Statistical Analysis of Sketch Estimators*

Florin Rusu
University of Florida
frusu@cise.ufl.edu

Alin Dobra
University of Florida
adobra@cise.ufl.edu

ABSTRACT

Sketching techniques can provide approximate answers to aggregate queries either for data-streaming or distributed computation. Small space summaries that have linearity properties are required for both types of applications. The prevalent method for analyzing sketches uses moment analysis and distribution independent bounds based on moments. This method produces clean, easy to interpret, theoretical bounds that are especially useful for deriving asymptotic results. However, the theoretical bounds obscure fine details of the behavior of various sketches and they are mostly not indicative of which type of sketches should be used in practice. Moreover, no significant empirical comparison between various sketching techniques has been published, which makes the choice even harder. In this paper, we take a close look at the sketching techniques proposed in the literature from a statistical point of view with the goal of determining properties that indicate the actual behavior and producing tighter confidence bounds. Interestingly, the statistical analysis reveals that two of the techniques, Fast-AGMS and Count-Min, provide results that are in some cases orders of magnitude better than the corresponding theoretical predictions. We conduct an extensive empirical study that compares the different sketching techniques in order to corroborate the statistical analysis with the conclusions we draw from it. The study indicates the expected performance of various sketches, which is crucial if the techniques are to be used by practitioners. The overall conclusion of the study is that Fast-AGMS sketches are, for the full spectrum of problems, either the best, or close to the best, sketching technique. This makes Fast-AGMS sketches the preferred choice irrespective of the situation.

Categories and Subject Descriptors: H.2.4 [Database Management]: Systems – Query processing; G.3 [Probability and Statistics]: Distribution functions

General Terms: Algorithms, Experimentation, Measurement, Performance

Keywords: Size of join estimation, AGMS sketches, Fast-AGMS sketches, Fast-Count sketches, Count-Min sketches

*Material in this paper is based upon work supported by the National Science Foundation under Grant No. 0448264.

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

1. INTRODUCTION

Through research in the last decade, sketching techniques evolved as the premier approximation technique for aggregate queries over data streams. All sketching techniques share one common feature: they are based on randomized algorithms that combine random seeds with data to produce random variables that have distributions connected to the true value of the aggregate being estimated. By measuring certain characteristics of the distribution, correct estimates of the aggregate are obtained. The interesting thing about all sketching techniques that have been proposed is that the combination of randomization and data is a linear operation with the result that, as observed in [5, 13], sketching techniques can be used to perform distributed computation of aggregates without the need to send the actual data values. The tight connection with both data-streaming and distributed computation makes sketching techniques important from both the theoretical and practical point of view.

Sketches can be used either as the actual approximation technique, in which case they require a single pass over the data or, in order to improve the performance, as the basic technique in multi-pass techniques such as *skimmed sketches* [10] and *red-sketches* [11]. For either application, it is important to understand as well as possible its approximation behavior depending on the characteristics of the problem and to be able to predict as accurately as possible the estimation error. As opposed to most approximation techniques – one of the few exceptions are sampling techniques [12] – theoretical approximation guarantees in the form of confidence bounds were provided for all types of sketches from the beginning [2]. All the theoretical guarantees that we know of are expressed as memory and update time requirements in terms of big-$\mathcal{O}$ notation, and are parameterized by ϵ, the target relative error, δ, the target confidence (the relative error is at most ϵ with probability at least $1 - \delta$), and the characteristics of the data – usually the first and the second frequency moments. While these types of theoretical results are useful in theoretical computer science, the fear is that they might hide details that are relevant in practice. In particular, it might be hard to compare methods, or some methods can look equally good according to the theoretical characterization, but differ substantially in practice. An even more significant concern, which we show to be per-

fectly justified, is that some of the theoretical bounds are too conservative.

In this paper, we set out to perform a detailed study of the statistical and empirical behavior of the four basic sketching techniques that have been proposed in the research literature for computing size of join and related problems: AGMS [2, 1], Fast-AGMS [5], Count-Min [6], and Fast-Count [20] sketches. The initial goal of the study was to complement the theoretical results and to make sketching techniques accessible and useful for the practitioners. While accomplishing these tasks, the study also shows that, in general, the theoretical bounds are conservative by at least a constant factor of 3. For Fast-AGMS and Count-Min sketches, the study shows that the theoretical prediction is off by many orders of magnitude if the data is skewed. As part of our study we provide practical confidence intervals for all sketches except Count-Min. We use statistical techniques to provide confidence bounds at the same time the estimate is produced without any prior knowledge about the distribution[1]. Notice that prior knowledge is required in order to use the theoretical confidence bounds provided in the literature and might not actually be available in practice. As far as we know, there not exists any detailed statistical study of sketching techniques and only limited empirical studies to asses their accuracy. The insight we get from the statistical analysis and the extensive empirical study we perform allows us to clearly show that, from a practical point of view, Fast-AGMS sketches are the best basic sketching technique. The behavior of these sketches is truly exceptional and much better than previously believed – the exceptional behavior is masked by the result in [5], but revealed by our detailed statistical analysis. The timing results for the three hash-based sketching techniques (Fast-AGMS, Fast-Count, and Count-Min) reveal that sketches are practical, easily able to keep up with streams of million tuples/second.

The rest of the paper is organized as follows. In Section 2 we provide important results from statistics that are needed throughout the paper. In Section 3 we give an overview of the four basic sketching techniques proposed in the literature. Section 4 contains our statistical analysis of the four sketching techniques with insights on their behavior. Section 5 contains the details and results of our extensive empirical study that corroborates the statistical analysis. In Section 6 we discuss the impact of our results from a practical point of view, and then we conclude in Section 7.

2. PRELIMINARIES

In this section we give a short overview of the results from statistics that are used throughout the paper. Some of them, like the central limit theorem for the mean estimator, are well known but require further comments, and others, like the central limit theorem and the confidence bounds for the median estimator, are unknown in the database community.

The abstract statistical problem we tackle throughout the paper is the following. Given $X_1, \ldots, X_n$ independent instances of a generic random variable X, estimate as precisely as possible the expected value $E[X]$ and provide confidence bounds of the estimate by considering only the observed values $X_1, \ldots, X_n$.

Usually – this is the prevalent situation in database literature, but also in statistics[2] – the *mean* of $X_1, \ldots, X_n$ is considered as the proper estimator for $E[X]$. It is known from statistics [19] that when the distribution of X is normal, the mean $\bar{X}$ is the *uniformly minimum variance unbiased estimator* (UMVUE), the *minimum risk invariant estimator* (MRIE), and the *maximum likelihood estimator* (MLE) for $E[X]$. This is strong evidence that $\bar{X}$ should be used as the estimator of $E[X]$ when the distribution is normal or almost normal. While this is the case in most circumstances, it is also known that there are distributions, like the Cauchy distribution, for which the mean is a poor choice as an estimator of $E[X]$. In the case of Cauchy distribution, the mean can be shown to have the same distribution as a single random sample X_i, thus averaging does not improve the error of the estimate over the error of a single sample. In such cases, the *median* of the samples has much better performance as an estimate of the expected value. The Cauchy distribution example is an extreme case in which the mean estimator is not efficient but, as we will see in Section 4, some of the distributions resulting from sketching can benefit from the use of medians instead of means. Consequently, in this section we explain not only how confidence bounds can be obtained for the mean estimator, but also for the median estimator. Moreover, we provide guidelines for choosing the appropriate estimator for particular situations.

2.1 Mean Estimator

Part of the appeal for using mean as an estimator of the expectation $E[X]$ is the *Central Limit Theorem* (CLT):

THEOREM 1 (MEAN CLT [19]). *Let $X_1, \ldots, X_n$ be independent random variables with the same distribution as X and $\bar{X}$ be the average of the n random variables. Then, as long as $Var[X] < \infty$:*

$$\bar{X} \rightarrow_d N\left(E[X], \frac{Var[X]}{n}\right)$$

Essentially, CLT states that the distribution of the mean is asymptotically a normal distribution centered on the expected value and having variance $\frac{\mathrm{Var}[X]}{n}$, as long as the variance $\mathrm{Var}[X]$ is finite. If $\mathrm{Var}[X]$ is known (or can be estimated from the samples), confidence bounds for $\bar{X}$ can be immediately derived:

PROPOSITION 1 (MEAN BOUNDS). *For the same setup as in Theorem 1, the asymptotic confidence bounds for $\bar{X}$ are:*

$$P\left[|\bar{X} - E[X]| \leq z_{\alpha/2}\sqrt{\frac{Var[X]}{n}}\right] \geq 1 - \alpha$$

where z_β is the β quantile of the normal $N(0,1)$ distribution (i.e., the point for which the probability of the $N(0,1)$ random variable to be smaller than the point is β).

