) = 1(\frac{1}{4} + \frac{5}{24} + \frac{1}{24}) + 2(\frac{5}{24} + \frac{1}{24}) = 1$. Similarly, $\mathsf{Var}_G[F_0(\mathcal{S})] = \mathsf{E}_G[F_0^2(\mathcal{S})] - \mathsf{E}_G^2[F_0(\mathcal{S})] = 1(\frac{12}{24}) + 4(\frac{6}{24}) - 1^2 = \frac{1}{2}$.*

**Using the Universal, Sampling-Based Estimator for $\mathsf{E}_G[F_0(\mathcal{S})]$.** We show the guarantees that our universal estimation algorithm (based on possible-worlds sampling) of Section 3 can provide for $F_0$ estimation over $\mathcal{S}$. From Theorem 3.1, we know that the universal estimator can guarantee a good approximation for $\mathsf{E}_G[F_0(\mathcal{S})]$ by sampling $s = O(\frac{1}{\epsilon^2}\frac{\mathsf{Var}_G[F_0(\mathcal{S})]}{\mathsf{E}_G^2[F_0(\mathcal{S})]})$ grounded streams $G \in \mathrm{grnd}(\mathcal{S})$, and using an efficient, guaranteed-error streaming estimator for $F_0(G)$ on each. From the above analysis based on independent Bernoulli random variables, we see that

$$\frac{\mathsf{Var}_G[F_0(\mathcal{S})]}{\mathsf{E}_G^2[F_0(\mathcal{S})]} = \frac{\sum_{t=1}^{M} p_t(1-p_t)}{(\sum_{t=1}^{M} p_t)^2} \leq \frac{\sum_{t=1}^{M} p_t}{(\sum_{t=1}^{M} p_t)^2} = \frac{1}{\mathsf{E}_G[F_0(\mathcal{S})]}.$$

In other words, the number of grounded-stream samples (and estimators) needed to guarantee small relative errors is inversely proportional to $\mathsf{E}_G[F_0(\mathcal{S})]$ (the quantity that we want to estimate). This is actually quite intuitive: Consider a probabilistic stream $\mathcal{S}$ comprising $N$ distinct tuples $t_i$, where all tuples have the *same* probability $p_i = p$. If $p$ is very small, which, of course, implies that $\mathsf{E}_G[F_0(\mathcal{S})]$ is small, then a large number of samples is needed to ensure a sampling estimate with small relative error; for instance, if $N = 10^4$ and $p = 10^{-10}$, then at least $10^6$ ($= \frac{1}{\mathsf{E}_G[F_0(\mathcal{S})]}$) samples are needed, on average, just to sample a *non-empty* possible world (i.e., give a non-zero estimate)! Whereas, if $p = 1$ (i.e., $\mathsf{E}_G[F_0(\mathcal{S})] = N$), then all tuples are *deterministic*, and a single instance of the $F_0(G)$ estimator (i.e., $s = 1$).

As our experiments show, in practice, $\mathsf{E}_G[F_0(\mathcal{S})]$ is typically of moderate size, and a small sample size $s$ is often sufficient to get a good approximation. On the other hand, as discussed in Section 3, such error guarantees unfortunately do not carry over to the case of the $F_0$-variance $\mathsf{Var}_G[F_0(\mathcal{S})]$.

## 4.1 The Probabilistic FM (pFM) Sketch

We now introduce a novel algorithm for $F_0$ estimation over probabilistic data streams. Our algorithm guarantees randomized $(\epsilon,\delta)$-estimates for both $\mathsf{E}_G[F_0(\mathcal{S})]$ and $\mathsf{Var}_G[F_0(\mathcal{S})]$, while using space that is only *poly-logarithmic* in the size of the stream $\mathcal{S}$, and independent of the value of $\mathsf{E}_G[F_0(\mathcal{S})]$.

Our technique is inspired by the popular *Flajolet-Martin (FM) algorithm* [11] for estimating the number of distinct elements over a deterministic data stream $G$. Briefly, letting $[M]$ denote the domain of the stream (as earlier), the FM algorithm employs a family $\mathcal{H}$ of hash functions $h : [M] \longrightarrow \{1, \ldots, \log M\}$, such that, for any $x \in [M]$ $\Pr[h(x) = i] = 2^{-i}$ (where probability is defined over the family of hash-function choices $\mathcal{H}$). The basic stream synopsis maintained by the FM algorithm (known as an *FM sketch*) is a bitmap of size $O(\log M)$ that is maintained using a specific hash function $h() \in \mathcal{H}$. Specifically, the sketch is initialized to all zeros and, for each incoming element $x$ in the stream, the bit at location $h(x)$ is turned on. By the properties of $h()$, we expect a fraction of $2^{-i}$ of the distinct values in the stream to map to location $i$ of the sketch; that is, we expect $\frac{1}{2}$ of the distinct values to land in bit 1, $\frac{1}{4}$ in bit 2, and so on. Thus, if we let $\lambda$ denote the highest bit location that has been turned on in the bitmap, then $2^{\lambda}$ is a good indicator of the number of distinct elements in $G$. Using several independent FM sketches (with different hash-function choices from $\mathcal{H}$), it is possible to boost estimation accuracy and confidence to user-defined $(\epsilon,\delta)$ levels (e.g., [4]).

Our proposed sketch synopsis for *probabilistic* data streams, termed *probabilistic FM (pFM) sketch*, is based on similar ideas. We employ the same class of hash functions $\mathcal{H}$ as the basic FM algorithm, and define a pFM sketch as an array $\mathsf{pFM}[]$ comprising $O(\log M)$ *(real-valued) probability entries* maintained using a hash function

**procedure** pFMestimate ( sketchMatrix[$1 \ldots s_1, 1 \ldots s_2$] )
**Input:** Matrix of $s_1 \times s_2$ independent probabilistic FM sketches for probabilistic stream $\mathcal{S}$, where $s_1 = O(\frac{1}{\epsilon^2})$ and $s_2 = O(\log(1/\delta))$.
**Output:** $(\epsilon, \delta)$-approximate estimate for $\mathsf{E}_G[F_0(\mathcal{S})]$.
1. let $\hat{d}_{ij}$ := basic pFM estimate of $E_G[F_0(\mathcal{S})]$ using the sketch of $\mathcal{S}$ at sketchMatrix[$i, j$] (see Eqn. (1))
2. $\hat{d} := \sum_{i,j} \hat{d}_{ij}/(s_1 \cdot s_2)$
3. $k^* := \lceil \log(8\hat{d}) \rceil$ *// find inference level*
4. **for** $j = 1$ **to** $s_2$ **do**
5. $Y_j := \sum_{i=1}^{s_1}$ sketchMatrix$[i, j][k^*]/s_1$
6. **return**( $2^{k^*} \times$ median$\{Y_1, \ldots, Y_{s_2}\}$ )

**Figure 1: $(\epsilon, \delta)$ Estimation Algorithm for $E_G[F_0(\mathcal{S})]$.**

$h() \in \mathcal{H}$. The pFM[] array is initialized to all zeros, and, for each streaming pair $\langle t_i, p_i \rangle$ in the probabilistic data stream $\mathcal{S}$, we update the pFM$[h(t_i)]$ entry by setting

$$\mathsf{pFM}[h(t_i)] \leftarrow \mathsf{pFM}[h(t_i)](1 - p_i) + p_i.$$

Let $h^{-1}(j)$ the set of tuples in $[M]$ that map onto bit $j$ through $h()$; that is $h^{-1}(j) = \{t \in [M] : h(t) = j\}$. Using a simple inductive argument, we can prove that, using the update rule above,

$$\mathsf{pFM}[j] = 1 - \prod_{t_i \in \mathcal{S}}(1 - p_i) = \Pr[h^{-1}(j) \neq \phi],$$

where the last equality follows from tuple independence. We compute a basic estimate for $\mathsf{E}_G[F_0(\mathcal{S})]$ from the pFM[] array as

$$\hat{d} \;=\; \sum_{j=1}^{O(\log M)} 2^j \mathsf{pFM}[j] \prod_{k=j+1}^{O(\log M)} (1 - \mathsf{pFM}[k]). \qquad (1)$$

Intuitively, the above formula computes an expectation over the $2^j$ estimates for all possible bitmap locations $j$ using the probability that $j$ is the highest non-empty location in the bitmap. In a sense, this can be seen as an "probabilistic expectation" version of the original FM idea. Our basic pFM estimate can be shown to guarantee constant estimation error with constant probability:

THEOREM 4.3. *Using a single pFM sketch of size $O(\log M)$, our basic pFM estimation algorithm outputs an estimate $\hat{d}$ for $\mathsf{E}_G[F_0(\mathcal{S})]$ such that $\Pr[\frac{1}{c} \leq \frac{\hat{d}}{\mathsf{E}_G[F_0(\mathcal{S})]} \leq c] \geq 1 - 2/c$.* ∎

**An $(\epsilon, \delta)$-Approximate Estimator for $\mathsf{E}_G[F_0(\mathcal{S})]$.** We now give a streaming $(\epsilon, \delta)$-approximate estimation algorithm for $\mathsf{E}_G[F_0(\mathcal{S})]$. Briefly, our algorithm employs a number of independent pFM sketches built over the $\mathcal{S}$ probabilistic stream. The basic idea behind our estimation process is to first use our constant-factor/probability basic estimator for $\mathsf{E}_G[F_0(\mathcal{S})]$ (described above) in order to identify an appropriate *inference level* $k^*$ in the pFM sketch structures. Our goal in determining this inference level is to ensure that the probability that more than one distinct elements map onto that level (over all possible worlds $G$ and choices of hash function $h()$) is small. We can achieve this by choosing $k^*$ to be a few levels higher than $\log \hat{d}$, where $\hat{d}$ is the estimate returned by averaging basic estimates across the individual pFM sketches. Then, we use averaging and median selection over the probabilities at level $k^*$ across all pFM sketches in order to produce an accurate estimate of $\mathsf{E}_G[F_0(\mathcal{S})]$. Pseudocode for our $(\epsilon, \delta)$ estimation algorithm is in Figure 1.

