

Sketches for Size of Join Estimation

FLORIN RUSU and ALIN DOBRA

University of Florida, Gainesville

15

Sketching techniques provide approximate answers to aggregate queries both for data-streaming and 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 article 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. We apply the insights obtained from the statistical study and the experimental results to design effective algorithms for sketching interval data. We show how the two basic methods for sketching interval data, DMAP and fast range-summation, can be improved significantly with respect to the update time without a significant loss in accuracy. The gain in update time can be as large as two orders of magnitude, thus making the improved methods practical. The empirical study suggests that DMAP is preferable when update time is the critical requirement and fast range-summation is desirable for better accuracy.

Categories and Subject Descriptors: H.2.4 [Database Management]: Systems—*Query processing*; G.3 [Mathematics of Computing]: Probability and Statistics—*Distribution functions*

General Terms: Algorithms, Experimentation, Measurement, Performance

Additional Key Words and Phrases: Size of join estimation, AGMS sketches, Fast-AGMS sketches, Fast-Count sketches, Count-Min sketches, DMAP, fast range-summation

This is an extended version of the paper “Statistical Analysis of Sketch Estimators” that was published in the *Proceedings of ACM SIGMOD 2007 Conference*.

Material in this article is based upon work supported by the National Science Foundation under grant number NSF-CAREER-IIS-0448264.

Authors’ addresses: F. Rusu; email: frusu@cise.ufl.edu; Alin Dobra; email: adobra@cise.ufl.edu.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or direct commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.
© 2008 ACM 0362-5915/2008/08-ART15 \$5.00 DOI 10.1145/1386118.1386121 <http://doi.acm.org/10.1145/1386118.1386121>

ACM Reference Format:

Rusu, F. and Dobra, A. 2008. Sketches for size of join estimation. *ACM Trans. Datab. Syst.* 33, 3, Article 15 (August 2008), 46 pages. DOI = 10.1145/1386118.1386121 <http://doi.acm.org/10.1145/1386118.1386121>

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 Cormode and Garofalakis [2005] and Kempe et al. [2003], 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 points of view.

Sketches can be used as the actual approximation technique or as the basic block in more complex techniques such as *skimmed sketches* [Ganguly et al. 2004] and *red-sketches* [Ganguly et al. 2005]. When used as the actual estimator, the sketch summarizes the entire data set. For the complex schemes, the sketch summarizes only the *least frequent* parts of the data set, while the *frequent* items are treated separately. This is necessary because the identity of the individual data elements is lost through sketching. Consequently, a good understanding of the basic sketching techniques is required both for identifying the properties of the different schemes as well as for determining what type of basic sketches to use as part of *skimmed sketches* [Ganguly et al. 2004] or *red-sketches* [Ganguly et al. 2005]. For either application, it is important to understand as well as possible the 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 [Haas and Hellerstein 1999]—theoretical approximation guarantees in the form of confidence bounds were provided for all types of sketches from the beginning [Alon et al. 1996]. 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 parametrized 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 perfectly justified, is that some of the theoretical bounds are too conservative.

In this article, 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 [Alon et al. 1996, 2002], Fast-AGMS [Cormode and Garofalakis 2005], Count-Min [Cormode and Muthukrishnan 2005a], and Fast-Count [Thorup and Zhang 2004] 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 too conservative (6 to 10 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 does not exist any detailed statistical study of sketching techniques and only limited empirical studies to assess 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 Cormode and Garofalakis [2005], but revealed by our detailed statistical analysis. While Cormode and Garofalakis [2005] provide only a big- \mathcal{O} notation analysis of Fast-AGMS sketches, showing that they have the same accuracy as the original AGMS sketches, we show that Fast-AGMS sketches have a completely different statistical behavior that produces significantly improved accuracy. 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.

We apply the results obtained from the statistical study to design effective algorithms for sketching interval data. Sketches over interval data can be useful by themselves, but they are also a building block in solutions to more complex problems like the size of spatial joins [An et al. 2001; Das et al. 2004]. DMAP and fast range-summation [Rusu and Dobra 2007; Das et al. 2004] are the existing solutions for sketching interval data. They are both inefficient when compared to hash-based sketches because of the use of AGMS sketches which have higher update time. In this article we study how DMAP and fast range-summation can use the more efficient hash-based sketches. In particular, we show that only DMAP can be extended to other types of sketches and, thus a significant improvement in update time can be gained by a simple

¹This is the common practice for sampling estimators [Haas and Hellerstein 1999].

replacement of the underlying sketching technique. To improve the accuracy of DMAP, significantly inferior to that of the fast range-summation method, we make use of a simple modification that keeps exact counts for some of the frequencies. We call this modification DMAP COUNTS. We also introduce a method to improve the update performance of fast range-summation AGMS sketches based on a simple equi-width partitioning of the domain. The experimental results show that these derived methods keep the advantage of their base methods, while significantly improving their drawbacks, to the point where they are efficient both in accuracy and update time.

Summarizing, our detailed contributions are

- We perform a statistical analysis of the basic sketch estimators proposed in the literature. Our goal is to improve the distribution-independent confidence bounds that are off by 6 to 10 orders of magnitude in some cases (for Fast-AGMS and Count-Min sketches). The main result of this statistical study is the much tighter distribution-dependent confidence bounds we derive for Fast-AGMS sketches. Although identical according to the distribution-independent bounds, AGMS and Fast-AGMS sketches have a completely different statistical behavior. Our contribution is to identify this significant discrepancy and to provide the statistical explanation based on higher frequency moments.
- We perform the first extensive empirical study designed to assess the performance of the proposed sketch estimators based on a large variety of parameters including data skew, data correlation, memory usage, and update time. The main result of this empirical study is to identify Fast-AGMS sketches as the sketching method with really good results, irrespective of the experimental setup. This result is surprising because, although designed to be faster than the original AGMS sketches, Fast-AGMS sketches are expected to have the same accuracy performance as AGMS sketches. Our statistical study provides sufficient insights to explain this discrepancy.
- We design effective algorithms for sketching interval data. The existing solutions are either designed only for AGMS sketches or have poor accuracy. We propose a fast range-summation algorithm that extends to hash-based sketches. We also show that a simple heuristic improves the accuracy of DMAP significantly. Our experimental study shows that the resulting algorithms are efficient both in accuracy and update time.

A significantly smaller version of this article appeared in the proceedings of ACM SIGMOD 2007 conference under the name *Statistical Analysis of Sketch Estimators*. The exposition of this manuscript is significantly changed to ensure easier understanding. The major pieces of added material are the following:

- A theoretical treatment of *confidence bounds* that is key for the development of the statistical analysis we provide (Sections 2.1 and 2.2).
- A discussion on confidence bounds for the minimum estimator (Section 2.7).
- A complete new section on sketches for interval data that complements the

original sketching techniques applicable to point relations. The results in this section are derived as a natural application of the statistical analysis and the empirical results we provide for the basic sketch estimators (Section 6).

1.1 Problem Formulation

Let $S = (e_1, w_1), (e_2, w_2), \dots, (e_s, w_s)$ be a data stream, where the keys e_i are members of the set $I = \{0, 1, \dots, N-1\}$ and w_i represent frequencies. The frequency vector $\vec{f} = [f_0, f_1, \dots, f_{N-1}]$ 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 \vec{x}_f [Cormode and Garofalakis 2005] 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 \vec{f} and \vec{g} , $\vec{f} \odot \vec{g} = \sum_{i=0}^{N-1} f_i g_i$. As shown in Rusu and Dobra [2007], 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, that is, $\alpha = \beta$. Also, the second frequency moment or the self-join size of S is nothing else than the inner-product $\vec{f} \odot \vec{f}$.

1.2 Outline

The rest of the article is organized as follows. In Section 2 we provide important results from measure theory and statistics that are needed throughout the article. 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 apply the results of the statistical and empirical studies to design efficient algorithms for sketching interval data. We conclude in Section 7.

2. CONFIDENCE BOUNDS

The abstract problem we study throughout the paper is the following. Given X_1, \dots, X_n independent instances of a generic random variable X , define an estimator for the expected value $E[X]$ and provide confidence bounds for the estimate. While $E[X]$ is the convergence value of the estimator (hopefully the true value of the estimated quantity), confidence bounds provide information about the interval where the expected value lies with high probability or, equivalently, the probability that a particular instance of X deviates by a given amount from the expectation $E[X]$.

In this section we provide an overview of the methods to derive confidence bounds for generic random variables in general, and sketches in

particular. There exist two types of confidence bounds: distribution-independent and distribution-dependent. Distribution-independent confidence bounds are derived from general notions in measure theory and are mainly used in theoretical computer science. Distribution-dependent confidence bounds assume certain distributions for the estimator of the expectation $E[X]$ and are largely used in statistics. While distribution-independent bounds are based on general inequalities, a detailed problem-specific analysis is required for distribution-dependent bounds. In the context of sketch estimators, we show that distribution-independent bounds, although easier to obtain, are unacceptably loose in some situations, thus making it necessary to derive tighter distribution-dependent bounds.

2.1 Distribution-Independent Confidence Bounds

As already specified, distribution-independent confidence bounds are derived from general inequalities on tail probabilities in measure theory. No assumption on the probability distribution of the estimator is made. The general inequalities used for characterizing sketching techniques are Markov inequality, Chebyshev inequality, and Chernoff bound [Motwani and Raghavan 1995]. The Markov inequality states that the probability that a random variable deviates by a factor larger than t from its expected value is smaller than $\frac{1}{t}$. This is the tightest bound that can be obtained when only the expectation $E[X]$ is known. Tighter confidence bounds can be derived using Chebyshev inequality and its extension to higher moments. The larger the number of moments computed, the tighter the confidence bounds. Unfortunately, the computation of higher moments demands larger degrees of independence between the instances of X , and it cannot always be carried out exactly. Moreover, the improvement gained by computing higher moments is usually a constant factor which is asymptotically insignificant. Thus, distribution-independent confidence bounds are mostly expressed in terms of the first two frequency moments (expectation and variance) using Markov and Chebyshev inequalities. Chernoff bounds are exponential tail bounds applicable to sums of independent Poisson trials. They are largely used for the analysis and design of randomized algorithms (including sketches) because of the tight bounds (logarithmic) they provide.

2.2 Distribution-Dependent Confidence Bounds

In order to compute distribution-dependent confidence bounds, a parametric distribution is assumed for the estimator of $E[X]$. Then confidence bounds are derived from the cumulative distribution function (cdf) of the assumed distribution. The parameters of the considered distribution are generally computed from the frequency moments of X , that is, a number of moments equal with the number of parameters have to be computed. Since a large number of distributions have only two parameters (e.g., Normal, Gamma, Beta, etc.), only the expectation and the variance of X have to be determined. Notice that although both types of confidence bounds require the computation of the frequency moments of X , the actual bounds are extracted in different ways. The question that immediately arises is which confidence bounds should be used. Typically,

distribution-dependent bounds are tighter, but there exist assumptions that need to be satisfied in order for them to hold. Specifically, the distribution of the estimator for $E[X]$ has to be similar in shape with the assumed parametric distribution. In the following we provide a short overview of the results from statistics on distribution-dependent confidence bounds that are used throughout the article.

2.3 Mean Estimator

Usually the *mean* \bar{X} of X_1, \dots, X_n is considered as the proper estimator for $E[X]$. It is known from statistics [Shao 1999] 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 of X is normal or almost normal. *Central Limit theorem* (CLT) extends the characterization of the mean estimator to arbitrary distributions of the random variable X .