Since fast series algorithms for the computation of z_β are widely available[3], the computation of confidence bounds for $\bar{X}$ is straightforward. Usually, the CLT approximation of

[1]This is the common practice for sampling estimators [12].

[2]Few statistics books, [19] for example, compare sample median and sample mean as estimators for $E[X]$.

[3]The GNU Scientific Library (GSL) implements pdf, cdf, inverse cdf, and other functions for the most popular distributions, including the normal distribution.

the distribution of the mean and the confidence bounds produced with it are correct starting with tens to hundreds of samples being averaged. Thus, from a practical point of view, we expect the confidence bounds in Proposition 1 to be accurate as long as tens of random variables are averaged.

Notice that in order to characterize the mean estimator, the variance of X has to be determined. When $\mathrm{Var}[X]$ is not known – this is the case for sketches since estimating the variance is at least as hard as estimating the expected value – the variance can be estimated from the samples in the form of sample variance. This is the common practice in statistics and also in database literature (approximate query processing with sampling).

2.2 Median Estimator

As we have already mentioned, there exist distributions for which the median is close to the expected value, for example symmetric or almost symmetric distributions. In such cases, sample median can also be used as an estimator of the expected value. While for the majority of the distributions encountered in practice the sample median is not as efficient as the sample mean, there exist distributions for which sample mean has poor performance, for example the Cauchy distribution. In such cases, sample median might be the only viable estimator. Since we want estimators that are efficient for each particular scenario, it worths investigating sample median as an estimator in order to asses its properties.

We start the investigation of the sample median estimator by introducing its corresponding central limit theorem. Then we identify what characteristics of the distribution of X indicate whether sample mean or sample median should be chosen as the estimator of $E[X]$. A more technical treatment can be found in [19] (Section 5.3).

THEOREM 2 (MEDIAN CLT [19]). *Let $X_1, \ldots, X_n$ be independent random variables with the same distribution as X and $\tilde{X}$ be the median of the n random variables. Then, as long as the density function f of the distribution of X has the property $f(\theta) > 0$:*

$$\tilde{X} \rightarrow_d N\left(\theta, \frac{1}{4n \cdot f(\theta)^2}\right)$$

where θ is the true median of the distribution.

2.2.1 Efficiency

For the cases when the distribution is symmetric, thus the expected value and the median coincide, or when the difference between the median and the expected value is insignificant, the decision with respect to which of the sample mean or sample median to be used as an estimate for the expected value is reduced to establishing which of the two has smaller variance. Since for both estimators the variance decreases by a factor of n, the question is further reduced to comparing the variance $\mathrm{Var}[X]$ and the quantity $\frac{1}{4f(\theta)^2}$. This relation is established in statistics through the notion of *asymptotic relative efficiency*:

DEFINITION 1 ([19]). *The relative efficiency of the median estimator $\tilde{X}$ with respect to the mean estimator $\bar{X}$, shortly the* efficiency *of the distribution of X with the density function f, is defined as:*

$$e(f) = 4f(\theta)^2 \, Var[X]$$

The efficiency of a distribution for which $E[X] = \theta$ indicates which of the sample mean or the sample median is a better estimator of $E[X]$. More precisely, $e(f)$ indicates the reduction in mean squared error if the sample median is used instead of the sample mean. When $e(f) > 1$, sample median is a better estimator, while for $e(f) < 1$ sample mean provides better estimates.

An important case to consider is when X has normal distribution. In this situation, the efficiency is independent of the distribution and it is equal to $\frac{2}{\pi} \approx 0.64$ (derived from the above definition and the pdf of the normal distribution). This immediately suggests that when the random variable X is itself an average of other random variables, i.e., by Mean CLT the distribution of X is asymptotically normal, the mean estimator is more efficient than the median estimator. We exploit this result for analyzing sketches in Section 4. In terms of mean squared error, the mean estimator has error 0.64 times smaller, while in terms of root mean squared error or relative error, the mean estimator has error 0.8 times smaller (it is 25% better). While this is a noticeable discrepancy between the mean and the median estimators, the performance of the median estimator can be also acceptable.

2.2.2 Signs of Supra-Unitary Efficiency

As we pointed out in the previous subsection, when the efficiency is supra-unitary, i.e., $e(f) > 1$, medians should be preferred to means for estimating the expected value, if the distribution is symmetric (or almost symmetric). An interesting question is what property of the distribution – hopefully involving only moments since they are significantly easier to determine than other characteristics of discrete distributions – indicates supra-unitary efficiency. According to the statistics literature [3], *kurtosis* is the best indicator of supra-unitary efficiency.

DEFINITION 2 ([3]). *The kurtosis k of the distribution of the random variable X is defined as:*

$$k = \frac{E\left[(X - E[X])^4\right]}{Var[X]^2}$$

For normal distributions, the kurtosis is equal to 3 irrespective of the parameters. Even though there not exists a distribution independent relationship between the kurtosis and the efficiency, empirical studies [16] showed that whenever $k \leq 6$ the mean is a better estimator of $E[X]$, while for $k > 6$ the median is the better estimator.

2.2.3 Confidence Bounds

For standard distributions, the quantity $f(\theta)$ is estimated directly from the pdf, which readily gives error bounds similar to the ones in Proposition 1. Unfortunately, for the distributions of the sketch estimators that appear in this paper, it is hard to compute $f(\theta)$, thus Median CLT cannot be used to derive confidence bounds. The rate of convergence and the normality of the distribution can be used though if efficiency is determined experimentally and the variance is known. As explained above, the variance usually needs to be estimated from the samples, thus methods that estimate confidence bounds for the median estimator directly from samples seem preferable. Fortunately, such methods were developed in the statistics literature [17, 15].

PROPOSITION 2 (MEDIAN BOUNDS [15]). *For the same setup as in Theorem 2, the sample-based confidence bounds for $\tilde{X}$ are:*

$$P\left[|\tilde{X} - \theta| \leq t_{p,1-\alpha/2}\mathtt{SE}(\tilde{X})\right] \geq 1 - \alpha$$

where $t_{p,\beta}$ is the β quantile of the Student t-distribution with p degrees of freedom and $\mathtt{SE}(\tilde{X})$ is the estimate for the standard deviation of $\tilde{X}$ given by:

$$\mathtt{SE}(\tilde{X}) = \frac{X_{(U_n)} - X_{(L_n+1)}}{2}$$
$$L_n = \left\lceil \frac{n}{2} \right\rceil - \left\lceil \sqrt{\frac{n}{4}} \right\rceil$$
$$U_n = n - L_n$$

2.3 Median of Means Estimator

Instead of using only the mean or the median as an estimator for the expected value, we can also consider combined estimators. One possible combination that is used in conjunction to sketching techniques (see Section 3) is to group the samples into chunks of equal size, compute the mean of each chunk, and then the median of the means, thus obtaining the overall estimator for the expected value. To characterize this estimator using distribution independent bounds, a combination of the Chebyshev and Chernoff bounds can be used:

PROPOSITION 3 ([2]). *The median Y of $2\ln(\frac{1}{\alpha})$ means, each averaging $\frac{8}{\epsilon^2}$ iid samples of the random variable X, has the property:*

$$P\left[|Y - E[X]| \leq \epsilon\sqrt{Var[X]}\right] \geq 1 - \alpha$$

Suppose that we want to obtain 95% confidence intervals using the above bound. Then, the number of means for which we compute the median should be $2\ln\frac{1}{.05} = 2\ln 20 \approx 9$. If we have a memory budget n, then each mean is the average of $\frac{n}{9}$ samples, thus $\epsilon = \sqrt{\frac{72}{n}} \approx 8.49 \cdot \sqrt{\frac{1}{n}}$. The width of the confidence interval in terms of $\sqrt{\frac{\mathrm{Var}[X]}{n}}$ is thus $2 \cdot 8.49$. If we apply the CLT theorems for means and medians, the means will have a normal distribution with variance $\frac{\mathrm{Var}[X]}{n/9}$ and the median of the 9 means will have the variance $\frac{1}{9e(N)} \cdot \frac{\mathrm{Var}[X]}{n/9}$, with $e(N) = \frac{2}{\pi}$ the efficiency of the normal distribution. The variance of Y is thus $\frac{1}{e(N)} \cdot \frac{\mathrm{Var}[X]}{n} \approx 1.57 \cdot \frac{\mathrm{Var}[X]}{n}$. With this, the width of the CLT-based confidence bound for Y with respect to $\sqrt{\frac{\mathrm{Var}[X]}{n}}$ is $2 \cdot 1.25 \cdot 2.24 = 2 \cdot 2.8$, which is $\frac{8.49}{2.8} \approx 3.03$ times smaller than the confidence interval obtained using Proposition 3.