THEOREM 4.4. *Algorithm* pFMestimate *returns an $(\epsilon, \delta)$-approximate estimate for $E_G[F_0(\mathcal{S})]$ using $O(\frac{\log(1/\delta)}{\epsilon^2})$ independent pFM sketch synopses for the probabilistic data stream $\mathcal{S}$.* ∎

*Proof (sketch):* We briefly discuss the main ideas behind our proof — the complete details, optimizations of constant factors, etc., are deferred to the full paper. We start with some notation. Fix a specific level $k$ in the pFM sketch. Let $X_k$ be an indicator random variable (RV), over all $G$ and $h()$, for the event that $\geq 1$ distinct stream elements map onto level $k$, and let $Z_k$ denote a RV (over all $G$ and $h()$) for the *number of distinct values* in the stream mapping onto level $k$. Now, fix a specific choice of hash function $h()$ and let $p_k(h) = \mathsf{pFM}[k]$ denote the (incrementally) computed probability at level $k$ of the sketch; this is exactly the fraction of possible worlds $G$ for which the $k^{th}$ bit of the corresponding FM sketch is on, i.e., $p_k(h) = \mathsf{E}_G[X_k(h)]$. In other words, $p_k(h)$ is a sample point (for a fixed $h()$) from the distribution of $X_k$ — note that the expectation of $X_k$ has a highly non-linear relationship with $F_0$; more specifically:

$$\mathsf{E}_{G,h}[X_k] = \sum_G \Pr[G] \cdot (1 - (1 - 2^{-k})^{F_0(G)})$$

since each distinct element in $G$ maps independently onto level $k$ with probability $1/2^k$. (We use the $G$, $h$ subscripts to denote expectation taken over the space of possible worlds and hash functions, respectively.) So, while we can use the sample points $p_k(h)$ to estimate $\mathsf{E}_{G,h}[X_k]$, there is no obvious way to go from that to our target quantity $E_G[F_0(\mathcal{S})] = \sum_G \Pr[G] \cdot F_0(G)$.

On the other hand, the expectation of $Z_k$ has an "easy" linear relationship to $E_G[F_0(\mathcal{S})]$, since

$$\mathsf{E}_{G,h}[Z_k] = \sum_G \Pr[G] F_0(G) \Pr[\text{level} = k] = 2^{-k} \mathsf{E}_G[F_0(\mathcal{S})] \quad (2)$$

Our goal is to pick out an inference level $k^*$ such that, with some constant probability, $\mathsf{E}_{G,h}[X_k] = \mathsf{E}_{G,h}[Z_k]$. We achieve that using our initial constant factor approximation to $E_G[F_0(\mathcal{S})]$ — based on our analysis, and using an additional averaging step over $m$ iid instantiations to reduce the variance (and, thus, the probability of error in the Chebyshev bound) by a factor of $m$, we can get an estimate $\hat{d}$ of $E_G[F_0(\mathcal{S})]$ such that $4E_G[F_0] < 4c\hat{d} < 4c^2 E_G[F_0]$ with probability at least $1 - \frac{m+1}{mc}$. In what follows, we assume our approximation bounds on $\hat{d}$ hold and add $\frac{m+1}{mc}$ to the probability of error. Define $k^* = \lceil \log(4c\hat{d}) \rceil$. Then, our bounds on $\hat{d}$ imply that

$$\frac{2^{k^*-3}}{c^2} < \mathsf{E}_G[F_0(\mathcal{S})] < 2^{k^*-2} \text{ or } \frac{1}{8c^2} < \frac{\mathsf{E}_G[F_0(\mathcal{S})]}{2^{k^*}} < \frac{1}{4} \quad (3)$$

Now, consider the RV $Z_{k^*}$. From Equations (2)–(3), application of the Markov inequality gives

$$\Pr_{G,h}[Z_{k^*} > 1] \;<\; 2^{-k^*} \mathsf{E}_G[F_0(\mathcal{S})] \;<\; \tfrac{1}{4}.$$

So at level $k^*$, we have $Z_{k^*} \leq 1$ (i.e., $Z_{k^*} \in \{0, 1\}$) with probability $\geq 3/4$. Again, we assume $Z_{k^*} \in \{0, 1\}$ and adjust the final probability of error by adding $1/4$. In this case, at level $k^*$,

$$\mathsf{E}_{G,h}[Z_{k^*}] \;=\; \mathsf{E}_{G,h}[X_{k^*}] \;=\; \mathsf{E}_h[p_{k^*}(h)] \;=\; 2^{-k^*} \mathsf{E}_G[F_0(\mathcal{S})]$$

(where the last equality follows from Equation (2)). Now, over all possible choices of $h()$, the RV $p_{k^*}(h)$ satisfies

$$\mathsf{Var}_h[p_{k^*}(h)] \;\leq\; \mathsf{E}_h[p_{k^*}^2(h)] \;\leq\; \mathsf{E}_h[p_{k^*}(h)] \;=\; 2^{-k^*} \mathsf{E}_G[F_0(\mathcal{S})],$$

(since ${}_{k^*}(h) \in [0, 1]$). Assuming $m$ iid instantiations (using independently chosen hash functions $h_1(), h_2(), \ldots, h_m()$) of our probabilistic FM sketch, we define the sample-average estimator $\hat{p} = \sum_{i=1}^m p_{k^*}(h_i)/m$ for the expectation $\mathsf{E}_h[p_{k^*}(h)]$ (Line 5 in our pFMestimate algorithm). From the Chebyshev bound:

$$\begin{aligned} \Pr\left[|2^{k^*}\hat{p} - \mathsf{E}_G[F_0(\mathcal{S})]| > \epsilon \mathsf{E}_G[F_0(\mathcal{S})]\right] &< \frac{\mathsf{Var}_h[p_{k^*}(h)]}{m\epsilon^2 \mathsf{E}_h^2[p_{k^*}(h)]} \\ &\leq \frac{2^{k^*}}{m\epsilon^2 \mathsf{E}_G[F_0(\mathcal{S})]} \;<\; \frac{8c^2}{m\epsilon^2} \end{aligned}$$

(where the last inequality follows from the above equation).

So $2^{k^*}\hat{p}$ is a randomized $\epsilon$-relative error estimator of $\mathsf{E}_G[F_0(\mathcal{S})]$ with an overall error probability upper bounded by $\frac{m+1}{mc} + \frac{1}{4} + \frac{8c^2}{m\epsilon^2}$, where $m$ is the number of iid pFM sketches used for averaging $p_{k^*}(h_i)$ instances, and $c$ is a constant $> 2$. Assuming a small $\epsilon$ (say, $\leq 1/4$), we can now choose $m$ and $c$ to ensure that this error

probability is upper bounded by a constant $< 1/2$. We can now use $O(\log(1/\delta))$ iid instantiations of the above procedure to bring the overall error probability down to $\delta$ (Lines 4–6 of pFMestimate). Thus, we have an $(\epsilon, \delta)$ randomized estimator for $\mathsf{E}_G[F_0(\mathcal{S})]$ using $O(\frac{\log(1/\delta)}{\epsilon^2})$ pFM sketch summaries. ∎

**Estimating** $\mathsf{Var}_G[F_0(\mathcal{S})]$**.** Computing $\mathsf{Var}_G[F_0(\mathcal{S})]$ initially seems harder than computing $\mathsf{E}_G[F_0(\mathcal{S})]$. As we demonstrate, however, the complexity of building a good estimator is no harder, by reduction to the expectation-estimation problem.

LEMMA 4.5. *Given a method to $(\epsilon, \delta)$ approximate $\mathsf{E}_G[F_0(\mathcal{S})]$, the quantity $\mathsf{Var}_G[F_0\mathcal{S})]$ can be estimated (using the same space) with error at most $3\epsilon\mathsf{E}_G[F_0(\mathcal{S})]$ with probability at least $1 - 2\delta$.* ∎

*Proof:*Given a probabilistic stream $\mathcal{S} = (\langle t_i, p_i\rangle : i = 1, \ldots, N)$, we define a new stream $\mathcal{S}_2 = (\langle t_i, 2p_i - p_i^2\rangle : 1 \le i \le N)$, and compute $\mathsf{E}_G[F_0(\mathcal{S}_2)]$. We claim that $\mathsf{E}_G[F_0(\mathcal{S}_2)] - \mathsf{E}_G[F_0(\mathcal{S})] = \mathsf{Var}_G[F_0(\mathcal{S})]$. We prove this claim by studying properties of $\overline{p}_t$, where $\overline{p}_t$ denotes the probability that $t$ does not occur in a ground stream of $\mathcal{S}$, i.e. $\overline{p}_t = \sum_{t\notin G, G\in \mathrm{grnd}(\mathcal{S})} \Pr[G] = \prod_{i,t_i=t}(1-p_i)$. Thus, $\mathsf{E}_G[F_0(\mathcal{S})] = \sum_{t=1}^M p_t = \sum_{t=1}^M 1 - \overline{p}_t$ and $\mathsf{E}_G[F_0(\mathcal{S}_2)] = \sum_{t=1}^M 1 - \sum_{G\in\mathrm{grnd}(\mathcal{S}_2)}^{t\notin G,} \Pr[G] = \sum_{t=1}^M 1 - \overline{p}_t^2$.