THEOREM 1 MEAN CLT [SHAO 1999]. *Let X_1, \dots, X_n be independent samples drawn from the distribution of the random variable X and \bar{X} be the average of the n samples. Then, as long as $\text{Var}[X] < \infty$:*

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

Essentially, the Central Limit theorem (CLT) states that the distribution of the mean is asymptotically a normal distribution centered on the expected value and having variance $\frac{\text{Var}[X]}{n}$ irrespective of the distribution of X . Confidence bounds for \bar{X} can be immediately derived from the cdf of the normal distribution:

THEOREM 2 MEAN BOUNDS [SACHS 1984]. *For the same setup as in Theorem 1, the asymptotic confidence bounds for \bar{X} are*

$$P\left[|\bar{X} - E[X]| > z_{\alpha/2} \sqrt{\frac{\text{Var}[X]}{n}}\right] < \alpha,$$

where z_β is the β quantile ($\beta \in [0, 1]$) of the normal $N(0, 1)$ distribution, i.e., the point for which the probability of $N(0, 1)$ to be smaller than the point is β .

Since fast series algorithms for the computation of z_β are widely available,² the computation of confidence bounds for \bar{X} is straightforward. Usually, the CLT approximation of the distribution of the mean and the confidence bounds produced with it are correct starting with hundreds of samples being averaged. If the number of samples is smaller, confidence bounds can be determined based on the Student t-distribution [Sachs 1984]. The only difference is that the β quantile $t_{n-1, \beta}$ of the Student t-distribution with $n - 1$ degrees of freedom has to be used instead of the β quantile z_β of the normal distribution in Theorem 2.

²The GNU Scientific Library (GSL) implements pdf, cdf, inverse cdf, and other functions for the most popular distributions, including Normal and Student t.

Notice that in order to characterize the mean estimator, only the variance of X has to be determined. When $\text{Var}[X]$ 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).

2.4 Median Estimator

Although the mean is the preferable estimator in most circumstances, there exist distributions for which the mean cannot be used as an estimator of $E[X]$. For Cauchy distributions (which have infinite variance) the mean can be shown to have the same distribution as a single random sample. In such cases the *median* \tilde{X} of the samples is the only viable estimator of the expected value. The necessary condition for the median to be an estimator of the expected value is that the distribution of the estimator be symmetric, in which case the mean and the median coincide. We start the investigation of the median estimator by introducing its corresponding Central Limit theorem and then show how to derive confidence bounds.

THEOREM 3 MEDIAN CLT [SHAO 1999]. *Let X_1, \dots, X_n be independent samples drawn from the distribution of the random variable X , and \tilde{X} be the median of the n samples. Then, as long as the probability 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.

Median CLT states that the distribution of the median is asymptotically a normal distribution centered on the true median and having the variance equal to $\frac{1}{4n \cdot f(\theta)^2}$. In order to compute the variance of this normal distribution and derive confidence bounds from it, the probability density function (pdf) of X has to be determined, or at least estimated, at the true median θ . Since $f(\theta)$ cannot be computed exactly in general, multiple estimators for the variance are proposed in the statistics literature [Price and Bonett 2001]. We use the variance estimator proposed in Olive [2005] for deriving confidence bounds:

THEOREM 4 MEDIAN BOUNDS [OLIVE 2005]. *For the same setup as in Theorem 3, the confidence bounds for \tilde{X} are given by*

$$P[|\tilde{X} - \theta| \leq t_{n-1, 1-\alpha/2} \text{SE}(\tilde{X})] \geq 1 - \alpha,$$

where $t_{n-1, \beta}$ is the β quantile of the Student t-distribution with $n - 1$ degrees of freedom and $\text{SE}(\tilde{X})$ is the estimate for the standard deviation of \tilde{X} given by

$$\begin{aligned} \text{SE}(\tilde{X}) &= \frac{X_{(U_n)} - X_{(L_n+1)}}{2} \\ L_n &= \left\lfloor \frac{n}{2} \right\rfloor - \left\lceil \sqrt{\frac{n}{4}} \right\rceil \\ U_n &= n - L_n. \end{aligned}$$

Notice that while the distribution corresponding to the mean estimator is centered on the expected value $E[X]$, the distribution of the median is centered on the true median, thus the requirement on the symmetry of the distribution for the median to be an estimator of $E[X]$.

2.5 Mean vs Median

For the cases when the distribution is symmetric, and 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 mean or the 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 $\text{Var}[X]$ and the quantity $\frac{1}{4f(\theta)^2}$. The relation between these two quantities is established in statistics through the notion of *asymptotic relative efficiency*:

Definition 1 [Shao 1999]. 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 probability density function f , is defined as

$$e(f) = 4f(\theta)^2 \text{Var}[X].$$

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

An important case to consider is when X has normal distribution. In this situation the efficiency is independent of the parameters of the distribution and it is equal to $\frac{2}{\pi} \approx 0.64$ (derived from Definition 1 and the pdf of the normal distribution). This immediately implies that when the estimator is defined as the average of the samples, that is, by Mean CLT the distribution of the estimator 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.

As already pointed out, when the efficiency is supra-unitary, that is, $e(f) > 1$, 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 easier to compute for discrete distributions—indicates supra-unitary efficiency. According to the statistics literature [Balanda and MacGillivray 1988], *kurtosis* is a good indicator of supra-unitary efficiency.

Definition 2 [Balanda and Macgillivray 1988]. The kurtosis k of the distribution of the random variable X is defined as

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

For normal distributions, the kurtosis is equal to 3 irrespective of the parameters. Even though there does not exist a distribution-independent relationship between the kurtosis and the efficiency, empirical studies [Penneccchi and Callegaro 2006] show that whenever $k \leq 6$ the mean is a better estimator of $E[X]$, while for $k > 6$ the median is the better estimator.

2.6 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 with sketching techniques (see Section 3) is to group the samples into groups of equal size, compute the mean of each group, 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:

Theorem 5. [ALON ET AL. 1996]. *The median Y of $2 \ln(\frac{1}{\alpha})$ means, each averaging $\frac{16}{\epsilon^2}$ independent samples of the random variable X , has the property*

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

We provide an example that compares the bounds for the median of means estimator. While distribution-independent bounds are computed using the results in Theorem 5, distribution-dependent bounds are computed through a combination of Mean CLT, Median CLT, and efficiency.

Example 1. Suppose that we want to compute 95% confidence bounds for the median of means estimator. Then the number of means for which we compute the median should be $2 \ln \frac{1}{0.05} = 2 \ln 20 \approx 9$ according to Theorem 5. If the number of samples is n , then each mean is the average of $\frac{n}{9}$ samples, thus $\epsilon = \sqrt{\frac{144}{n}} = 12 \cdot \sqrt{\frac{1}{n}}$. The width of the confidence bound in terms of $\sqrt{\frac{\text{Var}[X]}{n}}$ is thus 12.

By applying Mean CLT, the distribution of each mean is asymptotically normal with variance $\frac{\text{Var}[X]}{n/9}$. In practice, confidence bounds can be easily derived by applying the results in Theorem 4. We cannot do that in this example because the values of the 9 means are unknown. Instead we assume that the distribution of the means is asymptotically normal and, by Median CLT and the definition of efficiency, the median of the 9 means has variance $\frac{1}{9e(N)} \cdot \frac{\text{Var}[X]}{n/9}$, with $e(N) = \frac{2}{\pi}$ the efficiency of the normal distribution. The variance of Y is thus $\frac{\pi}{2} \cdot \frac{\text{Var}[X]}{n} \approx 1.57 \cdot \frac{\text{Var}[X]}{n}$. With this, the width of the CLT-based confidence bound for Y with respect to $\sqrt{\frac{\text{Var}[X]}{n}}$ is $\sqrt{1.57} \cdot 1.96 = 2.45$ (1.96 is the 95% quantile), which is $\frac{12}{2.45} \approx 4.89$ times smaller than the distribution-independent confidence bound.

This result confirms that distribution-dependent confidence bounds are tighter and is consistent with other results that compare the two types of bounds [Sachs 1984]. Confidence bounds of different widths can be computed in a similar manner, the only difference being the values that are plugged into

the formulas. For example, the ratio for the 99% confidence bound is 4.64 and 4.34 for the 99.9% confidence interval.

An important point in the above derivation of the CLT confidence bounds for Y is the fact that the confidence interval is wider by $\sqrt{\frac{\pi}{2}} \approx 1.25$ 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 as long as the distribution of Y is (asymptotically) normal. A simple explanation for this is that the asymptotic regime of Mean CLT starts to take effect (the distribution becomes normal), since means are computed first and the mean estimator is more efficient than the median estimator. Thus, from a practical and statistical point of view based on efficiency, if the distribution of the basic random variable X is symmetric the estimator should be either the mean ($e < 1$), or the median ($e > 1$). The combined median over means estimator is recommended whenever the distribution of X is not symmetric or when the number of means is not large enough to make the asymptotic behavior of Mean CLT effective.

2.7 Minimum Estimator

Although the minimum of the samples X_1, \dots, X_n is not an estimator for the expectation $E[X]$, a discussion on the behavior of the minimum estimator is included because of its relation to Count-Min sketches. It is known from statistics [Coles 2001] that the minimum of a set of samples has an asymptotic distribution called the *generalized extreme value distribution* (GEV), independent of the distribution of X . The parameters of the GEV distribution can be computed from the frequency moments of X , thus confidence bounds for the minimum estimator can be derived from the cdf of GEV. Although this is a straightforward method to characterize the behavior of the minimum estimator, we will see that it is not applicable to Count-Min sketches (Section 4).

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. All sketching techniques generate multiple random instances of an elementary sketch estimator, instances that are effectively random samples of the elementary estimator. While the random instances of the elementary sketch estimator are samples of the estimator, these samples should not be confused with the samples from the underlying data used by the sampling techniques [Haas and Hellerstein 1999]. The samples of the elementary sketch estimator are grouped and combined in different ways in order to obtain the overall estimate. Typically, in order to produce an elementary sketch estimator—the process is duplicated for each sample of the elementary sketch estimator—multiple counters corresponding to *random variables* with required properties are maintained. The collection of counters for all the samples of the elementary estimator is called the *sketch*. The existing sketching techniques differ in how the random variables are organized, thus

the update procedure, the way the elementary sketch estimator is computed, and how the answer to a given query is computed by combining the elementary sketch estimators. In this section we provide an overview of the existing sketching techniques used for approximating the size of join of two data streams (see Section 1.1). For each technique we specify the elementary sketch estimator, denoted by X , possibly with a subscript indicating the type of sketch, and the way the elementary sketches are combined to obtain the final estimate Z .

3.1 AGMS Sketches

The i th entry of the size n AGMS (or, *tug-of-war*) [Alon et al. 1996, 2002] 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 [Rusu and Dobra 2007]. 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 \vec{f} and \vec{g} , that is, $E[X[i]] = \vec{f} \odot \vec{g}$. The variance of the estimator is

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

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 , that is $\text{Var}[Y] = \frac{\text{Var}[X[i]]}{n}$, thus improving the estimation error. In order to make the estimation more stable, the original solution [Alon et al. 1996] returned as the result the median of m Y estimators, that is $Z = \text{Median}_{1 \leq k \leq m} Y[k]$. We provide an example to illustrate how AGMS sketches work.

Example 2. Consider two data streams F and G given as pairs (key, frequency):

$$\begin{aligned} F &= \{(1, 5), (4, -2), (1, 2), (2, 3), (3, 1), (1, -3), (3, 2), (5, 2), (4, 3)\} \\ G &= \{(2, 1), (4, 3), (3, 2), (1, 3), (3, -2), (1, 2), (5, -1), (1, 2), (4, -1)\} \end{aligned}$$

We want to estimate the size of join $|F \bowtie G|$ of the two streams using sketches consisting of 3 counters. A family of 4-wise independent ± 1 random variables corresponds to each counter. Let the mappings from the key domain $\{1, 2, 3, 4, 5\}$ in this case) to ± 1 to be given as in Table I.