An important point in the above derivation of the CLT confidence bounds for Y is the fact that the confidence interval is wider by $\sqrt{\frac{1}{e(N)}} \approx 1.25$ if medians are used, compared to the situation when the estimator is only the mean (with no medians). This implies that the median of means estimator is always inferior to the mean estimator irrespective of the distribution. Thus, from a practical point of view based on the efficiency of the distribution, the estimator should be either the mean ($e < 1$), or the median ($e > 1$), but never the median of means.

3. SKETCHES

Sketches are small-space summaries of data suited for massive, rapid-rate data streams processed either in a centralized or distributed environment. Queries are not answered precisely anymore, but rather approximately, by considering only the synopsis (sketch) of the data. Typically, a sketch consists of multiple counters corresponding to *random variables* with required properties in order to provide answers with provable probabilistic guarantees. The existing sketching techniques differ in how the random variables are organized, thus the update procedure, and how the answer to a given query is computed. In this section we provide an overview of the existing sketching techniques.

Let $S = (e_1, w_1), (e_2, w_2), \ldots, (e_s, w_s)$ be a data stream, where the keys e_i are members of the set $I = \{0, 1, \ldots, N-1\}$ and w_i represent frequencies. The *frequency vector* $\bar{f} = [f_0, f_1, \ldots, f_{N-1}]$ over the stream S consists of the elements f_i defined as $f_i = \sum_{j:e_j=i} w_j$. The key idea behind the existing sketching techniques is to represent the domain-size frequency vector as a much smaller *sketch* vector $\bar{x}_f$ [5] that can be easily maintained as the updates are streaming by and that can provide good approximations for a wide spectrum of queries.

Our focus is on sketching techniques that approximate the *size of join* of two data streams. The size of join is defined as the *inner-product* of the frequency vectors $\bar{f}$ and $\bar{g}$, $\bar{f} \odot \bar{g} = \sum_{i=0}^{N-1} f_i g_i$. As shown in [18], this operator is generic since other classes of queries can be reduced to the size of join computation. For example, a range query over the interval $[\alpha, \beta]$, i.e., $\sum_{i=\alpha}^{\beta} f_i$, can be expressed as the size of join between the data stream S and a virtual stream consisting of a tuple $(i, 1)$ for each $\alpha \leq i \leq \beta$. Notice that point queries are range queries over size zero intervals, i.e., $\alpha = \beta$. Also, the second frequency moment or the self-join size of S is nothing else than the inner-product $\bar{f} \odot \bar{f}$. The following sections introduce the existing sketching structures used for approximating the size of join of two data streams.

3.1 AGMS Sketches

The i^{th} entry of the size n AGMS (or, *tug-of-war*) [2, 1] sketch vector is defined as the random variable $x_f[i] = \sum_{j=0}^{N-1} f_j \cdot \xi_i(j)$, where $\{\xi_i(j) : j \in I\}$ is a family of uniformly distributed ± 1 4-wise independent random variables, with different families being independent. The advantage of using ± 1 random variables comes from the fact that they can be efficiently generated in small space [18]. When a new data stream item (e, w) arrives, all the counters in the sketch vector are updated as $x_f[i] = x_f[i] + w \cdot \xi_i(e)$, $1 \leq i \leq n$. The time to process an update is thus proportional with the size of the sketch vector.

It can be shown that $X[i] = x_f[i] \cdot x_g[i]$ is an unbiased estimator of the inner-product of the frequency vectors $\bar{f}$ and $\bar{g}$, i.e., $E[X[i]] = \bar{f} \odot \bar{g}$. The variance of the estimator is:

$$\mathrm{Var}[X[i]] = \left(\sum_{j \in I} f_j^2\right)\left(\sum_{k \in I} g_k^2\right) + \left(\sum_{j \in I} f_j g_j\right)^2 - 2 \cdot \sum_{j \in I} f_j^2 g_j^2 \quad (1)$$

By averaging n independent estimators, $Y = \frac{1}{n}\sum_{i=1}^{n} X[i]$, the variance can be reduced by a factor of n, i.e., $\mathrm{Var}[Y] =$

$\frac{\text{Var}[X[i]]}{n}$, thus improving the estimation error. In order to make the estimation more stable, the original solution [2] returned as the result the median of m Y estimators, i.e., $Z = Median_{1 \leq k \leq m} Y[k]$.

Notice the tradeoffs involved by the AGMS sketch structure. In order to decrease the error of the estimator (proportional with the variance), the size n of the sketch vector has to be increased. Since the space and the update-time are linear functions of n, an increase of the sketch size implies a corresponding increase of these two quantities.

The following theorem relates the accuracy of the estimator with the size of the sketch, i.e., $n = \mathcal{O}(\frac{1}{\epsilon^2})$ and $m = \mathcal{O}(\log \frac{1}{\delta})$.

THEOREM 3 ([1]). *Let $\bar{x}_f$ and $\bar{x}_g$ denote two parallel sketches comprising $\mathcal{O}\left(\frac{1}{\epsilon^2} \log \frac{1}{\delta}\right)$ counters each, where ϵ and $1-\delta$ represent the desired bounds on error and probabilistic confidence, respectively. Then, with probability at least $1-\delta$, $Z \in (\bar{f} \odot \bar{g} \pm \epsilon ||\bar{f}||_2 ||\bar{g}||_2)$. The processing time required to maintain each sketch is $\mathcal{O}\left(\frac{1}{\epsilon^2} \log \frac{1}{\delta}\right)$ per update.*

$||\bar{f}||_2 = \sqrt{\bar{f} \odot \bar{f}} = \sqrt{\sum_{i \in I} f_i^2}$ is the L_2 norm of $\bar{f}$ and $||\bar{g}||_2 = \sqrt{\bar{g} \odot \bar{g}} = \sqrt{\sum_{i \in I} g_i^2}$ is the L_2 norm of $\bar{g}$, respectively.

3.2 Fast-AGMS Sketches

As we have already mentioned, the main drawback of AGMS sketches is that any update on the stream affects all the entries in the sketch vector. Fast-AGMS sketches [5], as a refinement of Count sketches proposed in [4] for detecting the most frequent items in a data stream, combine the power of ± 1 random variables and hashing to create a scheme with a significantly reduced update time while preserving the error bounds of AGMS sketches. The sketch vector $\bar{x}_f$ consists of n counters, $x_f[i]$. Two independent random processes are associated with the sketch vector: a family of ± 1 4-wise independent random variables ξ and a 2-universal hash function $h : I \rightarrow \{1, \ldots, n\}$. The role of the hash function is to scatter the keys in the data stream to different counters in the sketch vector, thus reducing the interaction between the keys. Meanwhile, the unique family ξ preserves the dependencies across the counters. When a new data stream item (e, w) arrives, only the counter $x_f[h(e)]$ is updated with the value of the function ξ corresponding to the key e, i.e., $x_f[h(e)] = x_f[h(e)] + w \cdot \xi(e)$.

Given two parallel sketch vectors $\bar{x}_f$ and $\bar{x}_g$ using the same hash function h and ξ family, the inner-product $\bar{f} \odot \bar{g}$ is estimated by $Y = \sum_{i=1}^{n} x_f[i] \cdot x_g[i]$. The final estimator Z is computed as the median of m independent basic estimators Y, i.e., $Z = Median_{1 \leq k \leq m} Y[k]$. The following theorem relates the number of sketch vectors m and their size n with the error bound ϵ and the probabilistic confidence δ, respectively.