$$\text{So } \mathsf{E}_G[F_0(\mathcal{S}_2)] - \mathsf{E}_G[F_0(\mathcal{S})] = \sum_{t=1}^M (1-\overline{p}_t^2) - \sum_{t=1}^M (1-\overline{p}_t) = \sum_{t=1}^M p_t(1-p_t) = \mathsf{Var}_G[F_0(\mathcal{S})],$$

since Lemma 4.1 showed that $\mathsf{Var}_G[F_0(\mathcal{S})] = \sum_{t=1}^M p_t(1-p_t)$. We observe that $1 - \overline{p}_t^2 = 2p_t - p_t^2 \le 2p_t$, and thus

$$\mathsf{E}_G[F_0(\mathcal{S}_2)] = \sum_{t=1}^M 1 - \overline{p}_t^2 \le 2\sum_{t=1}^M p_t = 2\mathsf{E}_G[F_0(\mathcal{S})].$$

Thus, if we $(\epsilon, \delta)$ approximate $\mathsf{E}_G[F_0(\mathcal{S}_2)]$, the error is at most $2\epsilon\mathsf{E}_G[F_0(\mathcal{S})]$ with probability at least $1 - \delta$. Combining this with an $(\epsilon, \delta)$ approximation of $\mathsf{E}_G[F_0(\mathcal{S})]$ yields an approximation of $\mathsf{Var}_G[F_0(\mathcal{S})]$ which is within $3\epsilon\mathsf{E}_G[F_0(\mathcal{S})]$ with probability at least $1 - 2\delta$ by the union bound. ∎

Our proof argument demonstrates that we can compute $\mathsf{Var}_G[F_0(\mathcal{S})]$ using two distinct pFM sketches: one for estimating $\mathsf{E}_G[F_0(\mathcal{S})]$ over the original stream $\mathcal{S} = (\langle a_i, p_i\rangle : 1 \le i \le N)$, and one for estimating $\mathsf{E}_G[F_0(\mathcal{S}_2)]$ over a "modified" stream $\mathcal{S}_2 = (\langle a_i, 2p_i - p_i^2\rangle : 1 \le i \le N)$. But we can be even more space efficient: observe that our pFM sketch summaries find

$$\mathsf{pFM}[b] = \Pr[h^{-1}(b) \neq \phi \text{ in } \mathrm{grnd}(\mathcal{S})] = 1 - \Pr[h^{-1}(b) = \phi \text{ in } \mathrm{grnd}(\mathcal{S})] = 1 - \prod_{t,h(t)=b} \overline{p}_t$$

If we built a pFM summary $\mathsf{pFM}_2$ of $\mathcal{S}_2$ using the same hash function $h()$, we want to find $\mathsf{pFM}_2[b] = 1 - \prod_{t,h(t)=b} \overline{p}_t^2$. But,

$$1 - \prod_{t,h(t)=b} \overline{p}_t^2 = 1 - (\prod_{t,h(t)=b} \overline{p}_t)^2 = 1 - (1 - \mathsf{pFM}[b])^2.$$

Thus, we can actually estimate *both* $\mathsf{E}_G[F_0(\mathcal{S})]$ and $\mathsf{E}_G[F_0(\mathcal{S}_2)]$ using the information stored in a *single* pFM sketch data structure.

# 5. SECOND MOMENT AND JOIN SIZE

We next consider the complexity of computing the expectation and variance of the second frequency moment of a probabilistic stream. Later, we will go on to study the related question of the expected join size between two probabilistic streams.

## 5.1 Expectation of $F_2$ using pAMS

We introduce an estimation technique based on the randomized sketches of Alon *et al.* [2], which we call "probabilistic AMS", or pAMS for short. We proceed by reducing the problem from computing over exponentially many ground streams to tracking information about the up to $M$ distinct tuples in the stream, then show how to approximate this information in sublinear space.

THEOREM 5.1. *We can compute an estimator $\hat{F}_2$ for $\mathsf{E}_G[F_2(\mathcal{S})]$ such that $|\mathsf{E}_G[F_2(\mathcal{S})] - \hat{F}_2| \le \epsilon\mathsf{E}_G[F_2(\mathcal{S})]$ with probability at least $1 - \delta$ in space $O(\frac{1}{\epsilon^2}\log 1/\delta)$.* ∎

*Proof:*We first analyze the problem to show an algorithm that computes the exact value of $\mathsf{E}_G[F_2(\mathcal{S})]$ in space linear in $M$. Let $\mathcal{S}_t$ denote the substream of $\mathcal{S}$ corresponding only to tuples $t$, i.e. $\mathcal{S}_t = \{\langle t_i, p_i\rangle \in \mathcal{S}, t_i = t\}$. Let $X_t$ be a random variable which captures the distribution of the occurrences of $t$ ($f_t$). Since each $\langle t, p_i\rangle \in \mathcal{S}_t$ can be thought of as defining an (independent) Bernoulli random variable, we can write $\mathsf{E}[X_t] = f_t = \sum_{\langle t,p_i\rangle\in\mathcal{S}_t} p_i$. Moreover, each Bernoulli variable has variance equal to $p_i(1-p_i)$ and by the summation of variances $\mathsf{Var}[X_t] = \sum_i p_i(1-p_i)$. Now we have

$$\mathsf{E}[X_t^2] = \mathsf{Var}[X_t] + \mathsf{E}[X_t]^2 = \sum_{\langle t_i,p_i\rangle\in\mathcal{S}_t} p_i(1-p_i) + (\sum_{\langle t_i,p_i\rangle\in\mathcal{S}_t} p_i)^2$$

For a given $t$, this sum can be tracked in constant space: let $v_t = \sum_{\langle t_i,p_i\rangle\in\mathcal{S}_t} p_i$ and $w_t = \sum_{\langle t_i,p_i\rangle\in\mathcal{S}_t} p_i^2$. Then we need to compute $v_t^2 - w_t + v_t$. Each of $v_t$ and $w_t$ are easy to update as each tuple from $\mathcal{S}$, $\langle t, p_i\rangle$, is seen: we set $v_t \leftarrow v_t + p_i$ and $w_t \leftarrow w_t + p_i^2$. Finally observe that

$$\mathsf{E}_G[F_2(\mathcal{S})] = \sum_{t=1}^M \mathsf{E}[X_t^2] = \sum_{t=1}^M (v_t^2 + v_t - w_t),$$

showing that $\mathsf{E}_G[F_2(\mathcal{S})]$ can be found exactly in space $O(M)$.

In order to compute $\mathsf{E}_G[F_2(\mathcal{S})]$ in small space (sublinear in the number of domain size, $M$), we define the random variable $Y = \sum_{t=1}^M X_t^2$. By linearity of expectation,

$$\mathsf{E}_G[F_2(\mathcal{S})] = \mathsf{E}[Y] = \sum_{t=1}^M (\sum_{\langle t_i,p_i\rangle\in\mathcal{S}_t} p_i - p_i^2) + (\sum_{\langle t_i,p_i\rangle\in\mathcal{S}_t} p_i)^2)$$

The first term is precisely $\mathsf{Var}_G[F_1(\mathcal{S})]$, which we have already shown can be computed exactly in constant space by tracking the sum of all variances of each tuple, i.e. $\sum_{\langle t_i,p_i\rangle\in\mathcal{S}} p_i(1-p_i)$. The last term is a streaming second frequency moment computation applied to the stream of probabilities $\mathcal{S}$. We can treat $\mathcal{S}$ as a non-probabilistic stream which defines the vector $K_1$ such that

$$K_1[t] = v_t = \sum_{\langle t_i,p_i\rangle\in\mathcal{S}, t=t_i} p_i,$$

and write this term as $\|K_1\|_2^2$, the square of the $L_2$ norm of the vector $V$. Using the Alon *et al.* sketching technique, we can find $\|\hat{K}_1\|_2^2$ which is an $(\epsilon, \delta)$ estimator for $\|K_1\|_2$ in space $O(\frac{1}{\epsilon^2}\log\frac{1}{\delta})$ [2, 10]. Thus we define our estimator

$$\hat{F}_2(\mathcal{S}) = \mathsf{Var}_G[F_1(\mathcal{S})] + \|\hat{K}_1\|_2^2.$$

Since $\mathsf{Var}_G[F_1(\mathcal{S})] \ge 0$ provided all $0 \le p_i \le 1$, the sum of the first and third terms above is a non-negative quantity; hence, summing a $(1 \pm \epsilon)$ estimator for $\|K_1\|_2$ with exact values for $\mathsf{Var}_G[F_1(\mathcal{S})]$ yields an estimator for $\mathsf{E}_G[F_2(\mathcal{S})]$ that is within a $(1 \pm \epsilon)$ factor with probability at least $1 - \delta$. In other words, our probabilistic AMS technique (pAMS) is a guaranteed $(\epsilon, \delta)$ estimator for $\mathsf{E}_G[F_2(\mathcal{S})]$. Importantly, the space used is independent of $M$, the number of distinct items in the stream and $|\mathcal{S}| = N$, the length of the stream. ∎

EXAMPLE 5.2. *We again use the stream* $\mathcal{S} = (\langle x, \frac{1}{2}\rangle, \langle y, \frac{1}{4}\rangle, \langle y, \frac{1}{3}\rangle)$. *Here we can compute over all ground streams*

$$\mathsf{E}_G[F_2(\mathcal{S})] = 1(\tfrac{1}{4} + \tfrac{5}{24}) + 2\tfrac{5}{24} + 4\tfrac{1}{24} + 5\tfrac{1}{24} = \tfrac{5}{4}.$$

*By the above analysis, we have*

$$v_x = \tfrac{1}{2} \text{ and } w_x = \tfrac{1}{4}; v_y = \tfrac{7}{12} \text{ and } w_y = \tfrac{25}{144}.$$

*Thus, we can confirm that*

$$\mathsf{E}_G[F_2(\mathcal{S})] = (\tfrac{1}{2} + \tfrac{1}{4} - \tfrac{1}{4}) + (\tfrac{7}{12} + \tfrac{49}{144} - \tfrac{25}{144}) = \tfrac{5}{4}.$$

## 5.2 Variance of $F_2$ using pAMS

To compute $\mathsf{E}[F_2(\mathcal{S})]$, we made use of the facts that the expectation of the sum of independent random variables is equal to the sum of the expectations (linearity of expectation), and that the variance of the sum is equal to the sum of the variances (linearity of variance). These are two cases of more general properties of random variables based on the *cumulants* of the variables, denoted by $\kappa_j[X]$. The first cumulant of a random variable $X$, $\kappa_1[X]$, is just the mean of the distribution, $\mathsf{E}[X]$. The second cumulant is the variance: $\kappa_2[X] = \mathsf{Var}[X]$. Higher cumulants can be expressed in terms of the central moments of the variable: for example,

$$\kappa_3[X] = \mathsf{E}[(X - \mathsf{E}[X])^3] \text{ and } \kappa_4[X] = \mathsf{E}[(X - \mathsf{E}[X])^4] - 3\mathsf{Var}[X]^2.$$

The important property of cumulants that we will make extensive use of is that, for independent random variables $X$ and $Y$,

$$\forall j \in \mathbb{N}. \quad \kappa_j[X + Y] = \kappa_j[X] + \kappa_j[Y]$$

that is, cumulants generalize the linearity of expectation and variance to higher moments of independent variables.