As the data is streaming by, all the counters in the corresponding sketch vector are updated. For example, the pair $(1, 5)$ in F updates the counters in x_f as follows: $x_f[1] = 5$, $x_f[2] = 5$, and $x_f[3] = -5$ (the counters are initialized to 0), while after the pair $(4, -2)$ is processed, the counters have the following values: $x_f[1] = 7$, $x_f[2] = 3$, and $x_f[3] = -7$. After all the elements in the two

Table I. Mappings for ± 1 Random Variables

Counter	Key domain				
	1	2	3	4	5
1	+1	+1	+1	-1	-1
2	+1	-1	-1	+1	+1
3	-1	+1	-1	+1	-1

streams passed by, the two sketch vectors are $x_f = [7, 1, -5]$ and $x_g = [7, 7, -3]$. The estimator X for the size of join $|F \bowtie G|$ consists of the values $\bar{X} = [49, 7, 15]$ having the mean $Y = 23.66$. The correct result is 31. Multiple instances of Y can be obtained if other groups of 3 counters and their associated families of ± 1 random variables are added to the sketch. In this case the median of the instances of Y is returned as the final result.

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, that is, $n = \mathcal{O}(\frac{1}{\epsilon^2})$ and $m = \mathcal{O}(\log \frac{1}{\delta})$.

THEOREM 6 [ALON ET AL. 2002]. *Let \bar{x}_f and \bar{x}_g denote two parallel sketches comprising $\mathcal{O}(\frac{1}{\epsilon^2} \log \frac{1}{\delta})$ 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 $\mathcal{O}(\frac{1}{\epsilon^2} \log \frac{1}{\delta})$ 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. From the perspective of the abstract problem in Section 2, $X[i]$ represent the primitive instances of the generic random variable X . Median of means Z is the estimator for the expected value $E[X]$. The distribution-independent confidence bounds in Theorem 6 are derived from Theorem 5.

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 [Cormode and Garofalakis 2005], as a refinement of Count sketches proposed in Charikar et al. [2002] 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 $h : I \rightarrow \{1, \dots, n\}$. The role of the hash function is to scatter the keys in the data stream to different counters in the sketch vector, thus reducing

Table II. Hash Functions

Row	Key domain				
	1	2	3	4	5
1	1	1	3	2	3
2	1	3	2	1	2
3	3	2	3	3	1

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 , that is, $x_f[h(e)] = x_f[h(e)] + w \cdot \xi(e)$.

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 , that is, $Z = \text{Median}_{1 \leq k \leq m} Y[k]$. In light of Section 2, Y corresponds to the basic instances while the median is the estimator for the expected value. We provide a simple example to illustrate the Fast-AGMS sketch data structure.

Example 3. Consider the same data streams from Example 2. The sketch vector consists of 9 counters grouped into 3 rows of 3 counters each. The same families of ± 1 random variables (Example 2) are used, but a family corresponds to a row of counters instead of only one counter. An additional family of 2-universal hash functions, corresponding to the rows of the sketch, maps the elements in the key domain to only one counter in each row. The hash functions are specified in Table II.

For each stream element only one counter from each row is updated. For example, after the pair $(2, 3)$ in F is processed, the sketch vector x_f looks like $x_f[1] = [10, 2, 0]$, $x_f[2] = [5, 0, -3]$, and $x_f[3] = [0, 3, -9]$ (the counters are initialized to 0). After processing all the elements in the two streams, the two sketch vectors are $x_f = [[7, -1, 1], [5, -1, -3], [-2, 3, -6]]$ and $x_g = [[8, -2, 1], [9, -1, -1], [1, 1, -5]]$. The estimator for the size of join $|F \bowtie G|$ consists of the median of $Y = [59, 49, 31]$, which is 49, while the correct result is still 31.

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 7 [CORMODE AND GAROFALAKIS 2005]. *Let n be defined as $n = \mathcal{O}(\frac{1}{\epsilon^2})$ and m as $m = \mathcal{O}(\log \frac{1}{\delta})$. 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). Fast-AGMS sketches have the same accuracy guarantees as basic AGMS sketches because the variance of the basic estimator is the same for both schemes (see Eq. (1)). Moreover, notice that only the sketch vector size is dependent on the error ϵ .

At the first look, it may seem counter-intuitive that updating only one counter (Fast-AGMS) instead of all (AGMS) does not negatively affect the accuracy of the estimator. Since nothing comes for free, the cost of this gain in speed is an additional random process represented by hashing. Essentially, the effect of random hashing is somehow similar to the sketch-partitioning scheme in Dobra et al. [2002]. There exist cases when random hashing generates a small variance and cases for which the variance is significantly larger than the variance of AGMS sketches. Overall, the expected value of the variance due to hashing is identical to the variance of averaged AGMS estimators.

3.3 Fast-Count Sketches

Fast-Count sketches, introduced in Thorup and Zhang [2004], provide the error guarantees and the update time of Fast-AGMS sketches, while requiring only one underlying random process—hashing. The tradeoffs 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, that is, $x_f[h(e)] = x_f[h(e)] + w$, where $h : I \rightarrow \{1, \dots, n\}$ 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 Thorup and Zhang [2004]):

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

The complicated form of Y is due to the bias 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 $\tilde{f} \odot \tilde{g}$. The variance of the Fast-Count estimator is identical to the variance of the Y estimator for AGMS (Fast-AGMS) sketches in Eq. (1) if the Fast-Count sketch structure contains one extra counter. 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 guarantees in Theorem 7 apply. Notice that in practice multiple instances of Y are computed and the final estimator for the expected value of the size of join is the mean (average) of these instances. We provide an example that shows how Fast-Count sketches work.

Example 4. Consider the same data streams from Example 2. The sketch vector consists of 9 counters grouped into 3 rows of 3 counters each. The same hash functions as in Example 3 are used (suppose that they are 4-universal). For each stream element only one counter from each row is updated. For example, after the pair $(2, 3)$ in F is processed, the sketch vector x_f looks like $x_f[1] = [10, -2, 0]$, $x_f[2] = [5, 0, 3]$, and $x_f[3] = [0, 3, 5]$ (the counters are initialized

to 0). After processing all the elements in the two streams, the two sketch vectors are $x_f = [[7, 1, 5], [5, 5, 3], [2, 3, 8]]$ and $x_g = [[8, 2, -1], [9, -1, 1], [-1, 1, 9]]$. The estimator for the size of join $|F \bowtie G|$ consists of the vector $Y = [21, 6, 51]$. The average 26 of the elements in Y is returned as the final estimate.

3.4 Count-Min Sketches

Count-Min sketches [Cormode and Muthukrishnan 2005a] 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 $\tilde{f} \odot \tilde{g}$. In order to minimize the over-estimated quantity, the minimum over m independent Y estimators is computed, that is, $Z = \text{Min}_{1 \leq k \leq m} Y[k]$. 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 following example illustrates the behavior of Count-Min sketches.

Example 5. For the same setup as in Example 4, exactly the same sketch vectors are obtained after updating the two streams. Only the final estimator is different. It is the minimum of $Y = [53, 43, 73]$, that is, 43.

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

THEOREM 8 [CORMODE AND MUTHUKRISHNAN 2005A] $Z \leq \tilde{f} \odot \tilde{g} + \epsilon \|\tilde{f}\|_1 \|\tilde{g}\|_1$ 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 $\mathcal{O}(\log \frac{1}{\delta})$.

$\|\tilde{f}\|_1 = \sum_{i \in I} f_i$ and $\|\tilde{g}\|_1 = \sum_{i \in I} g_i$ represent the L_1 norms of the vectors \tilde{f} and \tilde{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 $\mathcal{O}(\frac{1}{\epsilon})$).

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 [Alon et al. 1996].

Table III. 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.

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

The size of join results is considered relative 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 III summarizes the results predicted by the theory based on distribution-independent confidence bounds. Since the bounds for AGMS, Fast-AGMS, and Fast-Count sketches are identical, they have the same behavior from a theoretical perspective. For small-size 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

The goals pursued in refining the sketching techniques were to leverage the randomness and to decrease the update time while maintaining the same error guarantees as for the original AGMS sketches. As we have previously seen, all kinds of tradeoffs are involved. The main drawback of the existing theoretical results is that they characterize only the asymptotic behavior, but do not provide enough details about the behavior of the sketching techniques in practice (they ignore important details about the estimator because they are derived from distribution-independent confidence bounds). From a purely practical point of view, we are interested in sketching 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 pursued from the theoretical point of view, in theory we insist on deriving simple formulas 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 expressible formulas,

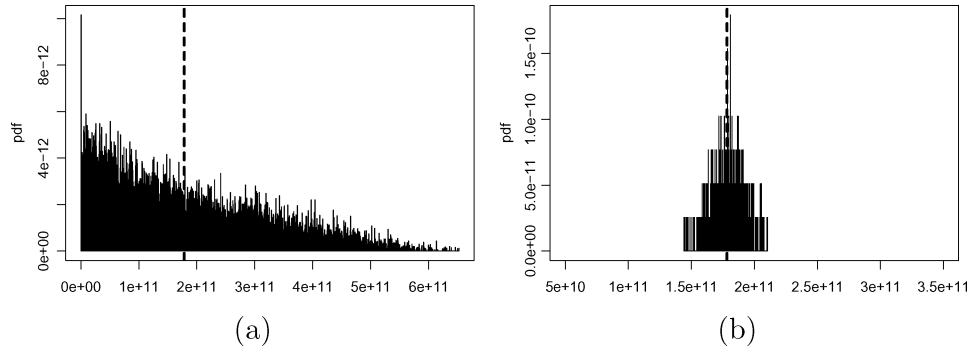


Fig. 1. The distribution of AGMS sketches for self-join size. (a) depicts the distribution of the basic AGMS sketch estimator. In (b) the distribution of the same data is plotted after grouping the basic estimators and taking their average. The x -axis corresponds to the actual value of the estimator, while the y -axis represents the experimental probability distribution. The dotted line corresponds to the true result or expected value.

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 the sketching techniques from a statistical perspective by asking the following questions that reflect the difference between the pragmatic and the theoretical points of view:

- All sketching techniques combine multiple independent instances of elementary sketches using the estimators from Section 2 (mean, median, minimum) in order to define a more accurate estimator for the expected value. We ask the question which of the estimators is more accurate for each of the four sketching techniques?
- How *tight* are the theoretical distribution-independent confidence bounds? And is it possible to derive tighter distribution-dependent confidence bounds that work in practice based on the estimator chosen in the previous question? We are not interested in tight bounds only for some situations, 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 [Haas and Hellerstein 1999].

We use a large-scale statistical analysis based on experiments in order to answer the above questions. The plots in this section have statistical significance and are not highly sensitive to the experimental setup (Section 5).

4.1 AGMS Sketches

We explore which estimator—mean, median, or minimum—to use for AGMS sketches instead of the median of means estimator proposed in the original paper [Alon et al. 1996] and if that would be advisable. In order to accomplish this task, we plotted the distribution of the basic sketch for a large spectrum of problems. Based on our experiments, Figure 1 is a representative 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 sketch is not symmetric and there exists a variable gap between the mean of the distribution and the median. This gap is not easy 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 1). As expected, the distribution appears to be normal and its expected value is the true result. As explained in Section 2.6, the mean is always preferable to the median of means as an estimator for the expected value of a random variable given as a sample. This is the case because once averaging over the sample space the distribution of the estimator starts to become normal (Mean CLT), and it is known that the mean is more efficient than the median for normal distributions (Section 2.5).

Although the median of means estimator has no statistical significance, it allows the derivation of exponentially decreasing distribution-independent confidence intervals based on a Chernoff bound. To derive tighter distribution-dependent confidence bounds based only on the mean estimator, we can use Theorem 2. 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 distribution-independent confidence bounds in Theorem 5 are wider by a factor of approximately 4 than the CLT bounds, as derived in Example 1. This discrepancy between the distribution-independent bounds and the effective error was observed experimentally in Rusu and Dobra [2007] and Das et al. [2004], but it was not explained. In conclusion, the mean estimator seems the right choice from a practical perspective, considering its advantages over the median of means estimator.