THEOREM 4 ([5]). *Let $n = \mathcal{O}(\frac{1}{\epsilon^2})$ and $m = \mathcal{O}(\log \frac{1}{\delta})$. Then, with probability at least $1-\delta$, $Z \in (\bar{f} \odot \bar{g} \pm \epsilon ||\bar{f}||_2 ||\bar{g}||_2)$. Sketch updates are performed in $\mathcal{O}(\log \frac{1}{\delta})$ time.*

The above theorem states that Fast-AGMS sketches provide the same guarantees as basic AGMS sketches, while requiring only $\mathcal{O}(\log \frac{1}{\delta})$ time to process the updates and using only one ξ family per sketch vector (and one additional hash function h). Moreover, notice that only the sketch vector size is dependent on the error bound ϵ.

3.3 Fast-Count Sketches

Fast-Count sketches, introduced in [20], provide the error guarantees and the update time of Fast-AGMS sketches, while requiring only one underlying random process – hashing. The tradeoffs involved are the size of the sketch vector (or, equivalently, the error) and the degree of independence of the hash function. The sketch vector consists of the same n counters as for AGMS sketches. The difference is that there exists only a 4-universal hash function associated with the sketch vector. When a new data item (e, w) arrives, w is directly added to a single counter, i.e., $x_f[h(e)] = x_f[h(e)] + w$, where $h : I \rightarrow \{1, \ldots, n\}$ is the 4-universal hash function.

The size of join estimator is defined as (this is a generalization of the second frequency moment estimator in [20]):

$$Y = \frac{1}{n-1} \left[n \cdot \sum_{i=1}^{n} x_f[i] \cdot x_g[i] - \left(\sum_{i=1}^{n} x_f[i] \right) \left(\sum_{i=1}^{n} x_g[i] \right) \right]$$

The complicated form of Y is due to the biasness of the natural estimator $Y' = \sum_{i=1}^{n} x_f[i] \cdot x_g[i]$. Y is obtained by a simple correction of the bias of Y'. It can be proved that Y is an unbiased estimator of the inner-product $\bar{f} \odot \bar{g}$. Its variance is almost identical to the variance of the Y estimator for AGMS (Fast-AGMS) sketches in (1). The only difference is the multiplicative factor, $\frac{1}{n-1}$ for Fast-Count sketches, compared to $\frac{1}{n}$ for AGMS sketches. Hence, given desirable error guarantees, Fast-Count sketches require one additional entry in the sketch vector. For large values of n, e.g., $n > 100$, the difference in variance between AGMS (Fast-AGMS) and Fast-Count sketches can be ignored and the results in Theorem 4 apply.

3.4 Count-Min Sketches

Count-Min sketches [6] have almost the same structure as Fast-Count sketches. The only difference is that the hash function is drawn randomly from a family of 2-universal hash functions instead of 4-universal. The update procedure is identical to Fast-Count sketches, only the counter $x_f[h(e)]$ being updated as $x_f[h(e)] = x_f[h(e)] + w$ when the item (e, w) arrives. The size of join estimator is defined in a natural way as $Y = \sum_{i=1}^{n} x_f[i] \cdot x_g[i]$ (notice that Y is actually equivalent with the above Y' estimator). It can be shown that Y is an overestimate of the inner-product $\bar{f} \odot \bar{g}$. In order to minimize the over-estimated quantity, the minimum over m independent Y estimators is computed, i.e., $Z = Min_{1 \leq k \leq m} Y[k]$. Notice the different methods applied to correct the bias of the size of join estimator Y'. While Fast-Count sketches define an unbiased estimator Y based on Y', Count-Min sketches select the minimum over multiple such overestimates.

The relationship between the size of the sketch and the accuracy of the estimator Z is expressed by the following theorem:

THEOREM 5 ([6]). *$Z \leq \bar{f} \odot \bar{g} + \epsilon ||\bar{f}||_1 ||\bar{g}||_1$ with probability $1-\delta$, where the size of the sketch vector is $n = \mathcal{O}(\frac{1}{\epsilon})$ and the minimum is taken over $m = \mathcal{O}(\log \frac{1}{\delta})$ sketch vectors. Updates are performed in time $\mathcal{O}(\log \frac{1}{\delta})$.*

$||\bar{f}||_1 = \sum_{i \in I} f_i$ and $||\bar{g}||_1 = \sum_{i \in I} g_i$ represent the L_1 norms of the vectors $\bar{f}$ and $\bar{g}$, respectively. Notice the dependence on the L_1 norm, compared to the dependence on the L_2 norm for AGMS sketches. The L_2 norm is always smaller

than the L_1 norm. In the extreme case of uniform frequency distributions, L_2 is quadratically smaller than L_1. This implies increased errors for Count-Min sketches as compared to AGMS sketches, or, equivalently, more space in order to guarantee the same error bounds (even though the sketch vector size is only $\mathcal{O}(\frac{1}{\epsilon})$).

Sketch	Size of Join		
	Large		Small
	Low Skew	High Skew	
AGMS	0	0	−
Fast-AGMS	0	0	−
Fast-Count	0	0	−
Count-Min	−	0	−

Table 1: Expected theoretical performance. The scale has three types of values: 0, +, and −. 0 is the reference value corresponding to the AGMS self-join size. − indicates worse results, while + indicates better results.

3.5 Comparison

Given the above sketching techniques, we qualitatively compare their expected performance based on the existing theoretical results. The techniques are compared relatively to the result obtained by the use of AGMS sketches for the self-join size problem, known to be asymptotically optimal [2]. The size of join results are considered relatively to the product of the L_2 (L_1 for Count-Min) norms of the data streams. Notice that large results correspond to the particular self-join size problem. Low skew corresponds to frequency vectors for which the ratio $\frac{L_1}{L_2}$ is close to $\sqrt{N}$ (uniform distribution), while for high skew the ratio $\frac{L_1}{L_2}$ is close to 1.

Table 1 summarizes the results predicted by the theory. Since the bounds for AGMS, Fast-AGMS, and Fast-Count sketches are identical, they have the same theoretical behavior. For small size of join results, the performance of these three methods worsens. Count-Min sketches have a distinct behavior due to their dependency on the L_1 norm. Their performance is highly influenced not only by the size of the result, but also by the skewness of the data. For low skew data, the performance is significantly worse than the performance of AGMS sketches. Since $L_1 \geq L_2$, the theoretical performance for Count-Min sketches is always worse than the performance of AGMS (Fast-AGMS, Fast-Count) sketches.

4. STATISTICAL ANALYSIS OF SKETCH ESTIMATORS

From a purely practical point of view, we are interested in approximation techniques that are reasonably easy to implement, are fast (i.e., small update time for the synopsis data-structure), have good accuracy and can estimate as precisely as possible their error through confidence intervals. Although the same goals are followed from the theoretical point of view, we insist on deriving simple formulae for the error expressed in terms of asymptotic big-$\mathcal{O}$ notation. This is perfectly reflected by the theoretical results we presented in the previous section. The problem with theoretical results is the fact that, since we always insist on simple/expressible formulae, we might ignore details that matter at least in some cases – the theoretical results are always conservative, but they might be too conservative sometimes. In this section, we explore this problem by asking the following three questions that reflect the difference between the pragmatic and the theoretical point of view:

- How can the theoretical bounds be used in practice to compute confidence bounds for the actual estimates?
- How *tight* are the theoretical confidence bounds? We are not only interested in tight bounds for some situations (i.e., tight in a theoretical sense), but in confidence bounds that are realistic for all situations. The golden standard we are aiming for is confidence bounds similar to the ones for sampling techniques.
- All sketching techniques consist in combining multiple independent instances of elementary sketches using *means*, *medians*, or *minimum* estimators in order to improve the accuracy. Which of the three estimators is more efficient for each of the four sketching techniques?

We use a large-scale statistical analysis based on experiments in order to answer the above questions. Due to space constraints, only the most relevant results are presented. The plots in this section have statistical significance and are not highly sensitive at the experimental setup (Section 5).

4.1 AGMS Sketches

As explained in Section 2.3, the mean is always preferable to the median of means as an estimator for the expected value of a random variable given as samples. The difference is a 25% reduction in error if the mean is used. To produce confidence bounds, we can use Proposition 1. The value of the variance is either the exact one (if it can be determined) or, more realistically, an estimate computed from the samples. The theoretical confidence bounds in Proposition 3 are 3 times wider than the CLT bounds, as explained in Section 2.3. This discrepancy between the theoretical bounds and the effective error was observed experimentally in [18, 8], but it was not explained.