THEOREM 5.3. *We can approximate* $\mathsf{Var}_G[F_2(\mathcal{S})]$ *with additive error* $\epsilon \mathsf{E}_G[F_2(\mathcal{S})]^{3/2}$ *in space* $O(\frac{1}{\epsilon^2} \log \frac{1}{\delta})$. ∎

*Proof:* By expanding and rearranging, we can write

$$\mathsf{E}[X^4] = \kappa_4[X] + 4\kappa_3[X]\kappa_1[X] + 3\kappa_2^2[X] + 6\kappa_2[X]\kappa_1^2[X] + \kappa_1^4[X]$$
$$\mathsf{E}[X^2]^2 = (\kappa_1^2[X] + \kappa_2[X])^2 = \kappa_1^4[X] + \kappa_2^2[X] + 2\kappa_1^2[X]\kappa_2[X]$$
$$\mathsf{Var}[X^2] = \mathsf{E}[X^4] - \mathsf{E}[X^2]^2$$
$$= \kappa_4[X] + 4\kappa_3[X]\kappa_1[X] + 2\kappa_2^2[X] + 4\kappa_2[X]\kappa_1^2[X]$$

Writing $B(p)$ for a Bernoulli random variable with parameter $p$, we have the following equalities:

$$\kappa_1[B(p)] = \mathsf{E}[B(p)] = p \qquad \kappa_2[B(p)] = \mathsf{Var}[B(p)] = p(1-p)$$
$$\kappa_3[B(p)] = (1-2p)p(1-p)\ \kappa_4[B(p)] = (1 - 6p + 6p^2)p(1-p)$$

Note that since $0 \le p \le 1$, we have

$$0 \le \kappa_2[B(p)] \le \kappa_1[B(p)] \le 1$$
$$-\kappa_2[B(p)] \le \kappa_3[B(p)] \le \kappa_2[B(p)]$$
$$-\kappa_2[B(p)]/2 \le \kappa_4[B(p)] = (1 - 6\kappa_2[B(p)])\kappa_2[B(p)] \le \kappa_2[B(p)]$$

Thus $\kappa_3[B(p)]$ and $\kappa_4[B(p)]$ may be negative, but for any random variable $\mathsf{Var}[X^2]$ must be non-negative (indeed, for a Bernoulli random variable, $B(p) = B(p)^2 = B(p)^4$ and so $\mathsf{Var}[B(p)^2] = \mathsf{Var}[B(p)] = \kappa_2[B(p)]$). By summation of cumulants, for an RV $X_t$ corresponding to the number of occurrences of $t$ in $\mathcal{S}$, we have

$$\kappa_k[X_t] = \textstyle\sum_{\langle t_i, p_i\rangle \in \mathcal{S}_t} \kappa_k[B(p_i)].$$

We will write vectors $K_j$ such that $K_j[t] = \kappa_j[X_t]$ (hence, $K_1$ is as before). By summation of variance, we can write

$$\mathsf{Var}_G[F_2(\mathcal{S})] = \sum_{t=1}^{M} K_4[t] + 4K_3[t]K_1[t] + 2K_2^2[t] + 4K_1^2[t]K_2[t]$$

The first term, which we denote $\kappa_4[\mathcal{S}]$, can be computed exactly in constant space, as it is a sum of a function of each $p_i$ in turn. The other three terms are more complex, but can be dealt with using techniques from computing over deterministic streams. The second and third terms are both the inner-product of vectors of dimension $M$, where each entry is the sum of values derived from individual tuples in $\mathcal{S}$. These can be approximated using the sketching technique of Alon *et al.* [1]. The fourth term can be thought of as a three-way join between relations encoded as vectors $x, y, z$, where it happens that $x = y$. This can be approximated using the technique of Dobra *et al.* [9]. Thus we can build an estimator $\widehat{\mathsf{Var}}_G[F_2(\mathcal{S})]$ by building four AMS sketches, one each for the vectors $K_3$ and $K_2$, and two for $K_1$ (two are needed to estimate the last term), plus the exact computation of $\|K_4\|_1$ in constant space. Our estimator is formed as:

$$\widehat{\mathsf{Var}} = \kappa_4[\mathcal{S}] + 4\widehat{K_3 \cdot K_1} + \|\hat{K}_2\|_2^2 + 4K_1 \cdot \widehat{K_1 \cdot K_2},$$

where $\widehat{K_3 \cdot K_1}$ is the (approximate) dot product of the vectors $K_3$ and $K_1$, and $K_1 \cdot \widehat{K_1 \cdot K_3}$ is the (approximate) three-way dot product of those vectors. With probability at least $1 - \delta$, the error $|\mathsf{Var}_G[F_2(\mathcal{S})] - \widehat{\mathsf{Var}}|$ is at most

$$\epsilon(4\|K_3\|_2\|K_1\|_2 + \|K_2\|_2^2 + 4\|K_1\|_2^2\|K_2\|_2).$$

By the above observations on the relative size of the cumulants from which the vectors are formed, we have

$$|\kappa_3[B(p)]| \le \kappa_2[B(p)] \le \kappa_1[B(p)].$$

Each variable $X_t$ is formed as the sum of Bernoulli random variables, giving $|\kappa_3[X_t]| \le \kappa_2[X_t] \le \kappa_1[X_t]$ and hence $\|K_3\|_2 \le \|K_2\|_2 \le \|K_1\|_2$. As a result, the error from the approximation is bounded by $9\|K_1\|_2^2\|K_2\|_2$. To write this in terms of the quantities we are estimating, we observe that $\|K_2\|_2 \le \|K_1\|_2$ for any stream, and note that $\mathsf{E}_G[F_2(\mathcal{S})] = (\|K_1\|_2^2 + \|K_2\|_1)$, so $\|K_1\|_2^3 \le \mathsf{E}_G[F_2(\mathcal{S})]^{3/2}$. We note that a similar line of reasoning using the cumulant representation can show that $\mathsf{Var}_G[F_2(\mathcal{S})] \le \mathsf{E}_G[F_2(\mathcal{S})]^{3/2}$, meaning that this bound is reasonable. We conclude that using our pAMS technique, we can build an estimator in constant space, independent of $N$ and $M$, so (after rescaling of $\epsilon$ by a constant $\le 9$):

$$\Pr[|\widehat{\mathsf{Var}}_G[F_2(\mathcal{S})] - \mathsf{Var}_G[F_2(\mathcal{S})]| \ge \epsilon \mathsf{E}_G[F_2(\mathcal{S})]^{3/2}] \le 1 - \delta.$$ ∎

EXAMPLE 5.4. *We study computing* $\mathsf{Var}[F_2(S)]$ *for* $\mathcal{S} = (\langle x, \frac{1}{2}\rangle, \langle y, \frac{1}{4}\rangle, \langle y, \frac{1}{3}\rangle)$, *to demonstrate the reduction to computation of cumulants. First, over all ground streams we compute*

$$\mathsf{E}_G[F_2(S)^2] = 1(\tfrac{6}{24} + \tfrac{5}{24}) + 4\tfrac{5}{24} + 25\tfrac{1}{24} + 16\tfrac{1}{24} = 3,$$

*so* $\mathsf{Var}_G[F_2(\mathcal{S})] = \mathsf{E}_G[F_2(\mathcal{S})^2] - \mathsf{E}_G[F_2(\mathcal{S})]^2 = 3 - \frac{25}{16} = \frac{23}{16}$.

*For the cumulant based approach, we have the vectors*

$$K_1 = [\tfrac{1}{2}, \tfrac{7}{12}],\ K_2 = [\tfrac{1}{4}, \tfrac{59}{144}],\ K_3 = [0, \tfrac{145}{864}],\ K_4 = [\tfrac{-1}{8}, \tfrac{-337}{3456}].$$

*Thus we confirm that, as claimed,* $\mathsf{Var}_G[F_2(S)] = \frac{23}{16} =$

$$(\tfrac{-1}{8} + \tfrac{-337}{3456}) + 4(\tfrac{7}{12} \cdot \tfrac{145}{864}) + 4(\tfrac{1}{4} \cdot \tfrac{1}{2}^2 + \tfrac{59}{144} \cdot \tfrac{7}{12}^2) + 2(\tfrac{1}{4}^2 + \tfrac{59}{144}^2).$$

## 5.3 Expected Join Size with pAMS

Consider two streams $\mathcal{S}_1$ and $\mathcal{S}_2$, and we wish to evaluate $\mathsf{E}_G[|\mathcal{S}_1 \bowtie \mathcal{S}_2|]$, i.e. the expected join size between all possible grounded streams of $\mathcal{S}_1$ and $\mathcal{S}_2$. This initially seems even more challenging, since we are now reasoning over a space of $2^{|\mathcal{S}_1|} \times 2^{|\mathcal{S}_2|}$ possible worlds. In fact, we can again use pAMS sketches.