4.2 Fast-AGMS Sketches

Comparing Theorems 6 and 7 that characterize AGMS and Fast-AGMS (F-AGMS) sketches, respectively, we observe that the predicted accuracy is identical, but Fast-AGMS have significantly lower update times. This immediately indicates that F-AGMS should be preferred to AGMS. In the previous section we saw a discrepancy of a factor of approximately 4 between the distribution-independent bounds and the CLT-based bounds for AGMS sketches and the possibility of a significant improvement if the median of means estimator is replaced by means only. In this section we investigate the statistical properties of F-AGMS sketches in order to identify the most adequate estimator and possibly derive tighter distribution-dependent confidence bounds.

We start the investigation on the statistical properties of Fast-AGMS sketches with the following result on the first two frequency moments (expected value and variance) of the basic estimator:

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

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

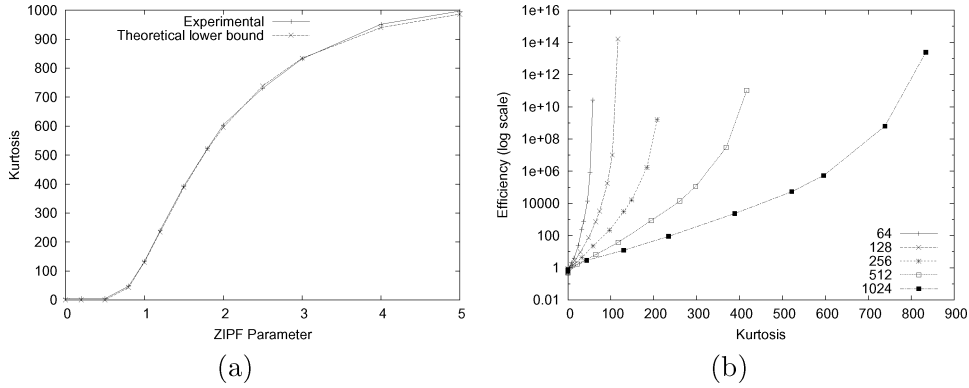


Fig. 2. F-AGMS kurtosis and efficiency. (a) depicts kurtosis as a function of the skewness of the data for self-join size. The theoretical lower bound is computed with the formula in Proposition 2. The efficiency as a function of kurtosis is plotted in (b) for sketches with various numbers of buckets in a row.

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 hashing plays the same role as averaging when it comes to reducing the variance and that the transformation on the distribution of elementary F-AGMS sketches is the same, that is, the distribution becomes normal and the variance is reduced by a factor equal to the number of buckets. The result, shown in Figure 2, on the fourth frequency moment of F-AGMS represents the first discrepancy between the distributions of Fast-AGMS and AGMS sketches:

PROPOSITION 2. *With the same setup as in Proposition 4.2, we have:*

$$\begin{aligned} \text{Var}_h[\text{Var}_\xi[X]] &= \frac{B-1}{B^2} \left[3 \left(\sum_i f_i^2 g_i^2 \right)^2 + 4 \sum_i f_i^3 g_i \sum_j f_j g_j^3 \right. \\ &\quad \left. + \sum_i f_i^4 \sum_j g_j^4 - 8 \sum_i f_i^4 g_i^4 \right]. \end{aligned}$$

In order to derive an exact closed-form formula for the fourth frequency moment the ξ family is required to be 8-wise independent. Given the practical 4-wise independence requirements for the ξ family, we are able to derive only $\text{Var}_h[\text{Var}_\xi[X]]$ which is a lower bound on the fourth moment of the estimator. We use kurtosis (the ratio between the fourth frequency moment and the square of the variance, see Section 2.5) to characterize the distribution of the Fast-AGMS basic estimator. From Figure 2, which depicts the experimental kurtosis and its lower bound in Proposition 2, we observe that when the Zipf coefficient is larger than 1 the kurtosis grows significantly, to the point that it gets around 1000 for a Zipf coefficient equal to 5. Given these values of the kurtosis, we expect that the distribution of the F-AGMS estimator to be (close to) normal for Zipf coefficients smaller than 1 (kurtosis is equal to 3 for normal distributions,

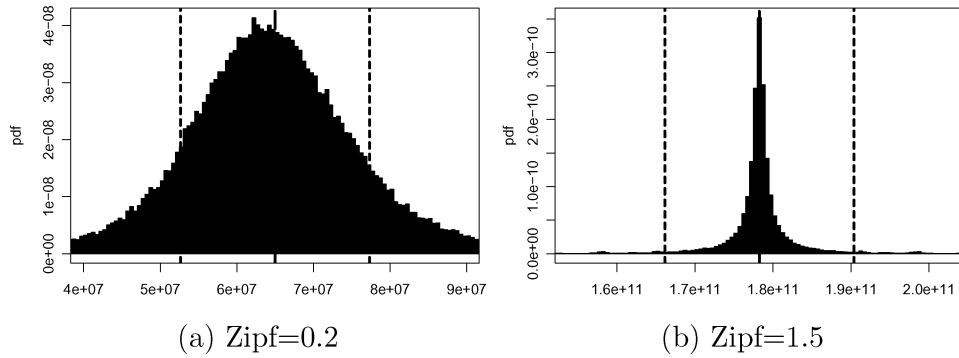


Fig. 3. The distribution of F-AGMS sketches for self-join size. The data in (a) has low skew (Zipf = 0.2), while the data in (b) is skewed (Zipf = 1.5). The red line corresponds to the true result or expected value. The blue lines are positioned one standard deviation away from the expected value.

see Section 2.5) and then to suffer a drastic change as the Zipf coefficient increases. Large kurtosis is an indicator of distributions that are more concentrated than the normal distribution, but also that have heavier tails [Balanda and MacGillivray 1988]. Indeed, Figure 3 confirms experimentally that these observations for Zipf coefficients are equal to 0.2 and 1.5, respectively. The interaction between hashing and the frequent items is an intuitive explanation for the transformation suffered by the F-AGMS distribution as a function of the Zipf coefficient. For low skew data (uniform distribution) there does not exist a significant difference between the way the frequencies are spread into the buckets by the hash function. Although there exists some variation due to the randomness of the hash function, the distribution of the estimator is normally centered on the true value. The situation is completely different for skewed data that consists of some extremely high frequencies and some other small frequencies. The impact of the hash function is dominant in this case. Whenever the high frequencies are distributed in different buckets (this happens with high probability), the estimation is extremely accurate. When at least two high frequencies are hashed into the same bucket (with small probability), the estimator is orders of magnitude away from the true result. This behavior explains perfectly the shape of the distribution for skewed data: the majority of the mass of the distribution is concentrated on the true result while some small mass is situated far away in the heavy tails. Notice that although large values of kurtosis capture this behavior, an extremely large number of experiments is required to observe the behavior in practice. For example, in Figure 2 the experimental kurtosis lies under the lower bound in some cases because the colliding events did not appear, even after 10 million experiments.

Given the different shapes of the distribution, no single estimator (mean or median, since minimum is clearly not a valid estimator for the expected value) is always optimal for Fast-AGMS sketches. While mean is optimal for low skew data since the distribution is normal (see Section 2.5), median is clearly preferable for skewed data because of the large values of kurtosis. The symmetry condition required for the median to be an estimator for the expected value

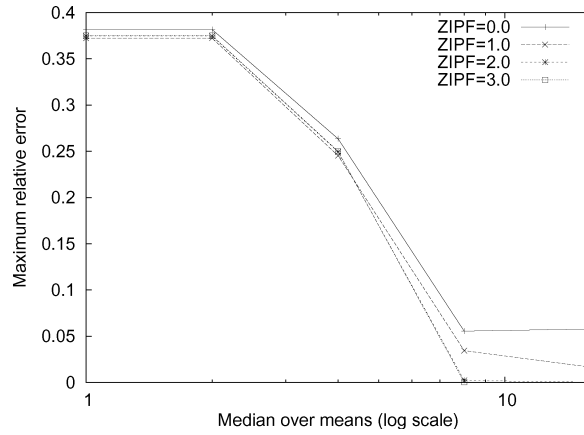


Fig. 4. Maximum relative error for F-AGMS sketches as a function of different combinations of the median and the mean estimators. The x -axis represents the number of means the median is computed over. Value 1 corresponds to taking the median over 1 mean. Value 16 corresponds to taking the median over 16 means. Since 16 is the total number of basic estimators, no mean is computed and the median is taken over all the basic estimators. For the intermediate values, a combination of the median over means estimator is used.

is satisfied because of the symmetric ± 1 random variables. In the following, we consider median as the estimator for Fast-AGMS sketches even though its error is larger by a factor of 1.25 for low skew data compared to the error of the mean. In order to quantify exactly what the gain of the median over the mean is, we use the concept of efficiency (see Section 2.5). Unfortunately, we cannot derive an analytical formula for efficiency because it depends on the value of the probability density function at the true median, which we actually try to determine. The alternative is to estimate empirically the efficiency as a function of kurtosis which, as we have already seen, is a good indicator for the distribution of the F-AGMS estimator and can be computed analytically. Figure 2 depicts the experimental efficiency as a function of kurtosis for sketches with various number of buckets. As expected, efficiency increases as the kurtosis increases, that is, as the data becomes more skewed, and gets to some extreme values in the order of 10^{10} . While efficiency is independent of the number of buckets in the sketch, the value of kurtosis is limited, with larger values corresponding to a sketch with more buckets. This implies that efficiency is not a simple function of the kurtosis and other parameters of the sketch and that the data has to be considered also. Consequently, although we could not quantify exactly what the gain of using the median instead of the mean is, the extremely large values of efficiency clearly indicate that median is the right estimator for F-AGMS sketches. In order to verify that this is the case, we plot in Figure 4 the maximum relative error obtained for different combinations of the mean and the median estimators. As expected, the error is significantly smaller when median is the chosen estimator. The error is the largest when mean is used, and it takes intermediate values for the median over means estimator.

The distribution-independent confidence bounds given by Theorem 7 are likely to be far too conservative because they are derived from the first two

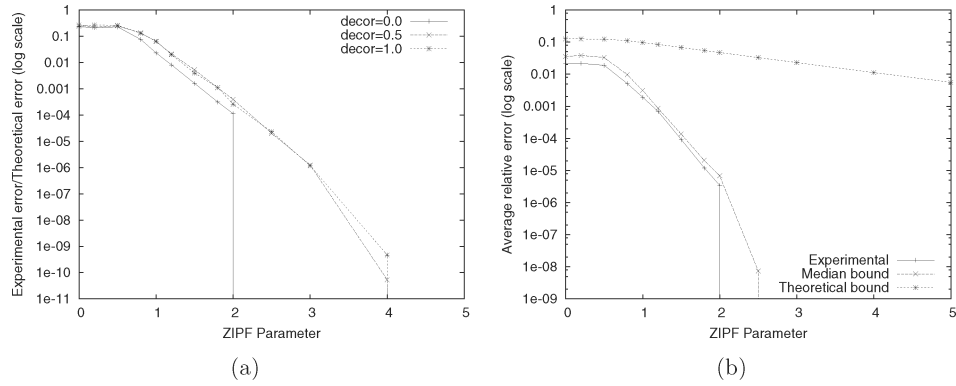


Fig. 5. Confidence bounds for F-AGMS sketches as a function of the skewness of the data. The ratio between the experimental bounds and the distribution-independent bounds for the size of join of two streams with various degrees of correlation ($\text{decor} = 0$ corresponds to streams fully correlated or self-join size, while $\text{decor} = 1$ corresponds to streams completely independent) is depicted in (a). Experimental error, distribution-independent bounds, and distribution-dependent bounds for self-join size are plotted in (b).

frequency moments using Chebyshev and Chernoff inequalities. These bounds are identical to the bounds for AGMS sketches since the two have the same expected value and variance. The significant discrepancy in the fourth moment and the shape of the distribution (Figures 1 and 3 depict the distributions for the same data) between F-AGMS and AGMS is not reflected by the distribution-independent confidence bounds. Figure 5(a) confirms the huge gap (as much as 10 orders of magnitude) that exists between the distribution-independent bounds and the experimental error. Practical distribution-dependent confidence bounds can be derived using the results in Theorem 4. A comparison between distribution-independent confidence bounds, distribution-dependent confidence bounds and the experimental error (95%) is depicted in Figure 5(b). Two important facts can be drawn from these results: first, the distribution-independent bounds are too large for large Zipf coefficients; and, second, the distribution-dependent bounds derived from Theorem 4 are always accurate. Figure 5(a) also reveals that the ratio between the actual error and the prediction is not strongly dependent on the correlation between the data for the same Zipf coefficient.