We explore which estimator – minimum, median, or mean – to use for AGMS sketches. In order to accomplish this task, we plotted the distribution of the basic estimator for a large spectrum of problems. Figure 1 is a generic example for the form of the distribution. It is clear from this figure that both the minimum and the median are poor choices. The median is a poor choice because the distribution of the elementary AGMS sketches is not symmetric and there exists a variable gap between the mean and the median of the distribution, gap that is not easily to compute and, thus, to compensate for. In order to verify that the mean is the optimal estimator (as the theory predicts), we plot its distribution for the same input data (Figure 2). As expected, the distribution is normal and its expected value is exactly the true result.

4.2 Fast-AGMS Sketches

Comparing Theorem 3 and 4 that characterize the AGMS and the Fast-AGMS (F-AGMS) sketches, respectively, we observe that the predicted accuracy is identical, but Fast-AGMS have significantly lower update time. The lower up-

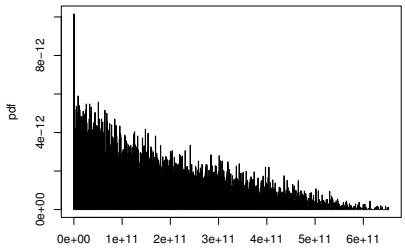

Figure 1: AGMS distribution

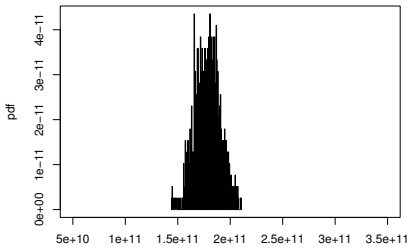

Figure 2: Mean AGMS distribution

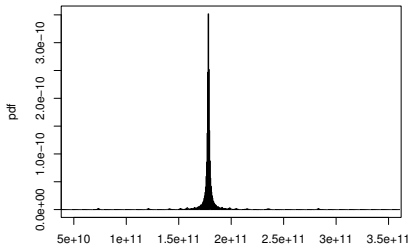

Figure 3: F-AGMS distribution

date time and the same theoretical accuracy immediately indicate that F-AGMS should be preferred to AGMS. In the previous section, we saw a discrepancy of a factor of 3 between the theoretical bounds and the CLT-based bounds for AGMS sketches and the possibility of a 25% improvement if medians are replaced by means. In this section, we investigate the statistical properties of F-AGMS sketches in order to determine possibly tighter confidence bounds and, thus, improve the error.

We start the investigation on the statistical properties of Fast-AGMS sketches with the following result that is used in the proof of Theorem 4:

PROPOSITION 4 ([5]). *Let X be the Fast-AGMS estimator obtained with a family of 4-universal hash functions $h : I \to B$ and a 4-wise independent family ξ of ± 1 random variables. Then,*

$$E_{h,\xi}[X] = E[X_{AGMS}]$$
$$E_h[Var_\xi[X]] = \frac{1}{B} Var[X_{AGMS}]$$

The first two moments of the elementary Fast-AGMS sketch coincide with the first two moments of the average of B elementary AGMS sketches (in order to have the same space usage). This is a somewhat unexpected result since it suggests that the hashing plays the same role as averaging when it comes to reducing the variance. This might suggest that the transformation on the distribution of elementary F-AGMS sketches is the same, i.e., the distribution becomes normal and the variance is reduced by the number of buckets. The following result gives the first discrepancy between Fast-AGMS and AGMS sketches:

PROPOSITION 5. *With the same setup as in Proposition 4, we have:*

$$Var_h[Var_\xi[X]] = \frac{B-1}{B^2}\left[3\left(\sum_i f_i^2 g_i^2\right)^2 + 4\sum_i f_i^3 g_i \sum_j f_j g_j^3 + \sum_i f_i^4 \sum_j g_j^4 - 8\sum_i f_i^4 g_i^4\right]$$

The moment $Var_h[Var_\xi[X]]$ is a lower bound on the fourth moment of X, for which we cannot derive a nice closed-form formula since the ξ family is only required to be 4-wise independent and 8-wise independence in needed to remove the dependency of the formula on the actual generating scheme. Also, if the hash function h is only 2-universal, instead of 4-universal, even more terms are added to the expression, thus the fourth moment is even higher, resulting in higher efficiency – a desirable outcome. To see why the distribution of Fast-AGMS can be highly different when compared to the distribution of B averages of AGMS sketches, we estimate the kurtosis of the distribution of the elementary Fast-AGMS sketch and compare it with 3, the value of the kurtosis for the average of B elementary AGMS sketches (by CLT the mean distribution is normal, thus its kurtosis is 3). From Figure 4, that depicts the experimental kurtosis and its lower bound in Proposition 5, we observe that when the Zipf coefficient is larger than 1, the kurtosis grows significantly, to the point that it is around 1000 for a Zipf coefficient equal to 5. Using the discussion in Section 2.2.2, starting with Zipf coefficients greater than 1, the median estimator should be preferred, as long as the distribution is (almost) symmetric.

From the above discussion, it seems that the distributions of Fast-AGMS and averages of AGMS are highly different, even though their first two moments coincide. The large kurtosis of Fast-AGMS suggests that the distribution has heavy tails. Indeed, Figure 2 and 3 confirm experimentally these observations. Since the bulk of the distribution for Fast-AGMS occupies much smaller space around the expected value than the bulk of the distribution for averages of AGMS, we expect the efficiency of Fast-AGMS to be large and, thus, the performance to be significantly better for large Zipf coefficients when compared to AGMS. It seems that for Fast-AGMS, as opposed to AGMS, it is preferable to use median as the estimator instead of the mean, since for small Zipf coefficients the distribution of Fast-AGMS is almost normal, thus the error is increased by only 25%, but for large Zipf coefficients the error of the median estimator could be substantially smaller compared to the error of the mean estimator.

The error bounds given by Theorem 4 are likely to be far too conservative since, as we explained above, Fast-AGMS sketches could significantly outperform AGMS sketches due to the large efficiency expected for Zipf coefficients greater than 1. Figure 5 confirms the huge gap (as much as 10 orders of magnitude) that exists between the predicted theoretical error and the experimental error. To obtain practical error bounds for Fast-AGMS sketches, we can use the estimator in Section 2.2.3. In Figure 6, we compare the error (95% quantile error) computed by Theorem 4 with that gener-

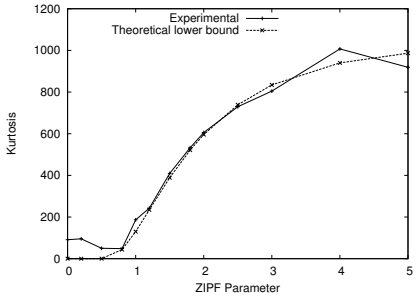

Figure 4: F-AGMS kurtosis

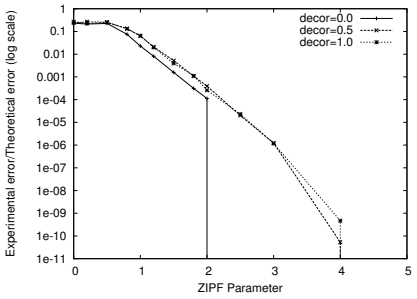

Figure 5: F-AGMS error comparison

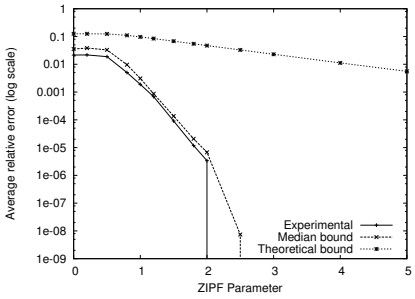

Figure 6: F-AGMS bounds

ated by Proposition 2, and the error measured experimentally. We observe two important facts from these results: (1) the prediction given by Theorem 4 is poor for large Zipf coefficients and (2) the prediction given by Proposition 2 is always accurate. An important difference though is the fact that Proposition 2 does not provide an apriori prediction based on properties of the distribution, but rather estimates the error from the same ingredients (instances of elementary sketches) as the estimator does. Theorem 4, at least in principle, can be used to make apriori predictions if the second frequency moments of the two streams are known.

4.3 Count-Min Sketches

Based on Theorem 5, we expect Count-Min (CM) sketches to have error proportional with the product of the sizes of the two streams and inversely proportional with the number of buckets of the sketch structure. This is the only sketch that has error dependencies on the size of the streams (the first frequency moment or L_1 norm), not the second frequency moment, and the amount of memory (number of hashing buckets), not the squared root of the amount of memory. The dependency on the first frequency moment is worse than the dependency on the squared root of the second frequency moment since the first is always larger or equal than the second. On the other hand, the dependency on the amount of memory is favorable to Count-Min sketches. Based on the theory, we would expect Count-Min sketches to have weak performance for relations with small skew, but comparable performance (not much better though) for skewed relations.