THEOREM 5.5. *In one pass we can compute an estimate* $J$ *of* $\mathsf{E}_G[|\mathcal{S}_1 \bowtie \mathcal{S}_2|]$ *such that* $|J - \mathsf{E}_G[|\mathcal{S}_1 \bowtie \mathcal{S}_2|]| \le \epsilon \|\mathcal{S}_1\|_2 \|\mathcal{S}_2\|_2$ *with probability at least* $1 - \delta$ *in space* $O(\frac{1}{\epsilon^2} \log 1/\delta)$. ∎

*Proof:* Let $K_1(\mathcal{S}_1)$ and $K_1(\mathcal{S}_2)$ be the vectors of first cumulants as defined above. Expanding out the definitions and using the independence of $\mathcal{S}_1$ and $\mathcal{S}_2$ we find $\mathsf{E}_G[\mathcal{S}_1 \bowtie \mathcal{S}_2] = K_1(\mathcal{S}_1) \cdot K_1(\mathcal{S}_2)$, the dot product of the two vectors. Using sketches again we can compute the estimate $J = \widehat{K_1(\mathcal{S}_1) \cdot K_1(\mathcal{S}_2)}$ and, by properties of the sketches, the error is bounded by

$$|J - \mathsf{E}_G[|\mathcal{S}_1 \bowtie \mathcal{S}_2|]| \le \epsilon \|\mathcal{S}_1\|_2 \|\mathcal{S}_2\|_2$$

with probability at least $1-\delta$ using a summary for each stream of size $O(\frac{1}{\epsilon}\log 1/\delta)$, independent of $M$ and $N$. This summary can be maintained incrementally as each new $\langle t_i, p_i\rangle$ is read. ∎

Note that in the case that each $p_i$ is 1 (a deterministic stream), this is identical to the regular join size estimation using sketches, as one would hope.

EXAMPLE 5.6. *Consider again our example stream,* $\mathcal{S} = (\langle x, \frac{1}{2}\rangle, \langle y, \frac{1}{4}\rangle, \langle y, \frac{1}{3}\rangle)$*, and let* $\mathcal{S}_1 = \mathcal{S}_2 = \mathcal{S}$ *be two* independent *streams which happen to have the same distribution. Even with this tiny example, it is somewhat laborious to evaluate all 25 possible combinations of grounded streams in order to compute* $\mathsf{E}_G[|\mathcal{S}_1 \bowtie \mathcal{S}_2|] = \frac{85}{144}$*. In comparison, it is straightforward to find*

$$p_{x,1} = p_{x,2} = \tfrac{1}{2}, \qquad p_{y,1} = p_{y,2} = \tfrac{7}{12},$$

*and so compute* $\mathsf{E}_G[|\mathcal{S}_1 \bowtie \mathcal{S}_2|] = \frac{1}{2}^2 + \frac{7}{12}^2 = \frac{85}{144}$.

Note that this is a different answer compared to $\mathsf{E}_G[F_2(\mathcal{S})]$ on the same stream, whereas for ground stream computations, the size of the join between a stream and itself is equal to the second frequency moment of the stream. There is no contradiction here: note that $\mathcal{S}_1$ and $\mathcal{S}_2$ are two *independent* streams. The join of two independent streams with the same distribution is different from the self-join of a single probabilistic stream (i.e. $\mathsf{E}_G[F_2(\mathcal{S})]$), since the former must consider all possible combinations of ground streams. In fact, we can always be sure that $\mathsf{E}_G[|\mathcal{S} \bowtie \mathcal{S}|] \leq \mathsf{E}_G[F_2(\mathcal{S})]$, since we can write $\mathsf{E}_G[|\mathcal{S} \bowtie \mathcal{S}|] = \|\mathcal{S}\|_2^2$, whereas $\mathsf{E}_G[F_2(\mathcal{S})] = \|\mathcal{S}\|_2^2 + \mathsf{Var}_G[F_1(\mathcal{S})]$. Equality occurs when all probabilities are 0 or 1, i.e. deterministic streams. This highlights an important example where our intuitions from dealing with deterministic streams do not directly carry over to the world(s) of probabilistic streams.

# 6. HIGHER MOMENTS ESTIMATION

We extend our approach to higher frequency moments ($F_k > 2$) and higher central moments ($\mathsf{E}_G[(X - \mathsf{E}_G[X])^k]$, $k > 2$).

## 6.1 Higher Central Moments

The central moments of a distribution are defined as

$$\mathsf{C}_k[X] = \mathsf{E}_G[(X - \mathsf{E}_G[X])^k]$$

for $k \in \mathbb{N}$. It follows that $\mathsf{C}_1[X] = 0$, and $\mathsf{C}_2[X] = \mathsf{E}[(X - \mathsf{E}[X])^2] = \mathsf{Var}[X]$. The higher central moments further define the distribution. By analogy to our earlier definitions, we can define for probabilistic streams and aggregate $F$:

$$\begin{aligned}\mathsf{C}_{k,G}[F] =& \mathsf{E}_G[(F(\mathcal{S}) - \mathsf{E}_G[F(\mathcal{S})])^k] \\ =& \sum_{G \in \mathrm{grnd}(\mathcal{S})} \Pr[G](F(G) - \mathsf{E}_G[F(\mathcal{S})])^k\end{aligned}$$

They can also be written in terms of cumulants. For example, $\mathsf{C}_3[X] = \kappa_3[X]$, and thus we can compute $\mathsf{C}_{3,G}[F_1(\mathcal{S})]$ in constant space. Continuing:

$$\begin{aligned}\mathsf{C}_4[X] =& \kappa_4[X] + 3\kappa_2[X]^2 \\ \mathsf{C}_5[X] =& \kappa_5[X] + 10\kappa_3[X]\kappa_2[X] \\ \mathsf{C}_6[X] =& \kappa_6[X] + 15\kappa_4[X]\kappa_2[X] + 15\kappa_2^3[X]\end{aligned}$$

Thus $\mathsf{C}_{4,G}[F_1(\mathcal{S})]$, $\mathsf{C}_{5,G}[F_1(\mathcal{S})]$ and $\mathsf{C}_{6,G}[F_1(\mathcal{S})]$ can also be computed exactly in constant space: we can compute each $\kappa_{k,G}[F_1(\mathcal{S})]$ exactly in constant space for $k \leq 6$, by computing $\kappa_k[B(p_i)]$ of each individual Bernoulli random variable defined by a $\langle t_i, p_i\rangle$ tuple, and using the summability of cumulants to get the correct result. This method works for all higher cumulants and central moments $\kappa_k[F_1(\mathcal{S})]$ and $\mathsf{C}_{k,G}[F_1(\mathcal{S})]$, thus we can completely characterize the $k^{th}$ central moments in $O(k)$ space.

EXAMPLE 6.1. *Consider computing* $\mathsf{C}_{4,G}[F_1(\mathcal{S})]$*, where* $\mathcal{S} = (\langle x, \frac{1}{2}\rangle, \langle y, \frac{1}{4}\rangle, \langle y, \frac{1}{3}\rangle)$*, as usual. We find* $\mathsf{E}_G[F_1(\mathcal{S})] = \frac{13}{12}$*, and thus* $\mathsf{E}_G[(F_1(\mathcal{S}) - \mathsf{E}_G[F_1(\mathcal{S})])^4] =$

$$\tfrac{-13}{12}^4\tfrac{1}{4} + \tfrac{-1}{12}^4\tfrac{11}{24} + \tfrac{11}{12}^4\tfrac{1}{4} + \tfrac{23}{24}^4\tfrac{1}{24} \approx 1.0832.$$

*We compute* $\kappa_2[F_1(\mathcal{S})] = \mathsf{Var}_G[F_1(\mathcal{S})] = \frac{95}{144}$*, and* $\kappa_4[F_1(\mathcal{S})] = \frac{1}{4}(1-\frac{3}{2}) + \frac{2}{9}(1-\frac{4}{3}) + \frac{3}{16}(1-\frac{9}{8}) = \frac{-769}{3456}$*. Thus using the cumulants approach we find*

$$\kappa_4[F_1(\mathcal{S})] + 3\kappa_2^2[F_1(\mathcal{S})] = \mathsf{C}_{4,G}[F_1(\mathcal{S})](\approx 1.0832). \quad \square$$

**Computing $\mathsf{C}_{3,G}[F_0(\mathcal{S})]$.** Now that we have completely characterized all central moments based on $F_1$ exactly, we turn our attention to central moments based on the $F_0$ aggregate, In Section 4.1 we showed how to estimate $\mathsf{C}_{2,G}[F_0(\mathcal{S})]$ (that is, $\mathsf{Var}_G[F_0(\mathcal{S})]$) by reducing to the estimation of $\mathsf{E}_G[F_0(\mathcal{S}_2])$ of a derived stream $\mathcal{S}_2$. The same approach can work for $\mathsf{C}_{3,G}[F_0(\mathcal{S})]$.