4.3 Count-Min Sketches

Based on Theorem 8, we expect Count-Min (CM) sketches to over-estimate the true value by a factor 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 first frequency moment, not the second frequency moment, and the amount of memory (number of hashing buckets), not the square root of the amount of memory. While the dependency on the first frequency moment is worse than the dependency on the square root of the second frequency moment, since the first is always larger or equal than the second, the dependency on

the amount of memory is favorable to Count-Min sketches. According to the theoretic distribution-independent confidence bounds, we expect Count-Min sketches to have weak performance for relations with low skew, but comparable performance to AGMS sketches (not much better though) for skewed relations. In this section we take a closer look at the distribution of the basic CM estimator and discuss the methods to derive confidence bounds for Count-Min sketches.

We start the study of the distribution of the elementary CM estimator with the following result that characterizes the frequency moments of the estimator:

PROPOSITION 3. *If X_{CM} is the elementary Count-Min estimator, then*

$$E[X_{CM}] = \sum_{i \in I} f_i g_i + \frac{1}{B} \left(\sum_{i \in I} f_i \sum_{j \in I} g_j - \sum_{i \in I} f_i g_i \right) \quad (2)$$

$$Var[X_{CM}] \geq \frac{1}{B} Var[X_{AGMS}]. \quad (3)$$

Equation (2) is proved in Cormode and Muthukrishnan [2005a]. The inequality in Eq. (3) becomes equality if the hash functions used are 4-universal. For 2-universal hashes, the variance increases depending on the particular generating scheme, and no simple formula can be derived. Most of the proof of Eq. (3) for 4-universal hashes is embedded in the computation of the variance for Fast-Count sketches (see Section 4.4), but the exact formula does not appear in previous work.³ The expected value of X_{CM} is always an overestimate for the true result—this is the reason why the minimum estimator is chosen. Interestingly, the variance of the estimator coincides with the variance of averages of B AGMS sketches and the variance of Fast-AGMS sketches. In order to characterize the distribution of CM sketches, we conducted an extensive statistical study based on experiments. As for Fast-AGMS sketches, the distribution of the X_{CM} estimator is highly dependent on the skewness of the data and the randomness of hashing. The fundamental difference is that the distribution is not symmetric anymore because ± 1 random variables are not used. The generic shape of the distribution has the majority of the mass concentrated to the left extremity, while the right tail is extremely long. The intuition behind this shape lies in the way hashing spreads the data into buckets: with high probability the data is evenly distributed into the buckets (this situation corresponds to the left peak), while with some extremely low probability a large number of items collide into the same bucket (this situation corresponds to the right tail). Although the shape is generic, the position of the left peak (the minimum of the distribution) depends heavily on the actual data. For low skew data the peak is far away from the true value. As the data becomes more skewed the peak starts to translate to the left, to the point where it gets to the true value. The movement towards the true value while increasing the Zipf parameter is due to the importance high frequencies start to gain. For low skew data (uniform distribution), the position of the peak is given by the average number of frequencies that are hashed into the same bucket. For skewed data dominated by some high frequencies,

³Since proving the formula is a simple matter of rewriting the equations in Thorup and Zhang [2004], we do not provide the proof here.

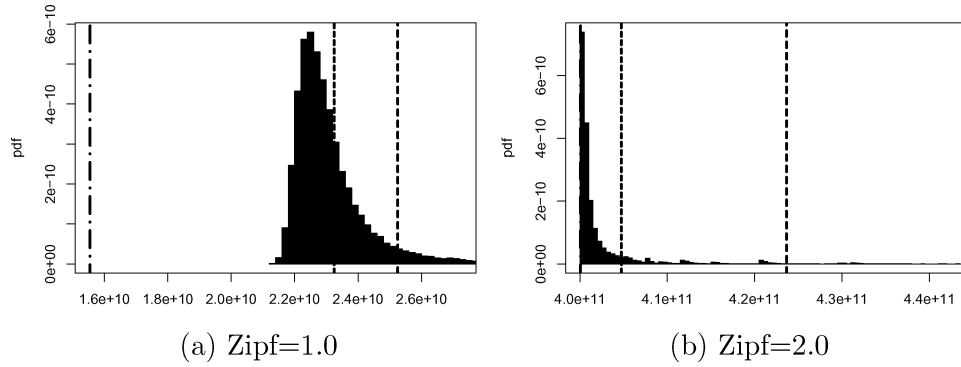


Fig. 6. The distribution of CM sketches for self-join size for Zipf = 1.0 and Zipf = 2.0, respectively. The dotted left line corresponds to the true result. The dashed central line corresponds to the expected value of the distribution, which is an over-estimate of the true result. The dashed right line is positioned one standard deviation away from the expected value.

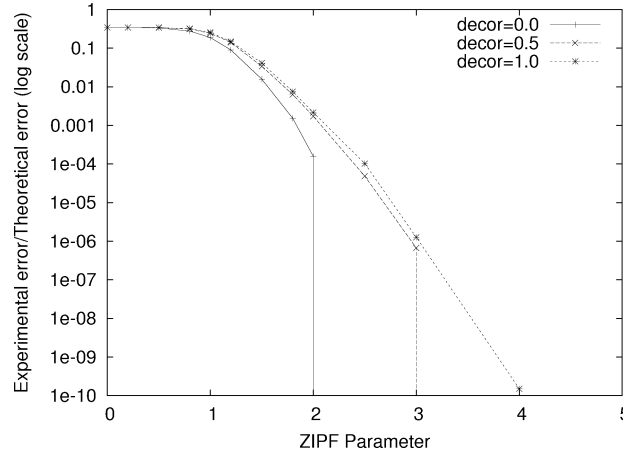


Fig. 7. Confidence bounds for CM sketches as a function of the skewness of the data. The ratio between the experimental bounds and the distribution-independent bounds is plotted for the size of join of two streams with various degrees of correlation (decor = 0 corresponds to streams fully correlated or self-join size, while decor = 1 corresponds to streams completely independent).

the peak is situated at the point corresponding to the high frequencies being hashed into different buckets. Since high frequencies dominate the result, the estimate is in this case closer to the true value. Figure 6 depicts the distribution of X_{CM} for Zipf coefficients equal to 1.0 and 2.0, respectively.

The distribution-independent confidence bounds for CM sketches in Cormode and Muthukrishnan [2005a] are derived from the Markov inequality. Essentially, the error bounds are expressed in terms of the expected value of the over-estimated quantity $\frac{1}{B}(\sum_{i \in I} f_i \sum_{j \in I} g_j - \sum_{i \in I} f_i g_i)$ in $E[X_{CM}]$. Neither the variance nor the bias are considered in deriving these bounds. To verify the accuracy of the confidence bounds, we plot in Figure 7 the ratio between the experimental error obtained for data sets with different Zipf and correlation coefficients (see Section 5) and the corresponding predicted error. The main

observation from these results is that the ratio between the actual error and the prediction decreases as the Zipf coefficient increases, to the point where the gap is many orders of magnitude. In what follows we provide an intuitive explanation for this behavior.

For low skew data the error is almost entirely due to the bias, correctly estimated by the expected value, thus the perfect correspondence between the actual error and the prediction. This observation is inferred from Figure 6(a), which plots the distribution of the elementary sketch estimator for Zipf coefficient equal with 1.0. In this situation, the standard deviation of the elementary estimator is much smaller than the bias. If multiple instances of the elementary sketch are obtained, they will all be relatively close to the expected value (no more than a number of standard deviations to the left), thus their minimum will be close to the expected value. The fact that the standard deviation is small when compared to the bias for low skew data can be predicted using Proposition 3, based on the fact that L_2 norm is much smaller than L_1 norm for low skew data.

For high skew, the standard deviation becomes significantly larger than the bias as can be seen in Figure 6(b). In this situation, even though the bias is still significant, with high probability some of the samples of the elementary sketch will be close to the true value, thus the minimum of multiple elementary sketches will have significantly smaller error. Notice how the shape of the distribution changes when the Zipf coefficient increases: it is normal-like for low skew, but it has no left tail for high skew. The distribution is forced to take this shape when the standard deviation is larger than the bias since CM sketch estimators cannot take values smaller than the true value. Referring back to the moments of the CM elementary estimator in Proposition 3, for large skew the standard deviation is comparable to the expected value, but the bias is much smaller since most of the result is given by the large frequencies whose contribution is accurately captured by the estimator.

While the above discussion gives a good intuition why the theory gives reasonable error predictions for low skew data and makes large errors for high skew data, unfortunately it does not lead to better bounds for skewed data. In order to provide tight confidence bounds, the distribution of the minimum of multiple elementary sketch estimators has to be characterized. While CLT-based results exist for the minimum estimator (see Section 2.7), they provide means to characterize the variance, but not the bias of the minimum estimator. Determining the bias of the minimum is crucial for correct predictions of the error for large skew, but it seems a difficult task since it depends on the precise distribution of the data not only some characteristics like the first few moments. It is worth mentioning that tighter bounds for CM sketches can be obtained if the Zipf coefficient of the data is determined by other means [Cor-mode and Muthukrishnan 2005b]. Notice the particular problems of deriving confidence bounds for CM sketches: high errors are correctly predicted while small errors are incorrectly over-estimated. Consequently, Count-Min sketches are difficult to use in practice because their behavior cannot be predicted accurately.

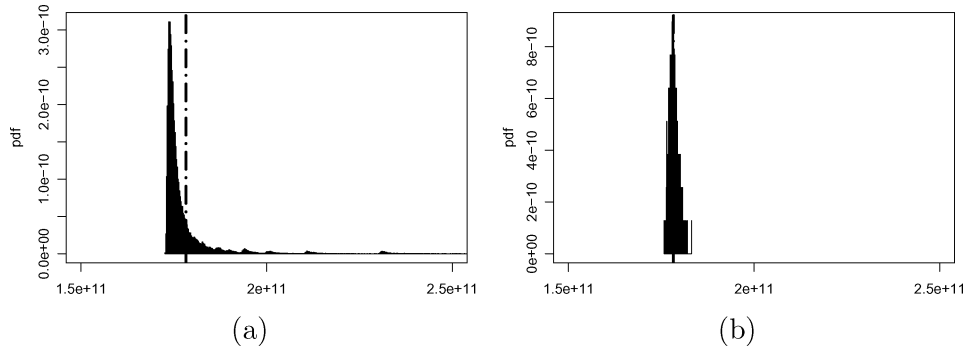


Fig. 8. The distribution of FC sketches for self-join size. (a) depicts the distribution of the basic FC sketch estimator. In (b) the distribution of the same data is plotted after grouping the basic estimators and taking the average. The dotted line corresponds to the true result or expected value.