Since the statistical analysis for AGMS and Fast-AGMS was fruitful, we perform a similar investigation for Count-Min. We start with the moments of the estimator:

PROPOSITION 6 ([6]). *If X_{CM} is the elementary Count-Min estimator then:*

$$E[X_{CM}] = \sum_{i \in I} f_i g_i + \frac{1}{B}\left(\sum_{i \in I} f_i \sum_{j \in I} g_j - \sum_{i \in I} f_i g_i\right)$$

$$Var[X_{CM}] = \frac{1}{B} Var[X_{AGMS}]$$

The estimator X_{CM} always overestimates the true result – that is why the minimum is chosen. The proof of Theorem 5 in [6] essentially uses the fact that on average the extra-amount in X_{CM} is $\frac{1}{B}\sum_{i \in I} f_i \sum_{j \in I} g_j$ (if the quantity $\sum_{i \in I} f_i g_i$ is neglected). Interestingly, the variance of the estimator coincides with the variance of averages of B AGMS sketches and the variance of Fast-AGMS sketches. The fundamental difference is that X_{CM} is biased.

When the extra-term $\frac{1}{B}\sum_{i \in I} f_i \sum_{j \in I} g_j$ is significantly larger than $\sqrt{\frac{1}{B} Var[X_{AGMS}]}$ (i.e., the extra-term in the expectation is larger than the standard deviation), we expect the distribution of X_{CM} to look like a normal distribution. In such a case, the minimum is just slightly better than the mean and the error is around $\frac{1}{B}\sum_{i \in I} f_i \sum_{j \in I} g_j$, as predicted by the theory. This regime coincides with the situation in which Count-Min sketches have worse performance compared to the other three methods (the conservative error for all the other methods is $\sqrt{\frac{1}{B} Var[X_{AGMS}]}$). When the extra-term in the expectation of X_{CM} is smaller than the standard deviation, the distribution of X_{CM} starts to be severely skewed to the left. The right side of the distribution becomes extremely short, to the point it completely disappears. This starts to happen for the self-join size problem for Zipf coefficients larger than 1. Figure 8 depicts the distribution of X_{CM} for Zipf equal to 1 – the phenomenon we described just started to happen. For distributions that have this shape, the peak and the left side are far from the expected value due to the heavy right tail. For this reason, the minimum statistic used by the X_{CM} estimator behaves much better than predicted by the theory (that suggests that it is close to the expected value). For large Zipf coefficients, when there is a significant probability for the prediction to be perfect, the minimum gives a perfect prediction. To investigate the extent to which Count-Min sketches behave better than the theory predicts, we depicted in Figure 7 the ratio of the theoretical and the actual error for datasets with different Zipf and correlation coefficients (see Section 5). As it can be observed from these results, the theoretical prediction is much larger than the actual error for large Zipf coefficients.

Unfortunately, as opposed to the median estimator, determining error bounds for the minimum estimator is extremely hard and essentially requires perfect knowledge of the distribution. For the situation when the Zipf coefficient is large, since the difference between the error of the minimum and the expected value can be significant, extremely precise information about the distribution is required to produce reasonable error bounds. We could not find any solution to obtain such precise information that would allow us to provide tight confidence bounds for the minimum estimator.

4.4 Fast-Count Sketches

The Fast-Count (FC) elementary estimator is essentially the bias-corrected version of the Count-Min elementary es-

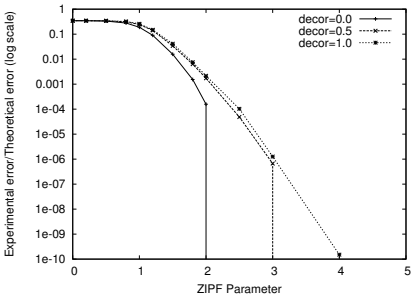

Figure 7: CM error comparison

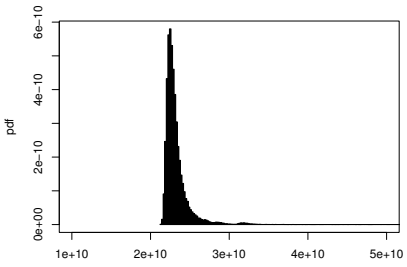

Figure 8: CM distribution

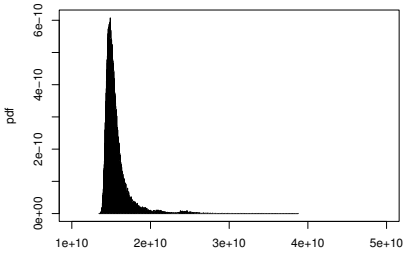

Figure 9: FC distribution

timator. The bias correction is a translation by bias and a scaling by the factor $\frac{B}{B-1}$. This can be observed by comparing Figure 8 and 9 that show the distribution of Count-Min and Fast-Count sketches, respectively, for the self-join size problem with Zipf coefficient equal to 1. The major difference is that Fast-Count elementary sketches are unbiased, while Count-Min sketches are not. Since the distribution can be skewed, there could be a significant difference between the median and the expected value, thus medians cannot be used as an estimate for the expected value. Minimum is not useful either since it will be severely biased, which leaves only the mean as a viable estimator for the expected value in the case of Fast-Count sketches.

If the mean is used to combine the elementary Fast-Count sketches, the resulting estimator has almost (the difference is only $\frac{B}{B-1}$) the same variance as averages of AGMS sketches. Since the distribution of Fast-Count is skewed and the number of elementary sketches that are averaged is usually only in the tens (for efficiency reasons), we do not expect the distribution to be close enough to normal. For this reason, it is safer to use the Chebyshev inequality to give confidence bounds instead of CLT, which gives bounds only 40% wider for 95% confidence intervals. From a practical point of view, we expect the estimation using Fast-Count sketches to have slightly larger fluctuations but essentially the same error behavior as the AGMS sketches. This is confirmed by the experimental results we present in Section 5. This immediately suggests that Fast-Count sketches should be preferred to AGMS sketches since they have essentially the same error but much better update time.

5. EMPIRICAL EVALUATION

The main purpose of the experimental evaluation is to validate and complement the statistical results we obtained in Section 4 for the four sketching techniques. The specific goals are: (1) establish the relative accuracy performance of the four sketching techniques for various problems, and (2) determine the actual update performance. Our main tool in establishing the accuracy of sketches is to measure their error on synthetic datasets for which we control both the skew, via the Zipf coefficient, and the correlation. This allows us to efficiently cover a large spectrum of problems and to draw insightful observations about the performance of sketches. We validate the findings on real-life data sets and other synthetic data generators.

The main findings of the study are:

- AGMS and Fast-Count (FC) sketches have virtually identical accuracy throughout the spectrum of problems if only averages are used for AGMS. FC sketches are preferable since they have significantly smaller update time.
- The performance of Count-Min sketches is strongly dependent on the skew of the data. For small skew, the error is orders of magnitude larger than the error of the other types of sketches. For large skew, CM sketches have the best performance – much better than AGMS and FC.
- Fast-AGMS (F-AGMS) sketches have error at most 25% larger than AGMS sketches for small skew, but the error is orders of magnitude (as much as 6 orders of magnitude for large skew) smaller for moderate and large skew. Their error for large skew is slightly larger than the error of CM sketches.
- All sketches, except CM for small skew, are practical in evaluating self-join size queries. This is to be expected since AGMS sketches are asymptotically optimal [2] for this problem. For size of join problems, F-AGMS sketches remain practical well beyond AGMS and FC sketches. CM sketches have good accuracy as long as the data is skewed.
- F-AGMS, FC, and CM sketches (all of them are based on random hashing) have fast and comparable update performance, ranging between $50 - 400$ ns.

5.1 Testbed and Methodology

Sketch Implementation. We implemented a generic framework that incorporates the sketching techniques mentioned throughout the paper. Algorithms for generating random variables with limited degree of independence [14, 18] are at the core of the framework. Since the sketching techniques have a similar structure, they are designed as a hierarchy parameterized on the type of random variables they employ. Applications have only to instantiate the sketching structures with the corresponding size and random variables, and to call the update and the estimation procedures.