THEOREM 6.2. *Given an* $(\epsilon, \delta)$ *estimator for* $\mathsf{E}_G[F_0(\mathcal{S})]$*, the quantity* $\mathsf{C}_{3,G}[F_0(\mathcal{S})]$ *can be estimated with error at most* $13\epsilon\mathsf{E}_G[F_0(\mathcal{S})]$ *with probability at least* $1 - 3\delta$*.* ∎

*Proof:* As before, define $\mathcal{S}_2 = \langle t_i, 2p_i - p_i^2\rangle$ and additionally define $\mathcal{S}_3 = \langle t_i, 3p_i - 3p_i^2 + p_i^3\rangle$. Observe that both new streams generate new probabilities which are well behaved, i.e., lie in the range $[0 \ldots 1]$. We claim that

$$\mathsf{E}_G[F_0(\mathcal{S})] - 3\mathsf{E}_G[(F_0(\mathcal{S}_2)] + 2\mathsf{E}_G[F_0(\mathcal{S}_3)] = \mathsf{C}_{3,G}[F_0(\mathcal{S})].$$

Defining $\overline{p}_t = 1 - p_t$, we have $\mathsf{E}_G[F_0(\mathcal{S}_3)] = \sum_{t=1}^M 1 - \overline{p}_t^3$ And so, simplifying to the definition of $\kappa_3[X_t]$:
$\mathsf{E}_G[F_0(\mathcal{S})] - 3\mathsf{E}_G[F_0(\mathcal{S}_2)] + 2\mathsf{E}_G[F_0(\mathcal{S}_3)] = \sum_{t=1}^M p_t - 3p_t^2 + 2p_t^3$ which is $\mathsf{C}_{3,G}[F_0(\mathcal{S})]$. From Lemma 4.5, we have $\mathsf{E}_G[F_0(\mathcal{S}_2)] \leq 2\mathsf{E}_G[F_0(\mathcal{S})]$. Similarly, we can observe that

$$3p_t - 3p_t^2 + p_t^3 \leq 3p_t - 2p_t^2 - p_t^2(1-p_t) \leq 3p_t.$$

So $\mathsf{E}_G[F_0(\mathcal{S}_3)] \leq 3\mathsf{E}_G[F_0(\mathcal{S})]$. Consequently, the net error in the estimate of $\mathsf{C}_{3,G}[F_0(\mathcal{S})]$ is bounded by $(1+2\cdot3+3\cdot2)\mathsf{E}_G[F_0(\mathcal{S})]$, with probability at least $1-3\delta$ by the union bound. ∎

We note that the same idea of keeping a single pFM sketch and deriving a sketch $\mathsf{pFM}_2$ of $\mathcal{S}_2$ and $\mathsf{pFM}_3$ of $\mathcal{S}_3$ works here, meaning that the estimation can be made using a single sketch.

Similar techniques can be adapted for higher central frequency moments, $\mathsf{C}_{k,G}[F_0(\mathcal{S})]$, although they become more involved, as the terms become harder to approximate, and the relevant constants increase, so we omit the lengthy details. Likewise, we can build estimators for $\mathsf{C}_{k,G}[F_2(\mathcal{S})]$ by generalizing the technique we used for $\mathsf{Var}_G[F_2(\mathcal{S})]$, but these also become quite involved and are of lesser value compared to $\mathsf{E}_G[F_2(\mathcal{S})]$ and $\mathsf{Var}_G[F_2(\mathcal{S})]$, so we do not discuss them further here.

## 6.2 Higher Frequency Moments

A similar technique cam be applied to approximate the expected value of higher frequency moments. We focus on $\mathsf{E}_G[F_3(\mathcal{S})]$, the expected value of the third frequency moment, defined as

$$\mathsf{E}_G[F_3(\mathcal{S})] = \textstyle\sum_{G \in \mathrm{grnd}(\mathcal{S})} \Pr[G] \cdot F_3(G).$$

We make use of properties of the *third cumulant*, $\kappa_3[X]$, which can also be written as

$$\kappa_3[X] = \mathsf{E}[X^3] - 3\kappa_1[X]\kappa_2[X] - \kappa_1^3[X].$$

Thus, to find $\mathsf{E}_G[F_3(\mathcal{S})]$ we compute

$$\mathsf{E}_G[F_3(\mathcal{S})] = \textstyle\sum_{t=1}^M \kappa_3[X_t] + 3\kappa_1[X_t]\kappa_2[X_t] + \kappa_1^3[X_t]$$

We address each of these terms in turn. We already know that we can compute $\kappa_3[\mathcal{S}]$, the third cumulant of the whole stream, exactly with a single variable. The second term is an inner product

between the vectors $K_2$ and $K_1$ that were defined in Section 5.2 on estimating $\mathsf{Var}_G[F_2(\mathcal{S})]$. Using the AMS sketching technique, this quantity can be estimated up to error $\epsilon\|K_1\|_2\|K_2\|_2$ in space $O(\frac{1}{\epsilon^2}\log\frac{1}{\delta})$. Lastly, we need to estimate

$$\sum_{t=1}^{M}\kappa_1[X_t]^3 = \sum_{t=1}^{M}(\sum_{\langle t_i,p_i\rangle\in\mathcal{S}_t} p_i)^3,$$

which is a (generalized version of a) third moment computation of the stream of $p_i$'s. In order to approximate this, we define a variation of the generic AMS estimator for the more general problem of estimating the $F_k(\mathcal{S})$ for any $k$ [2], and prove that it gives the necessary accuracy[2]. Note that the original estimator is only defined for "unary" streams where each arrival of an item counts for an increment of exactly one for the frequency of that item, and so requires some careful analysis and proof.

**Estimator for $F_k$ on Streams of Fractional Values.** We define the generalized $k^{th}$ frequency moment as:

$$F_k(\mathcal{S}) = \sum_{t=1}^{M}(\sum_{\langle t_i,p_i\rangle\in\mathcal{S}_t} p_i)^k).$$

Note that this is quite distinct from the definition of $\mathsf{E}_G[F_k]$ above: this question does not respect the semantics of $\mathcal{S}$ being a probabilistic stream, instead treating it as a deterministic stream of fractional non-negative updates. We can think of it as $\|V\|_k^k$, in terms of a vector norm on a vector $V$ defined by the stream. We define an estimator as follows. Let $P_j = \sum_{\langle t_i,p_i\rangle\in\mathcal{S},i<j} p_j$, i.e. $P_j$ is $\mathsf{E}_G[F_1(\mathcal{S})]$ for the stream *seen so far*. From the stream, randomly sample the $t$-th tuple, $\langle t_j, p_j\rangle$, with probability $p_j/P_j$. This can be done using a weighted version of standard reservoir sampling: track the value of $P_j = \sum_{i=1}^{j} p_i$, and sample the next tuple $j$ (replacing the currently sampled item) with probability $p_j/P_j$. It follows inductively that the probability of an item surviving this process is exactly $p_j/P_j$, as required. Then compute $p_{t,j} = \sum_{\langle t_i,p_i\rangle,i>j,t_i=t} p_i$, the sum of probabilities of all tuples containing $t$ which occur after the sampled $j-$th tuple. We output the estimator

$$E_j = ((p_j + p_{t,j})^k - p_{t,j}^k)P_j/p_j.$$

THEOREM 6.3. *Taking the mean of $O(\frac{k}{\epsilon^2}M^{1-1/k})$ copies of the estimator $E_j$, and the median of $O(\log 1/\delta)$ means gives an $(\epsilon,\delta)$ approximation $\|\hat{V}\|_k^k$ for $F_k(\mathcal{S})$ in one pass.* ∎

Thus we can estimate the $\sum_{t=1}^{M}\kappa_1^3[X_t] = \|K_1\|_3^3$ term in the expression for $\mathsf{E}_G[F_3(\mathcal{S})]$ in space sublinear in the support size of the stream $M$, and independent of the stream length $N$. Putting these together, we form our estimator for $\mathsf{E}_G[F_3(\mathcal{S})]$ as $\hat{F}_3 = 3\widehat{K_1\cdot K_2} + \|\hat{K}_1\|_3^3 + \kappa_3[\mathcal{S}]$. With probability at least $1-\delta$,

$$|\hat{F}_3 - \mathsf{E}_G[F_3(\mathcal{S})]| \le \epsilon(3\|K_1\|_2\|K_2\|_2 + \|K_1\|_3^3).$$

Note that the $\kappa_3[\mathcal{S}]$ term may be negative, however one can show $\mathsf{E}_G[F_3(\mathcal{S})] \ge \|K_1\|_3^3$, which is non-negative. If we assume that the average value of $K_1$ is no less than 1 (i.e., $\|K_1\|_1 \ge M$), and recalling that $\|K_2\|_2 \le \|K_1\|_2$ (see proof of Theorem 5.3), we can prove $\|K_1\|_2^2 \le \|K_1\|_3^3$, giving a bound on the error as $4\epsilon\|K_1\|_3^3$. Thus, after rescaling $\epsilon$ by a constant at most 4, we have a relative error approximation in sublinear space, and so we conclude,

THEOREM 6.4. *In one pass, we can find an $(\epsilon,\delta)$ estimator for $\mathsf{E}_G[F_3(\mathcal{S})]$ in space $O(\frac{1}{\epsilon^2}M^{2/3}\log 1/\delta)$, if $\mathsf{E}_G[F_1(\mathcal{S})] \ge M$.* ∎

# 7. EXPERIMENTAL RESULTS

We implemented our algorithms for the main frequency moments $F_0$, $F_1$ and $F_2$ in C: probabilistic FM (pFM), probabilistic AMS (pAMS), and the universal sampling algorithm using FM and AMS estimators, so our results are fairly compared. We exhaustively computed the exact answers to our aggregates, in order to compare them to the approximate values we obtain from our algorithms. Our experiments were performed on standard desktop class machines. We found our algorithms to be very efficient, and we report timings as a general guide to relative performance, since the exact behavior will vary depending on optimizations, cache policy and so on.