4.4 Fast-Count Sketches

Fast-Count (FC) elementary estimator is essentially the bias-corrected version of the Count-Min elementary estimator. The bias correction is a translation by bias and a scaling by the factor $\frac{B}{B-1}$. This can be observed in Figure 8 which depicts the distribution of Fast-Count sketches. Everything stated for CM sketch distribution still holds for the distribution of FC sketches, with the major difference that Fast-Count sketches are unbiased, while Count-Min sketches are biased. Given the unbiased estimator and the asymmetric shape of the distribution, mean is the only viable estimator for the expected value, which is also the true value in this case.

The distribution-independent confidence bounds for FC sketches, derived in a similar manner using Chebyshev and Chernoff bounds, are identical to those for AGMS and Fast-AGMS sketches because the first two moments of the distributions are equal. Tighter distribution-dependent confidence bounds are derived using Mean CLT for AGMS and Median CLT for Fast-AGMS sketches, respectively. Although the mean estimator is also used for FC sketches, the asymptotic regime of Mean CLT does not apply in this case because the number of samples averaged is only in the order of tens. The alternative is to use the Student t-distribution for modeling the behavior of the mean (see Section 2.3), but the improvement over the distribution-independent bounds is not so remarkable. In conclusion, both distribution-independent and distribution-dependent bounds can be used for FC sketches without a significant advantage for any of them.

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 data sets 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 then 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 [Alon et al. 1996] 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 that ranges between 50 – 400 ns, depending on the size of the sketch.

5.1 Testbed and Methodology

Sketch Implementation. We implemented a generic framework that incorporates the sketching techniques mentioned throughout the article. Algorithms for generating random variables with limited degree of independence [MassDAL 2006; Rusu and Dobra 2007] are at the core of the framework. Since the sketching techniques have a similar structure, they are designed as a hierarchy parametrized 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 extensive range of possible inputs, thus allowing us to infer general results on the behavior of the compared sketching techniques.

Census Data Set [CPS 2006]. 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

Synthetic Data Generator [Estan and Naughton 2006]. 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 experimental evaluation on frequency vectors over a $2^{14} = 16,384$ size domain that contain 1 million tuples and has 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.

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 9 both on a normal (a) as well as logarithmic (b) scale. As expected, the errors of AGMS and FC sketches are similar (the difference for (close to) uniform distributions is due to the EH3 [Rusu and Dobra 2007] random number generator). While F-AGMS has almost the same behavior as FC (AGMS) for small Zipf coefficients, the F-AGMS error 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 that are (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 nonzero

⁴After eliminating the records with missing values.

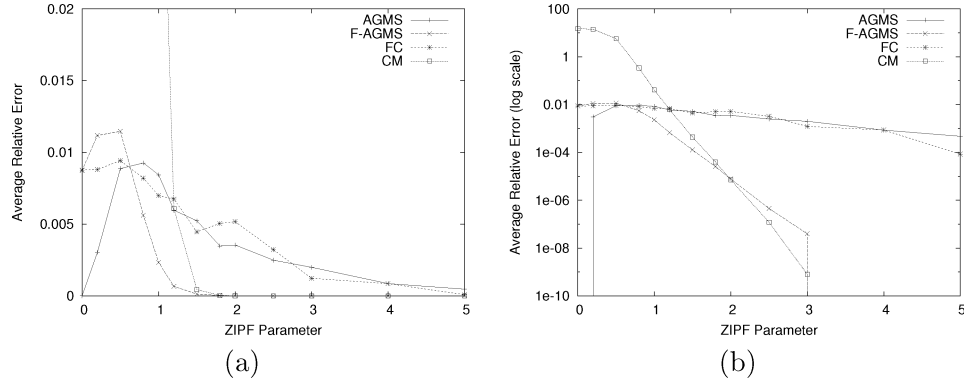


Fig. 9. Accuracy as a function of the Zipf coefficient for self-join size estimation. Both (a) and (b) are plotted from the same experimental data, (a) on a normal scale, while (b) on a logarithmic scale.

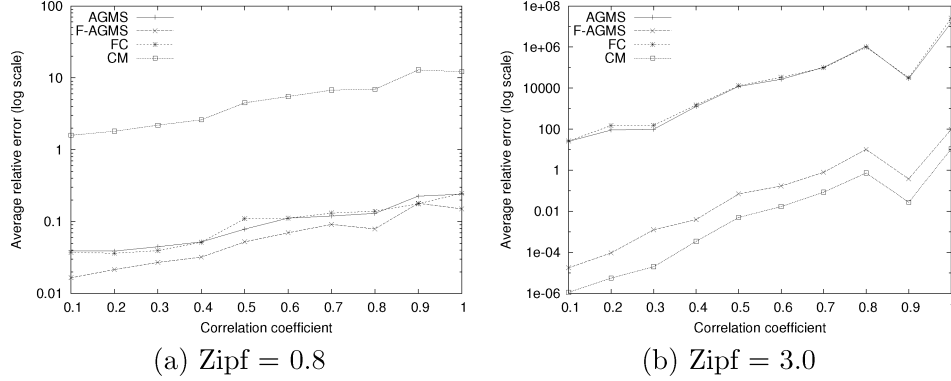


Fig. 10. Accuracy as a function of the correlation coefficient for size of join estimation. The data streams in (a) have Zipf coefficient equal to 0.8, while (b) is for streams with a Zipf coefficient equal to 3.0.

frequencies 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 nonzero 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) 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 10 as a function of the correlation. It can be clearly seen how the relation between the sketch estimators changes as a function of the skew (behavior identical to the self-join size). Moreover, it seems that the degree of correlation

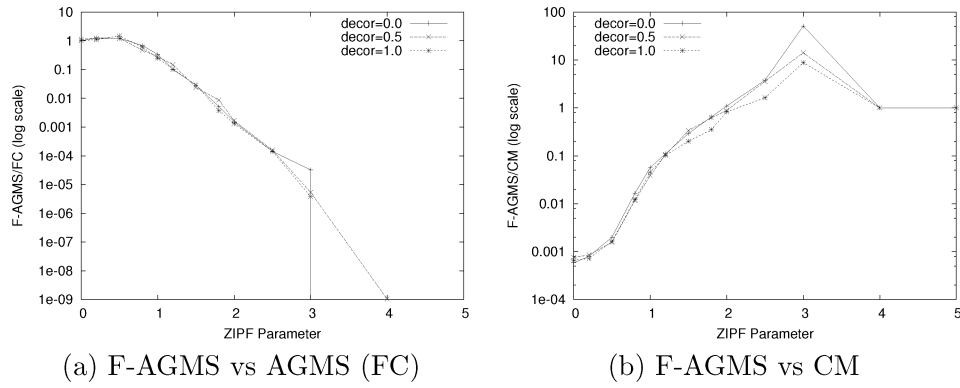


Fig. 11. Relative performance for size of join estimation between pairs of sketch estimators. The ratio of the average relative error as a function of the skewness of the data between F-AGMS and AGMS (a), and F-AGMS and CM (b), respectively, is depicted for different degrees of correlation.

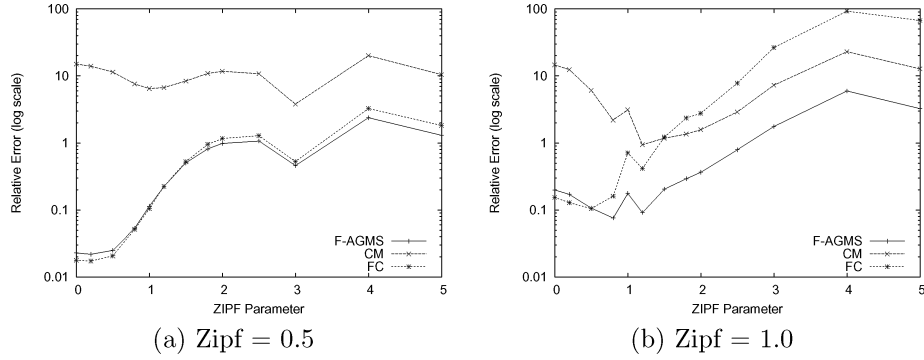


Fig. 12. Accuracy as a function of the skewness of the data for size of join estimation of two data streams having different Zipf coefficients. One of the streams has a constant skew coefficient, 0.5 for (a) and 1.0 for (b), while the skew of the other stream ranges from 0.0 (uniform) to 5.0.

affects all the estimators similarly (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 11 which depicts the relative performance, that is, the ratio of the average relative errors, between pairs of estimators for computing the size of join. Figure 12 plots the accuracy for estimating the size of join of two streams with different skew coefficients. While the error of F-AGMS and FC increases with the skewness of the free stream, the error of CM stays almost constant, having a minimum where the two streams have equal Zipf coefficients. At the same time, it seems that the value of the error is determined by the smallest skew parameter. 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, but its update time is much larger) as a

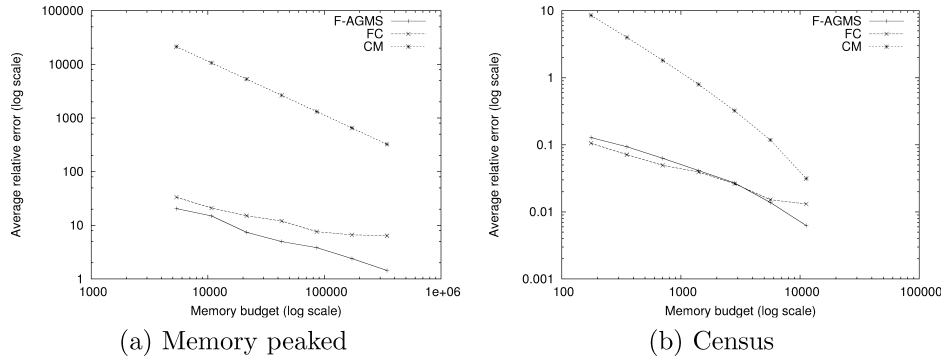


Fig. 13. Accuracy as a function of the available space budget for a synthetic (a) and a real (b) data set. The memory budget is given as the number of counters used by all sketch structures.

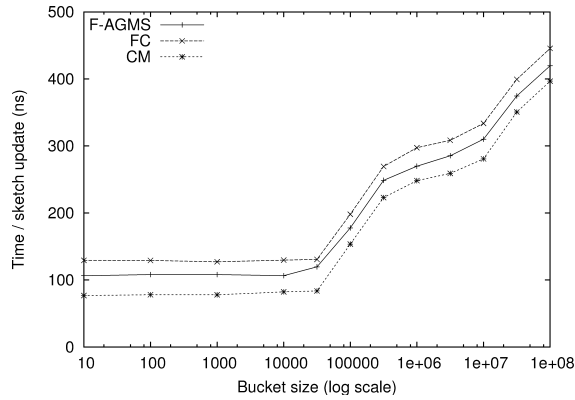


Fig. 14. Update time as a function of the number of counters in a sketch that has only one row.

function of the space available is represented in Figure 13 for one of Estan's synthetic data sets (a) and for the census real-life data set (b). 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 are different. As shown in Figure 14, 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

Table IV. Expected Statistical/Empirical Performance
(same scale as in Table III).

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

per second on the test machine⁵), making hash sketches a viable solution for high-speed data stream processing.

5.3 Discussion

As we have seen, the statistical and empirical study in this article paints a different picture than suggested by the theory (see Table III). Table IV 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 shows 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 for 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 in Figure 11. 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 11 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. The better accuracy of F-AGMS when compared to AGMS is somehow counter-intuitive, since only one counter is updated instead of all the counters. The main reason, is that F-AGMS sketches have a good chance

⁵The results in Figure 14 are for a Xeon 2.8 GHz processor with 512 KB of cache. The main memory is 4 GB.

to separate the most frequent items from each other—if a bucket contains a single frequent item and multiple infrequent items, the error is small since the frequent item dominates. When frequent items are not separated (they collide in the same hash bucket), the error could be extremely large (this is why on average the error is the same). Luckily, the median removes these outlier situations, and thus the error is smaller overall. 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.