Data Sets. We used two synthetic data generators and one real-life data set in our experiments. The data sets cover an

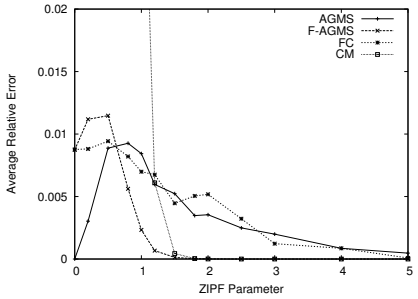

Figure 10: Self-join size

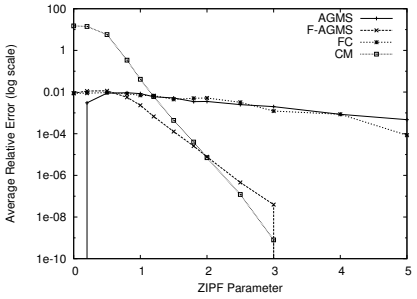

Figure 11: Self-join size (log scale)

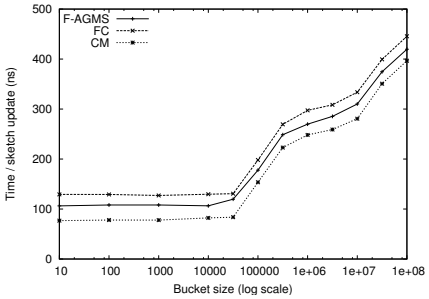

Figure 12: Timing results

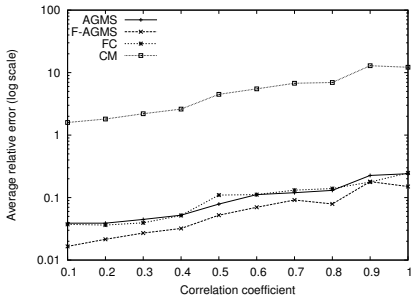

Figure 13: Size of join (Zipf=0.8)

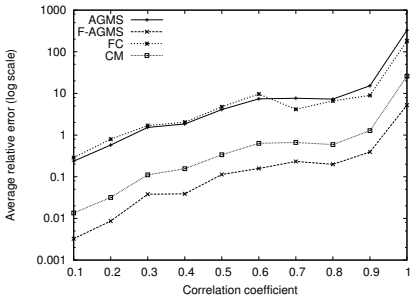

Figure 14: Size of join (Zipf=1.5)

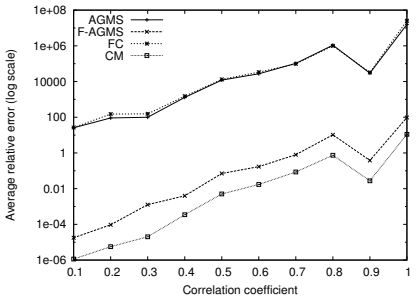

Figure 15: Size of join (Zipf=3.0)

extensive range of possible inputs, thus allowing us to infer general results on the behavior of the compared sketching techniques.

Census data set [7]. This real-life data set was extracted from the Current Population Survey (CPS) data repository, which is a monthly survey of about 50,000 households. Each month's data contains around 135,000 tuples with 361 attributes. We ran experiments for estimating the size of join on the *weekly wage (PTERNWA)* numerical attribute with domain size 288,416 for the surveys corresponding to the months of September 2002 (15,563 records) and September 2006 (14,931 records)[4].

Estan's et al. [9] synthetic data generator. Two tables with approximately 1 million tuples each with a Zipf distribution for the frequencies of the values are randomly generated. The values are from a domain with 5 million values, and for each of the values its corresponding frequency is chosen independently at random from the distribution of the frequencies. We used in our experiments the *memory-peaked* (Zipf=0.8) and the *memory-unpeaked* (Zipf=0.35) data sets.

Synthetic data generator. We implemented our synthetic data generator for frequency vectors. It takes into account parameters such as the domain size, the number of tuples, the frequency distribution, and the correlation (decor = 1 − correlation) coefficient. Out of the large variety of data sets that we conducted experiments on, we focus in this paper on frequency vectors over a $2^{14} = 16,384$ size domain that contain 1 million tuples and having Zipf distributions (the Zipf coefficient ranges between 0 and 5). The degree of correlation between two frequency vectors varies from full correlation to complete independence.

[4]After eliminating the records with missing values.

Answer-Quality Metrics. Each experiment is performed 100 times and the average relative error, i.e., $\frac{|\text{actual}-\text{estimate}|}{\text{actual}}$, over the number of experiments is reported. In the case of direct comparison between two methods, the ratio between their average relative errors is reported. Although we performed the experiments for different sketch sizes, the results are reported only for a sketch structure consisting of 21 vectors with 1024 counters each ($n = 1024$, $m = 21$), since the same trend was observed for the other sketch sizes.

5.2 Results

Self-Join Size Estimation. The behavior of the sketching techniques for estimating the self-join size as a function of the Zipf coefficient of the frequency distribution is depicted in Figure 11 on a logarithmic scale. Figure 10 is a focused view of the same results. As expected, the errors of AGMS and FC sketches are similar (the difference for (close to) uniform distributions is due to the EH3 [18] random number generator). While F-AGMS has almost the same behavior as FC (AGMS) for small Zipf coefficients, the F-AGMS error is drastically decreasing for Zipf coefficients larger than 0.8. These are due to the effect the median estimator has on the distribution of the predicted results: for small Zipf coefficients the distribution is normal, thus the performance of the median estimator is approximately 25% worse, while for large Zipf coefficients the distribution is focused around the true result (Section 4). CM sketches have extremely poor performance for distributions (close to) uniform. This can be explained theoretically by the dependency on the L_1 norm, much larger than the L_2 norm in this regime. Intuitively, uniform distributions have multiple non-zero fre-

quencies that are hashed into the same bucket, thus highly over-estimating the predicted result. The situation changes dramatically at high skew when it is highly probable that each non-zero frequency is hashed to a different bucket, making the estimation almost perfect. Based on these results, we can conclude that F-AGMS is the best (or close to the best – less than 1%) sketch estimator for computing the second frequency moment, irrespective of the skew.

Join Size Estimation. In order to determine the performance of the sketching techniques for estimating the size of join, we conducted experiments based on the Zipf coefficient and the correlation between the two frequency vectors. A correlation coefficient of 0 corresponds to two identical frequency vectors (self-join size). For a correlation coefficient of 1, the frequencies in the two vectors are completely shuffled. The results for different Zipf coefficients are depicted in Figure 13, 14, and 15 as a function of the correlation. It can be clearly seen how the relation between the sketch estimators is changing as a function of the skew (behavior identical to the self-join size). Moreover, it seems that the degree of correlation is affecting similarly all the estimators (the error increases as the degree of correlation is increasing), but it does not affect the relative order given by the Zipf coefficient. The same findings are reinforced in Figure 16, 17, and 18 which depict the relative performance, i.e., the ratio of the average relative errors, between pairs of estimators for computing the size of join. Consequently, we conclude that, as in the case of self-join size, the Zipf coefficient is the only parameter that influences the relative behavior of the sketching techniques for estimating the size of join of two frequency vectors.

Memory Budget. The accuracy of the sketching methods (AGMS is excluded since its behavior is identical to FC) as a function of the space available (in number of counters) is represented in Figure 19 and 20 for Estan's synthetic data sets, and in Figure 21 for the census real-life data set. The error of CM sketches is orders of magnitude worse than the error of the other two methods for the entire range of available memory (due to the low skew). The accuracy of F-AGMS is comparable with that of FC for low skew data, while for skewed data F-AGMS is clearly superior. Notice that the relative performance of the techniques is not dependent on the memory budget.

Update Time. The goal of the timing experiment is to clarify if there exist significant differences in update time between the hash sketches since the random variables they use differ. As shown in Figure 12, all the schemes have comparable update time performance, CM sketches being the fastest, while FC sketches are the slowest. Notice that the relative gap between the schemes shrinks when the number of counters is increasing since more references are made to the main memory. As long as the sketch vector fits into the cache, the update rate is extremely high (around 10 million updates can be executed per second on the test machine[5]), making hash sketches a viable solution for high-speed data stream processing.

[5]The results in Figure 12 are for a Xeon 2.8 GHz processor with 512 KB of cache. The main memory is 4 GB wide.

Sketch	Size of Join		
	Large		Small
	Low Skew	High Skew	
AGMS	0	0	−
Fast-AGMS	0	+	+
Fast-Count	0	0	−
Count-Min	−	+	+

Table 2: Expected statistical/empirical performance (same scale as Table 1).