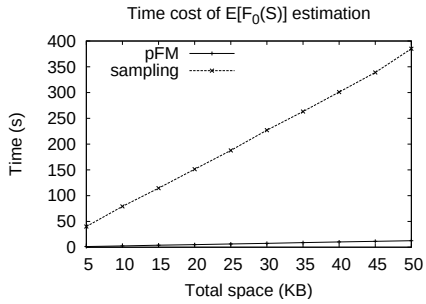


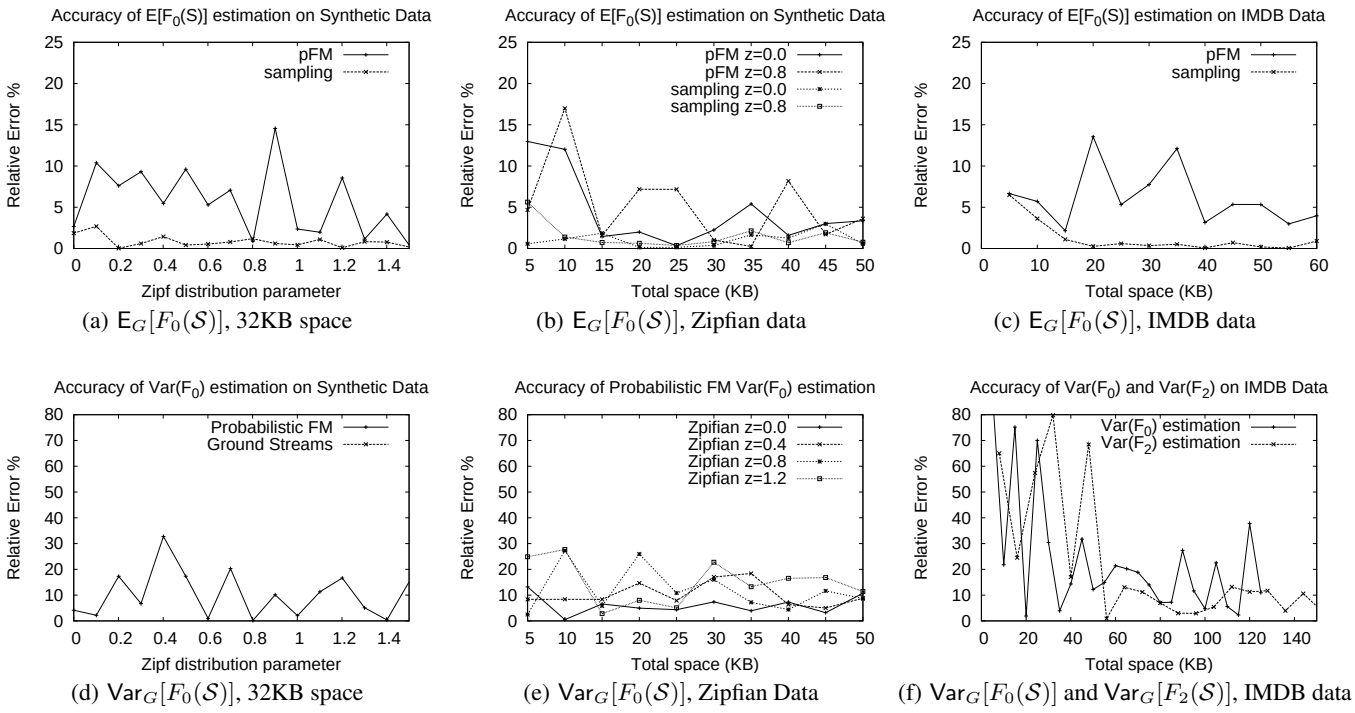

**Figure 3: Time for $\mathsf{E}_G[F_0(\mathcal{S})]$ estimation**

**Data Sets.** We considered a variety of real and synthetic data. For synthetic data, we created each tuple $\langle t_i, p_i\rangle$ independently by drawing each $t_i$ from a Zipfian distribution, and each $p_i$ uniform from the range $[0,1]$. This captures many settings where data is drawn from distributions with skew, and with a variety of uncertainty values. We vary the skewness parameter of the Zipfian distribution from $z = 0.0$ (uniform) to $z = 2.0$ (highly skewed). In each experiment, we generated $10^6$ tuples from this distribution. We also used a real data set from the MYSTIQ project[3] which includes approximately 250,000 probabilistic tuples. Each tuple links a film in the IMDB database to a product in Amazon.com's inventory, and includes a "probability of match" value. So $\mathsf{E}_G[F_0(\mathcal{S})]$ is the expected number of (distinct) matched titles matched, and $\mathsf{E}_G[F_2(\mathcal{S})]$ is the expected self-join size.

**Experiments on $F_0$ Estimation.** Our experiments on $F_0$ estimation are shown in Figure 2. We see that, given a fixed amount of space, the universal sampling algorithm consistently obtains an approximation of $\mathsf{E}_G[F_0(\mathcal{S})]$ that is within a few percentage points, while the Probabilistic FM approach is typically between 5% to 10% relative error from uniform to skewed data (Figure 2(a)). Partly this is due to the space efficiency of the algorithms: the universal algorithm can use FM sketches which are based on bitmaps, whereas for each bit in a regular FM sketch the pFM structure uses a 64 bit integer; thus, it uses 64 times as many structures in the same space. The trend is for the accuracy to improve as more space becomes available (Figure 2(b)). Because the FM algorithm is itself randomized, accuracy does not improve uniformly with more space, but rather the trend is for the variability to decrease as more space is used, as seen more clearly in Figure 2(c). The conclusion appears to be that for the task of estimating $\mathsf{E}_G[F_0(\mathcal{S})]$, the universal algorithm is the most accurate within a given space bound. However, there is a price to pay in terms of the time cost. Since it has 64 times as many data structures, the time to update is correspondingly slower. Figure 3 shows that processing time increases linearly with space for both algorithms, as predicted by our analysis. But the universal algorithm is about thirty times slower to process a million tuples in our implementation (it manipulates bit vectors instead of floating-point numbers, so is not fully 64 times slower).

[2]We comment that the algorithm of Ganguly *et al* [12] can also be used to obtain an improved dependency on $M$ but a higher dependency on $\epsilon$.

[3]See `http://www.cs.washington.edu/homes/suciu/project-mystiq.html`; we thank the authors for sharing their data.




(a) $\mathsf{E}_G[F_0(\mathcal{S})]$, 32KB space (b) $\mathsf{E}_G[F_0(\mathcal{S})]$, Zipfian data (c) $\mathsf{E}_G[F_0(\mathcal{S})]$, IMDB data

(d) $\mathsf{Var}_G[F_0(\mathcal{S})]$, 32KB space (e) $\mathsf{Var}_G[F_0(\mathcal{S})]$, Zipfian Data (f) $\mathsf{Var}_G[F_0(\mathcal{S})]$ and $\mathsf{Var}_G[F_2(\mathcal{S})]$, IMDB data

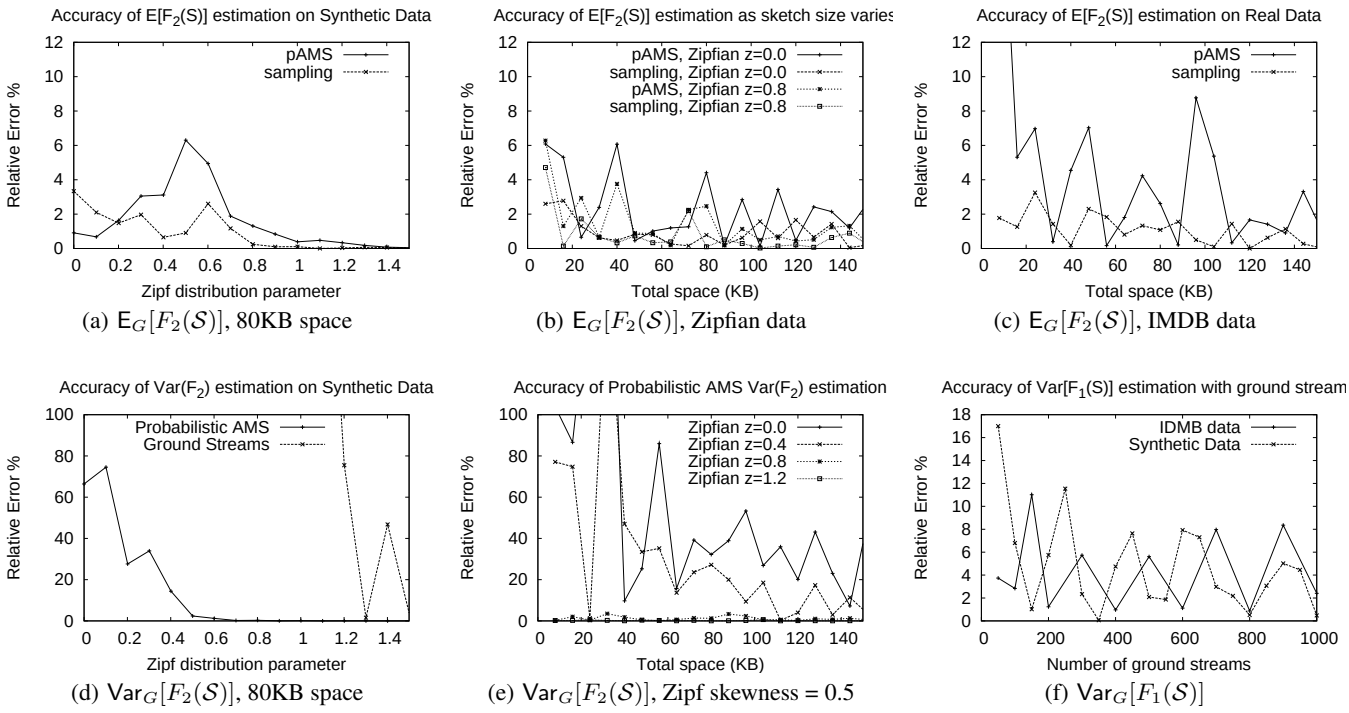
**Figure 2: Results on $\mathsf{E}_G[F_0(\mathcal{S})]$ and $\mathsf{Var}_G[F_0(\mathcal{S})]$ estimation on real and synthetic data**

When it comes to computing $\mathsf{Var}_G[F_0(\mathcal{S})]$, the tables are turned, and quite dramatically so. The probabilistic FM algorithm consistently achieves a relative accuracy within 10-20% with a moderate amount of memory (recall, its guarantee is in terms of $\mathsf{E}_G[F_0(\mathcal{S})]$, not $\mathsf{Var}_G[F_0(\mathcal{S})]$). Meanwhile, computing the variance of the samples found by the universal algorithm yields estimates that are many orders of magnitude adrift. The reason for this is given in Section 3.1: By studying the output of the algorithms, we verified that it obtains a very accurate estimate of $\mathsf{E}_G[F_0(\mathcal{S})]^2$ and $\mathsf{E}_G[F_0(\mathcal{S})]^2$. But these terms are both very large compared to $\mathsf{Var}_G[F_0(\mathcal{S})]$, so the final error becomes huge, in relative terms. Meanwhile, the probabilistic FM algorithm, which gives a guarantee on its error proportional to $\mathsf{E}_G[F_0(\mathcal{S})]$ instead of $\mathsf{E}_G[F_0(\mathcal{S})]^2$, can give much better estimates. This is true across a broad range of synthetic data (Figures 2(d) and 2(e)), and on real data (Figure 2(f)), with a trend for improving accuracy as space increases.