6. SKETCHES FOR INTERVAL DATA

The problem we treat in this section is to estimate the size of join between a data stream given as points and a data stream given as intervals using sketches. Notice that this remains a size of join problem defined over the frequencies of individual points, but since one of the streams is specified by intervals rather than individual points, if basic sketches are used the update time is proportional to the size of the interval, which is undesirable. There exist two solutions, DMAP and fast range-summation, that have update time sublinear in the interval size for this problem. DMAP [Das et al. 2004] consists in mapping both the intervals and the points into the space of dyadic intervals in order to reduce the size of the interval representation. Since both intervals and points map to a logarithmic number of dyadic intervals in the dyadic space, the update time becomes poly-logarithmic with respect to the input. Fast range-summation [Rusu and Dobra 2007] uses properties of the pseudo-random variables in order to sketch intervals in sublinear time, while points are sketched as before. While the update time of these methods is poly-logarithmic with respect to the size of the interval, since they are based on AGMS sketches, the update time is also proportional with the size of the sketch. In the light of the statistical and empirical evaluation of hash-based sketches, the update time due to sketching could be significantly improved without loss in accuracy. The question we ask and thoroughly explore in this section is whether hash-based sketches can be combined successfully with the two methods to sketch interval data. The insights gained from the statistical analysis and the empirical evaluation of the hash-based sketching techniques are applied in this section to provide variants of the two methods for sketching interval data that have significantly smaller update time and comparable accuracy.

As mentioned in Section 1, sketching interval data is a fundamental problem that is used as a building block in more complex problems such as estimating the size of spatial joins and building dynamic histograms. For example, for the size of spatial joins problem in which two data streams of intervals are given, two sketches are built for each stream, one for the entire interval and one for the end-points. The size of the spatial join between the two interval data streams is subsequently estimated as the average of the product of the interval sketch from one stream and the sketch for the end-points from the other stream (see Das et al. [2004] for complete details).

6.1 Problem Formulation

The problem considered in this section is a derivation of the size of join problem defined in Section 1.1, in which one of the two data streams is given by interval, frequency pairs rather than key, frequency pairs. The frequency is attached to each element in the interval, not only to a single key. Formally, let $S = (e_1, w_1), \dots, (e_s, w_s)$ and $T = ([l_1, r_1], v_1), \dots, ([l_t, r_t], v_t)$ be two data streams, where the keys e_i and the intervals $[l_i, r_i]$, with $l_i \leq r_i$, are members of the set $I = \{0, 1, \dots, N - 1\}$, and w_i and v_i , respectively, represent frequencies. The computation using sketches of the *size of join* of the two data streams defined as the *inner-product* of their frequency vectors remains our focus. Notice that it is straightforward to reduce this problem to the basic size of join problem by observing that a pair $([l_i, r_i], v_i)$ in T can be represented as an equivalent set of pairs (e_j, v_i) in S , with e_j taking all the values between l_i and r_i , $l_i \leq e_j \leq r_i$. The drawback of this solution is the time to process an interval which is linear in the size of the interval.

6.2 Dyadic Mapping (DMAP)

The DMAP method uses dyadic intervals in order to reduce the representation of an interval, thus making possible the efficient sketching of intervals (see [Gilbert et al. 2005; Das et al. 2004; Rusu and Dobra 2007] for details). DMAP is based on a set of three transformations to which the size of join operation is invariant. The original domain is mapped into the domain of all possible dyadic intervals that can be defined over it. An interval in the original domain is mapped into its minimal dyadic cover in the new domain. By doing this, the representation of the interval reduces to at most a logarithmic number of points in the new domain, that is, the number of sketch updates reduces from linear to logarithmic in the size of the interval. At the same time, a point in the original domain maps to the set of all dyadic intervals that contain the point in the new domain, thus increasing the number of sketch updates from one to logarithmic in the size of the original domain. DMAP allows the correct approximation of the size of join in the mapped domain with the added benefit that the sketch of each relation can be computed efficiently, since both for an interval, as well as a point, at most $\log |I| = n$ dyadic intervals have to be sketched.

The application of any of the sketching methods in the dyadic domain is straightforward. For a point, the sketch data structure is updated with all the dyadic intervals that contain the point (exactly $\log |I| = n$). For an interval, the sketch is updated with the dyadic intervals contained in the minimal dyadic cover (at most $2n - 2$, but still logarithmic in the size of the interval). Specifically, in the case of AGMS sketches, all the counters are updated for each dyadic interval, while in the case of hash-based sketches (Fast-AGMS, Fast-Count, Count-Min), only one counter in each row is updated for each dyadic interval. Notice that the update procedure is identical to the procedure for point data streams since a dyadic interval is represented as a point in the dyadic domain. Once the sketches for the two data streams are updated, the estimation procedure corresponding to each type of sketch described in Section 3 is immediately applicable.

The experimental results in Rusu and Dobra [2007] show that DMAP has significantly worse accuracy, as much as a factor of 8, than fast range-summable methods for AGMS sketches. We provide an explanation for this behavior based on the statistical analysis in Section 4 and the empirical results in Section 5. At the same time, we provide evidence that the performance of DMAP for hash-based sketches (Fast-AGMS in particular) cannot be significantly better. To characterize statistically the performance of DMAP, we first look at the distribution of the two data streams in the dyadic domain. The distribution of the point data stream has a peak corresponding to the domain (the largest dyadic interval) due to the fact that this dyadic interval contains all the points, so its associated counters get updated for each streaming point. The dyadic intervals at the second level, of size half of the domain size, have high frequencies for the same reason. As the size of dyadic intervals decreases, their frequency decreases too, to the point it is exactly the true frequency for point dyadic intervals. Unfortunately, the high frequencies in the dyadic domain are outliers because their impact on the size of join result is minimal (for example, the domain dyadic interval appears in the size of join only if the interval data stream contains the entire domain as an interval). Practically, DMAP transforms the distribution of the point data stream into a skewed distribution dominated by outliers corresponding to large dyadic intervals. The effect of DMAP over the distribution of the interval data stream is far less dramatic, but more difficult to quantify. This is due to the fact that both the size of the interval and the position are important parameters. For example, two intervals of the same size, one which happens to be dyadic and one translated by only a position, can generate extremely different minimal dyadic covers and, thus, distributions in the dyadic domain. Even without any further assumptions on the distribution of the interval data stream, we expect the skewed distribution of the point stream to affect negatively the estimate, due to the outliers corresponding to large dyadic intervals. At the same time, we would expect not to have a significant difference between AGMS (Fast-Count) and Fast-AGMS sketches unless the distribution of the interval stream is also skewed towards large dyadic intervals. The reason for this lies in the fact that since the point stream over the dyadic domain is skewed and the behavior of the sketch estimators for the size of join of two streams with different Zipf coefficients is governed by the smallest skew factor, the overall behavior is determined by the skew of the interval stream. Figure 12 shows a significant difference between FC and F-AGMS only when both streams are skewed. The experimental results in Section 6.4 verify these hypotheses.

An evident drawback of DMAP is that it cannot be extended easily to the case when both input data streams are given as intervals. If the sketches are simply updated with the dyadic intervals in the minimal dyadic cover, the size of join of the points in the dyadic domain is computed which is different from the size of join in the original domain because a point in the dyadic domain corresponds to a range of points in the original domain. Updating one of the sketches with the product of the size of the dyadic interval and the frequency instead of only the frequency seems to be an easy fix that would compensate for the reduction in the representation. This is not the case because a point can be

part of different dyadic intervals with different sizes, a situation that cannot be eliminated by moving in the dyadic domain.

6.2.1 DMAP Counts. A possible improvement to the basic DMAP method is to keep exact counts for large dyadic intervals in both streams and to compute sketches only for the rest of the data. The idea of keeping exact counts for the first few levels of the hierarchy was proposed for Count-Min sketches in Cormode and Muthukrishnan [2005a]. By doing this, the distribution of the point stream in the dyadic domain becomes closer to the original distribution since the effect of the outliers is neutralized. The contribution of the large dyadic intervals to the size of join is computed exactly through the counts, while the contribution of the rest of dyadic intervals is better approximated through the sketches. Although the evident resemblance between this technique and other types of complex sketches (e.g., count sketches [Charikar et al. 2002], skimmed sketches [Ganguly et al. 2004], and red sketches [Ganguly et al. 2005]), there is a subtle difference. While for all the other techniques the high frequencies have to be determined and represent an important fraction of the result, in this case they are known before and represent an outlier whose effect has to be minimized. In order to quantify the error of this method and to determine the optimal number of exact counts, similar solutions to Charikar et al. [2002] and Ganguly et al. [2005] can be applied with the added complexity of dealing with interval distributions over a dyadic domain. The deeper insights such an analysis could reveal are hard to determine, since even the exact behavior of DMAP is only loosely quantified in Das et al. [2004] and Rusu and Dobra [2007]. The empirical results we provide in Section 6.4 show that the improvement is effective.

6.3 Fast Range-Summation

While DMAP uses mappings in the dyadic domain in order to sketch intervals efficiently, fast range-summation methods are based on properties of the random variables that allow the sketching of an interval in a number of steps sublinear in the size of the interval. Specifically, the sum of random variables over dyadic intervals is computed in a constant number of steps, and, since there exists a logarithmic number of dyadic intervals in the minimal dyadic cover of any interval, the number of steps to sketch the entire interval is logarithmic in its size. Rusu and Dobra [2007] show that fast range-summation is a property of the generation scheme of the random variables and that there exist only two practical schemes applicable to AGMS sketches, EH3 and BCH3, respectively. Moreover, the performance of BCH3 is highly sensitive to the input data, so we consider only EH3 in this article. Rusu and Dobra [2007] use fast range-summation only in the context of AGMS sketches where the update of each key (interval) affects each counter in the sketch structure. More exactly, for all the elements in an interval, the same counter has to be updated (and all the counters overall). This is not the case anymore for hash-based sketches where different counters are updated for different keys (unless they are hashed into the same bucket). In this section we show that fast range-summation and random hashing are conflicting operations and, consequently, that fast

range-summation is not applicable to hash-based sketches (Fast-AGMS, Fast-Count, Count-Min). Fortunately, we show that fast range-summation for AGMS sketches can be applied in conjunction with deterministic partitions of the domain without loss in error, but with a significant improvement in the update time.

As mentioned in Section 3, the main drawback of AGMS sketches is the update time. For each stream element, each counter in the sketch structure has to be updated. Essentially, each counter is a randomized synopsis of the entire data. Fast range-summation exploits exactly this additive property to sketch intervals efficiently since the update corresponding to each element in the interval has to be added to the same counter. Hash-based sketches randomly partition the domain I of the key attribute and associate a single counter in the sketch structure with each of these partitions. For each stream element, only the counter corresponding to its random partition is updated, thus the considerable gain in update time. In order to determine if fast range-summation can be extended to hash-based sketches, we focus on the interaction between random hashing, whose goal is to evenly partition the keys into buckets, and the efficient sketching of continuous intervals for which the maximum benefit is obtained when all the elements in the interval are placed into the same bucket. Proposition 4 relates the number of counters that have to be updated in a hash-based sketch to the size of the input interval:

PROPOSITION 4. *Given a hash function $h : I \rightarrow B$ and an interval $[\alpha, \beta]$ of size l , the number of buckets touched by the function h when applied to the elements in $[\alpha, \beta]$ is on expectation $B[1 - (1 - \frac{1}{B})^l]$.*

PROOF. Let X_i be a 0/1 random variable corresponding to each of the B buckets of the hash function h , $0 \leq i < B$. X_i takes the value 1 when at least one element in the range $[\alpha, \beta]$ is hashed into the bucket i and the value 0 otherwise:

$$X_i = \begin{cases} 1, & \text{if } \exists j \in [\alpha, \beta] \text{ with } h(j) = i \\ 0, & \text{otherwise.} \end{cases}$$

The expected value $E[X_i]$ can be computed as

$$E[X_i] = P[X_i = 1] = 1 - P[X_i = 0] = 1 - \left(1 - \frac{1}{B}\right)^l$$

since the probability of an element to be hashed in the i th bucket is $\frac{1}{B}$. The expected value of the number of buckets touched by h over $[\alpha, \beta]$ is then

$$E[X] = E\left[\sum_{i=0}^{B-1} X_i\right] = \sum_{i=0}^{B-1} E[X_i] = B \left[1 - \left(1 - \frac{1}{B}\right)^l\right] \quad (4)$$

where X is defined as $X = \sum_{i=0}^{B-1} X_i$. \square

In order to give some practical interpretation to the above proposition, we consider the size l to be proportional with the number of buckets B , i.e., $l = kB$,

for $k > 0$. This allows us to rewrite Eq. (4) as

$$B \left[1 - \left(1 - \frac{1}{B} \right)^l \right] = B \left[1 - \left(1 - \frac{1}{B} \right)^{kB} \right] \approx B \left(1 - \frac{1}{e^k} \right)$$

where we used the approximation $\frac{1}{e} = \lim_{B \rightarrow \infty} (1 - \frac{1}{B})^B$.