6. DISCUSSION

As we have seen, the statistical and empirical study in this paper paints a different picture than suggested by the theory (see Table 1). Table 2 summarizes these results qualitatively and indicates that on skewed data, F-AGMS and CM sketches have much better accuracy than expected.

The statistical analysis in Section 4 revealed that the theoretical results for Fast-AGMS (F-AGMS) and Count-Min (CM) sketches do not capture the significantly better accuracy with respect to AGMS and Fast-Count (FC) sketches for skewed data. The reason there exists such a large gap between the theory and the actual behavior is the fact that the median, for F-AGMS, and the minimum, for CM, have a fundamentally different behavior than the mean on skewed data. This behavior defies statistical intuition since most distributions that are encountered in practice have relatively small kurtosis, usually below 20. The distributions of approximation techniques that use hashing on skewed data can have kurtosis in the 1000 range, as we have seen for F-AGMS sketches. For these distributions, the median, as an estimator for the expected value, can have error 10^6 smaller than the mean.

An interesting property of all sketching techniques is that the relationship between their accuracy does not change significantly when the degree of correlation changes, as indicated by Figure 16, 17, and 18. The relationship is strongly influenced by the skew though, which suggests that the nature of the individual relations, but not the interaction between them, dictates how well sketching techniques behave.

The relationship between sketches in Figure 16, 17, and 18 also indicates that F-AGMS sketches essentially work as well as AGMS and FC for small skew and just slightly worse than CM for large skew. It seems that F-AGMS sketches combine in an ideal way the benefits of AGMS sketches and hashes and give good performance throughout the spectrum of problems without the need to determine the skew of the data. While CM sketches have better performance for large skew, their use seems riskier since their performance outside this regime is poor and their accuracy cannot be predicted precisely for large skew. It seems that, unless extremely precise information about the data is available, F-AGMS sketches are the safe choice.

7. CONCLUSIONS

In this paper we studied the four basic sketching techniques proposed in the literature, AGMS, Fast-AGMS, Fast-Count, and Count-Min, from both a statistical and empirical point of view. Our study complements and refines the theoretical results known about these sketches. The analysis reveals that Fast-AGMS and Count-Min sketches have

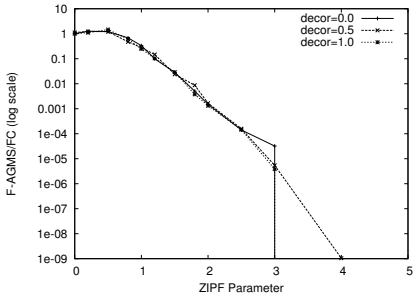

Figure 16: F-AGMS vs FC

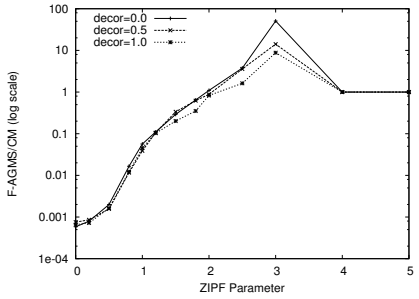

Figure 17: F-AGMS vs CM

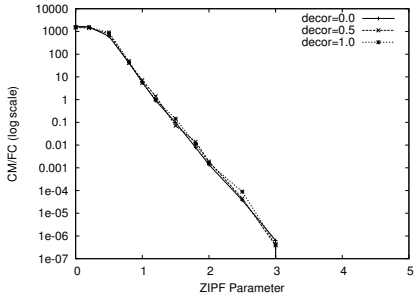

Figure 18: CM vs FC

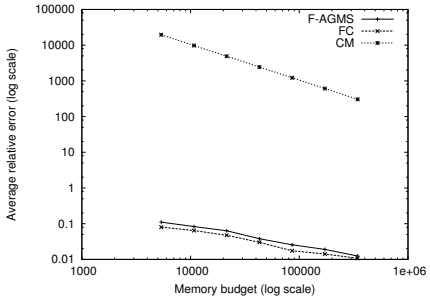

Figure 19: Memory unpeaked

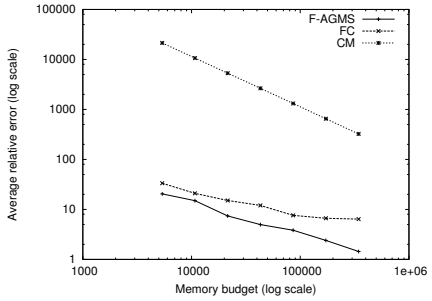

Figure 20: Memory peaked

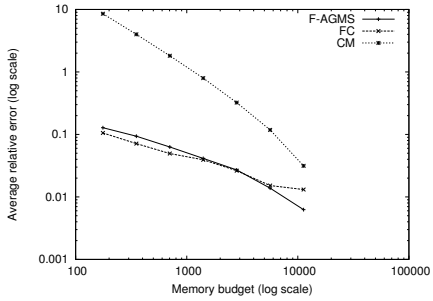

Figure 21: Census

much better performance than the theoretical prediction for skewed data, by a factor as much as $10^6 - 10^8$ for large skew. Overall, the analysis indicates strongly that Fast-AGMS sketches should be the preferred sketching technique since it has consistently good performance throughout the spectrum of problems. The success of the statistical analysis we performed indicates that, especially for estimators that use minimum or median, such analysis gives insights that are easily missed by classical theoretical analysis. Given the good performance, the small update time, and the fact that they have tight error guarantees, Fast-AGMS sketches are appealing as a practical basic approximation technique that is well suited for data-stream processing.

8. REFERENCES

[1] N. Alon, P. B. Gibbons, Y. Matias, and M. Szegedy. Tracking join and self-join sizes in limited storage. *J. Comput. Syst. Sci.*, 64(3):719–747, 2002.

[2] N. Alon, Y. Matias, and M. Szegedy. The space complexity of approximating the frequency moments. In *STOC 1996*, pages 20–29.

[3] K. P. Balanda and H. L. MacGillivray. Kurtosis: A critical review. *J. American Statistician*, 42(2):111–119, 1988.

[4] M. Charikar, K. Chen, and M. Farach-Colton. Finding frequent items in data streams. In *ICALP 2002*, pages 693–703.

[5] G. Cormode and M. Garofalakis. Sketching streams through the net: distributed approximate query tracking. In *VLDB 2005*, pages 13–24.

[6] G. Cormode and S. Muthukrishnan. An improved data stream summary: the count-min sketch and its applications. *J. Algorithms*, 55(1):58–75, 2005.

[7] Current Population Survey (CPS). `http://www.census.gov/cps`.

[8] A. Das, J. Gehrke, and M. Riedewald. Approximation techniques for spatial data. In *SIGMOD 2004*, pages 695–706.

[9] C. Estan and J. F. Naughton. End-biased samples for join cardinality estimation. In *ICDE 2006*, pages 20–31.

[10] S. Ganguly, M. Garofalakis, and R. Rastogi. Processing data-stream join aggregates using skimmed sketches. In *EDBT 2004*, pages 569–586.

[11] S. Ganguly, D. Kesh, and C. Saha. Practical algorithms for tracking database join sizes. In *FSTTCS 2005*, pages 297–309.

[12] P. J. Haas and J.M. Hellerstein. Ripple joins for online aggregation. In *SIGMOD 1999*, pages 287–298.

[13] D. Kempe, A. Dobra, and J. Gehrke. Computing aggregate information using gossip. In *FOCS 2003*.

[14] MassDAL. `http://www.cs.rutgers.edu/~muthu/massdal.html`.

[15] D. J. Olive. A simple confidence interval for the median. Manuscript, 2005.

[16] F. Pennecchi and L. Callegaro. Between the mean and the median: the L_p estimator. *Metrologia*, 43(3):213–219, 2006.

[17] R. M. Price and D. G. Bonett. Estimating the variance of the sample median. *J. Statistical Computation and Simulation*, 68(3):295–305, 2001.

[18] F. Rusu and A. Dobra. Fast range-summable random variables for efficient aggregate estimation. In *SIGMOD 2006*, pages 193–204.

[19] J. Shao. *Mathematical Statistics*. Springer-Verlag, 1999.

[20] M. Thorup and Y. Zhang. Tabulation based 4-universal hashing with applications to second moment estimation. In *SODA 2004*, pages 615–624.