**Experiments on $F_2$ Estimation.** Our experimental results on $F_2$ estimation are shown in Figure 4. Again, the universal algorithm does well with an average of 2–3% error for estimating $\mathsf{E}_G[F_2(\mathcal{S})]$ on all data types, and improving accuracy as the given space increases (Figures 4(b) and 4(c)). Data with moderate skew seems to be the most challenging for the probabilistic AMS algorithm, which obtained accuracy of better than 6% throughout. Timing results are quite comparable: our implementation of pAMS used 64 bit floating point values where the universal algorithm uses the same structure but with 32 bit integers. So the universal algorithm did twice as many hashing operations, but then uses integer arithmetic to update counts. On our set up, the time cost was about the same for both algorithms: approximately 1 second for our algorithms to completely process a data set of $10^6$ items for $\mathsf{E}_G[F_2(\mathcal{S})]$ estimation, independent of the total summary size. This increases to 4 seconds for $\mathsf{Var}_G[F_2(\mathcal{S})]$, since our algorithm requires four sketches: one for each of the vectors $K_3$ and $K_2$, and two for $K_1$ (to perform the estimation of $K_1 \cdot K_1 \cdot K_2$). Thus either algorithm scales well to large quantities of data and restricted resources.

Again for $\mathsf{Var}_G[F_2(\mathcal{S})]$ estimation, we see the limitation of the universal algorithm. Only when the data set is extremely skewed ($z > 1$) does it have a chance of making any reasonable estimate. This is for the same reason as in the $F_0$ case: the error in its estimate is proportional to $\mathsf{E}_G[F_2(\mathcal{S})]^2$, which is very large in comparison to $\mathsf{Var}_G[F_2(\mathcal{S})]$. Our analysis shows that the error in pAMS estimation depends in the worst case on $\mathsf{E}_G[F_2(\mathcal{S})]^{3/2}$, which is enough of an edge to make its estimates quite tolerably accurate. The hard case for both algorithms is on more uniform data. Figure 4(e) shows that increasing the space for probabilistic FM improves the accuracy for the 'hard' data, while the error on more skewed data is close to zero throughout. The same conclusion is shown on the IMDB data in Figure 2(f). Likewise, increasing the amount of space available did tend to improve the accuracy of the universal algorithm, but not to sufficiently reliable levels even for highly skewed data. Interestingly, increasing the number of ground streams used to make the estimate did not seem to reliably improve the quality, and in fact tended to reduce the accuracy.

Lastly, we looked at estimating $\mathsf{Var}_G[F_1(\mathcal{S})]$ using the universal algorithm. Note that we can compute this exactly in constant space using our analysis from Section 3.2. However, this was a useful example to study since there is no error introduced in computing $F_1$ of the ground stream: it is precisely the number of items observed in that stream (consequently, the skewness of the synthetic Zipfian distribution is irrelevant for this aggregate). Figure 4(f) shows that increasing the number of ground streams does seem to improve the accuracy, but somewhat slowly and unreliably. Even with 1000 ground streams, the expected accuracy is above 5% relative error. For other aggregates, drawing even a hundred ground streams is not always be feasible, since this entails storing 100 separate AMS or FM summary structures, each of which is typically 10s of KBs




(a) $\mathsf{E}_G[F_2(\mathcal{S})]$, 80KB space (b) $\mathsf{E}_G[F_2(\mathcal{S})]$, Zipfian data (c) $\mathsf{E}_G[F_2(\mathcal{S})]$, IMDB data

(d) $\mathsf{Var}_G[F_2(\mathcal{S})]$, 80KB space (e) $\mathsf{Var}_G[F_2(\mathcal{S})]$, Zipf skewness = 0.5 (f) $\mathsf{Var}_G[F_1(\mathcal{S})]$

**Figure 4: Results on $\mathsf{E}_G[F_2(\mathcal{S})]$, $\mathsf{Var}_G[F_2(\mathcal{S})]$ and $\mathsf{Var}_G[F_1(\mathcal{S})]$ estimation**

or more. Moreover, the throughput decreases by a factor of 100s compared to the probabilistic sketching methods we propose here.

**Experimental Conclusions.** From our experiments, we conclude that, given a fixed amount of space, the universal sampling algorithm obtains high accuracy for computing the first central moment, such as $\mathsf{E}_G[F_0(\mathcal{S})]$ or $\mathsf{E}_G[F_2(\mathcal{S})]$. But, some cases such as $\mathsf{E}_G[F_0(\mathcal{S})]$ it can be many times slower than the probabilistic equivalent, which is somewhat less accurate. For the second central moment, $\mathsf{Var}_G[F_0(\mathcal{S})]$ and $\mathsf{Var}_G[F_2(\mathcal{S})]$, we can do dramatically better using methods tailored to the aggregate. This seems the only way to obtain efficient estimates of these quantities, since the amount of space to estimate them well using the universal algorithm is greater than the space needed to run the exact algorithm.

# 8. REFERENCES


[1] N. Alon, P. Gibbons, Y. Matias, and M. Szegedy. Tracking join and self-join sizes in limited storage. In *ACM PODS*, 1999.

[2] N. Alon, Y. Matias, and M. Szegedy. The space complexity of approximating the frequency moments. In *ACM STOC*, 1996.

[3] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom. Models and issues in data stream systems. In *ACM PODS*, 2002.

[4] Z. Bar-Yossef, T.S. Jayram, R. Kumar, D. Sivakumar, and L. Trevisian. Counting distinct elements in a data stream. In *RANDOM*, 2002.

[5] O. Benjelloun, A. Das Sarma, C. Halevy, and J. Widom. Uldbs: Databases with uncertainty and lineage. In *VLDB*, 2006.

[6] G. Cormode, F. Korn, S. Muthukrishnan, and D. Srivastava. Space- and time-efficient deterministic algorithms for biased quantiles over data streams. In *PODS*, 2006.

[7] G. Cormode and S. Muthukrishnan. Improved data stream summary: The count-min sketch and its applications. *J. Alg.*, 55(1), 2005.

[8] N. Dalvi and D. Suciu. Efficient query evaluation on probabilistic databases. In *VLDB*, 2004.

[9] A. Dobra, M. Garofalakis, J. E. Gehrke, and R. Rastogi. Processing complex aggregate queries over data streams. In *SIGMOD*, 2002.

[10] J. Feigenbaum, S. Kannan, M. Strauss, and M. Viswanathan. An approximate $L_1$-difference algorithm for massive data streams. In *IEEE FOCS*, 1999.

[11] P. Flajolet and G. N. Martin. Probabilistic counting algorithms for database applications. *J. Computer and System Sciences*, 31, 1985.

[12] L. Bhuvanagiri, S. Ganguly, D. Kesh, and C. Saha. Simpler algorithm for estimating frequency moments of data streams. In *SODA*, 2006

[13] M. Garofalakis, J. Gehrke, and R. Rastogi. Querying and mining data streams: You only get one look. In *SIGMOD*, 2002.

[14] A. Gilbert, Y. Kotidis, S. Muthukrishnan, and M. Strauss. Surfing wavelets on streams: One-pass summaries for approximate aggregate queries. In *VLDB*, 2001.

[15] M. Greenwald and S. Khanna. Space-efficient online computation of quantile summaries. In *SIGMOD*, 2001.

[16] T.S. Jayram, S. Kale, and E. Vee. Efficient aggregation algorithms for probabilistic data. In *SODA*, 2007.

[17] T.S. Jayram, R. Krshnamurthy, S. Raghavan, S. Vaithyanathan, and H. Zhu. Avatar information extraction system. *IEEE Data Eng. Bulletin*, 29(1), 2006.

[18] T.S. Jayram, A. McGregor, S. Muthukrishnan, E. Vee. Estimating Statistical Aggregates on Probabilistic Data Streams. In *PODS*, 2007.

[19] N. Khoussainova, M. Balazinska, and D. Suciu. Towards correcting input data errors probabilistically using integrity constraints. In *MobiDE*, 2006.

[20] G.S. Manku and R. Motwani. Approximate frequency counts over data streams. In *VLDB*, 2002.

[21] A. Metwally, D. Agrawal, and A. El Abbadi. Efficient computation of frequent and top-k elements in data streams. In *ICDT*, 2005.

[22] J. Misra and D. Gries. Finding repeated elements. *Science of Comp. Programming*, 2, 1982.

[23] A. Das Sarma, O. Benjelloun, A. Halevy, and J. Widom. Working models for uncertain data. In *IEEE ICDE*, 2006.

[24] N. Shrivastava, C. Buragohain, D. Agrawal, and S. Suri. Medians and beyond: New aggregation techniques for sensor networks. In *SenSys*, 2004.