Example 6. For a hash function with B buckets, 63.21% of the buckets are touched on expectation when h is applied to an interval of size B . The number of buckets increases to 86.46% when the size of the interval is twice the number of buckets B , and to 98.16% for $k = 4$.

The above corollary states that for intervals of size at least four times the size of the hash, almost all the buckets are touched on expectation. This completely eliminates the effect of hashing for sketching intervals since AGMS sketches already require the update of each counter in the sketch. The difference is that, for AGMS sketches, each counter is updated with the entire interval, while for hash-based sketches a counter is updated only with the elements in the interval assigned to the random partition corresponding to that counter. Although the number of updates per counter is smaller for hash-based sketches, determining how many (and which) elements in the given interval update the same counter without looking at the entire interval is a difficult problem. The only solution we are aware of is for 2-universal hash functions, so applicable only to Fast-AGMS and Count-Min sketches. It consists in applying the sublinear algorithm proposed in Aduri and Tirthapura [2005] for counting how many elements in the interval are hashed into a range of buckets either for each bucket or for ranges of increasing size. Notice that actually this is not even enough for Fast-AGMS sketches for which the interaction between hashing and EH3 (or BCH3) [Rusu and Dobra 2007] has to be quantified. While fast range-summation takes advantage of properties of the generating scheme for the particular form of dyadic intervals, determining the contribution of the elements in the same random partition without considering each element separately has to be more difficult due to the lack of structure. Consequently, fast range-summation is directly applicable only to Count-Min sketches throughout the hash-based sketching techniques, with the requirement that each counter is updated when sketching an interval.

6.3.1 Fast Range-Summation with Domain Partitioning. The intermediate solution between AGMS sketches, which update all the counters, and hash-based sketches, which update only one counter for a given key, is sketch partitioning [Dobra et al. 2002]. The domain I is partitioned in continuous blocks rather than random blocks. A number of counters from the sketch structure proportional to the size of the block is assigned to each block. When the update of a key has to be processed, only the counters in the block corresponding to that key are updated. This method can be easily extended to fast range-summation intervals without the need to update all the counters unless the size of the interval is close to the size of the domain. The intersection between the given interval and each partition is first determined and, for each nonempty intersection, the fast

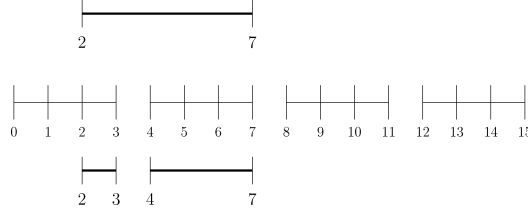


Fig. 15. Fast range-summation with domain partitioning. The domain $I = \{0, \dots, 15\}$ is split into four partitions of equal size. The intersection between the input interval $[2, 7]$ and the partitions is computed. For each nonempty intersection, fast range-summation is applied.

range-summation algorithm is applied only to the set of associated counters. Thus, a number of counters proportional with the number of nonempty intersections (and indirectly proportional to the size of the interval) has to be updated. In what follows we provide an example to illustrate how fast range-summation with sketch partitioning works.

Example 7. Consider the domain $I = \{0, \dots, 15\}$ to be split into 4 equi-width partitions as depicted in Figure 15. For simplicity, assume that the available AGMS sketch consists of 8 counters which are evenly distributed between the domain partitions, 2 for each partition. Instead of having a single estimator for the entire domain, a sketch estimator combining the counters in the partition is built for each partition. The final estimator is the sum of these individual estimators corresponding to each partition.

Figure 15 depicts the update procedure for the interval $[2, 7]$. The non-empty intersections $[2, 3]$ and $[4, 7]$ correspond to partition 0 and 1, respectively. The fast range-summation algorithm is applied to each of these intervals only for the counters associated with the corresponding domain partition, not for all the counters in the sketch. In our example, fast range-summation is applied to interval $[2, 3]$ and the two counters associated to partition 0, and to interval $[4, 7]$ and the two counters associated to partition 1, respectively. Overall, only four counters are updated, instead of eight, for sketching the interval $[2, 7]$.

The advantage of domain partitioning is the fact that the update time is smaller when compared to the basic fast range-summation method. This is the case because only a part of the sketch has to be updated if the interval is not too large with respect to the size of a partition. In particular, only the sketches corresponding to the partitions that intersect the interval need to be updated, which means that the speedup is proportional to the ratio between the number of partitions and the average number of partitions an interval intersects. When points are sketched, only the counters corresponding to the partition the point belongs to need to be updated instead of all the counters in the sketch. In the above example only two counters have to be updated for each point, instead of eight.

Notice that, as shown in Dobra et al. [2002], any partitioning of the domain can be used and the number of counters associated to each partition can also be different from partition to partition. More precisely, any partitioning and

any allocation scheme for the counters results in an unbiased estimator for the size of join. An important question though is what is the variance of the estimator, which is an indicator for accuracy. In Dobra et al. [2002] a sophisticated method to partition and allocate the counters per partition was proposed in order to minimize the variance of the estimator. For gains to be obtained, regions of the domain where high frequencies in one stream match small frequencies in the other have to be identified. Since in this particular situation we do not expect large frequencies for the interval stream, as explained in Section 6.2, we do not expect the sketch partitioning technique in Dobra et al. [2002] to be able to reduce the variance significantly. Moreover, using the fact that the variance of the estimator remains the same if the partitioning is random (see Proposition 4.2), as long as there does not exist significant correlation between the partitioning scheme and the input frequencies, we expect the variance of the estimator to remain the same. The expected distribution of the interval frequencies also suggests that a simple equi-width partitioning should behave reasonably well. Indeed, the experimental results in Section 6.4 show that this partitioning is effective in reducing the update time, while the error of the estimate remains roughly the same. Notice that in the worst case when the frequency distribution is focused on a small range of the domain, equi-width partitioning has a negative effect on the accuracy because the value of the variance increases. If multiple counters are assigned to the problematic partitions, for example using the allocation scheme in Dobra et al. [2002], the update time increases since more counters need to be updated and the advantage of domain partitioning on fast range-summing is lost.

6.4 Experimental Results

In this section we present the results of the empirical study designed to evaluate the performance of five of the algorithms for efficiently sketching intervals introduced previously. The five methods tested are: AGMS DMAP, F-AGMS DMAP (F-AGMS), F-AGMS DMAP with exact counts (F-AGMS COUNTS), fast range-summation AGMS (AGMS), and fast range-summation AGMS with sketch partitioning (AGMS P). Methods based on Count-Min sketches are excluded due to their high sensitivity to the input data, while for Fast-Count sketches the same behavior as for AGMS is expected (see Section 5), where applicable. The accuracy and the update time per interval are the two quantities measured in our study for the size of a spatial join problem involving intervals (see Das et al. [2004]). Following the experimental setup in Das et al. [2004] and Rusu and Dobra [2007], three real data sets are used in our experiments: LANDO, describing land cover ownership for the state of Wyoming and containing 33,860 objects; LANDC, describing land cover information such as vegetation types for the state of Wyoming and containing 14,731 objects; and SOIL, representing the Wyoming state soils at a $1 : 10^5$ scale and containing 29,662 objects. The use of synthetic generators for interval data is questionable because it is not clear what are acceptable distributions for the size of the intervals, as well as the position of the interval end-points. In a similar manner to Section 5, each experiment is performed 100 times, and either the average relative error, that is,

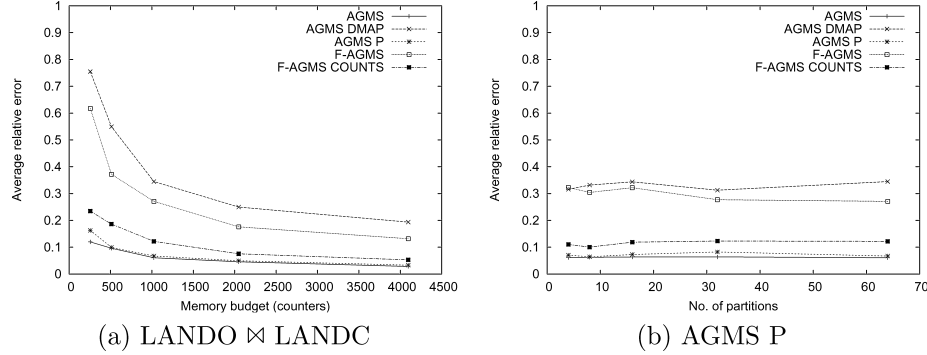


Fig. 16. Accuracy. In (a) the average relative error is depicted as a function of the memory budget for the size of spatial join between the data sets LANDO and LANDC. The sketch contains 4 rows and AGMS P uses 64 partitions. The dependence of the accuracy on the number of partitions is plotted in (b) for a sketch of 4 rows with 256 counters each. The same data sets, LANDO and LANDC, are joined. Seven exact counts at the first 3 dyadic levels are used in the implementation of F-AGMS COUNTS.

$\frac{|\text{actual} - \text{estimate}|}{\text{actual}}$, or the median update time over the number of experiments is reported.

Accuracy. We pursue two goals in our accuracy experiments. First, we determine the dependence of the average relative error on the memory budget, that is, the number of counters in the sketch structure. For this, we run experiments with different sketch configurations having either 4 or 8 rows in the structure and varying the number of counters in a row between 64 and 1024. Second, we want to establish the relation between the accuracy and the number of partitions for AGMS P. For this, we distribute the counters in the sketch into 4 to 64 groups corresponding to an equal number of partitions of the domain. Given the previous results in Rusu and Dobra [2007] for AGMS and AGMS DMAP, we expect the results for F-AGMS to be close to AGMS DMAP, with some improvement for F-AGMS COUNTS, which eliminates the effect of outliers to some extent. At the same time, we expect that partitioning does not significantly deteriorate the performance of AGMS P unless it is applied to the extreme where only one counter corresponds to each partition.

Figure 16 depicts the accuracy results for a specific parameter configuration. The same trend was observed for the other settings, with the normal behavior for the confined action of each parameter. As expected, the error decreases as the memory budget increases for all the methods (left plot). The behavior of DMAP methods is more sensitive to the available memory, without a significant difference between AGMS and F-AGMS sketches, but still slightly favorable to F-AGMS. What is significant is the effect of maintaining exact counts for F-AGMS DMAP. The error reduces drastically, to the point it is almost identical with the error of fast range-summation AGMS, the most accurate of the studied methods. This is due to limiting the effect of outliers that would otherwise significantly deteriorate the accuracy of the sketch. Notice the reduced levels of the error for fast range-summation methods, even when only low memory

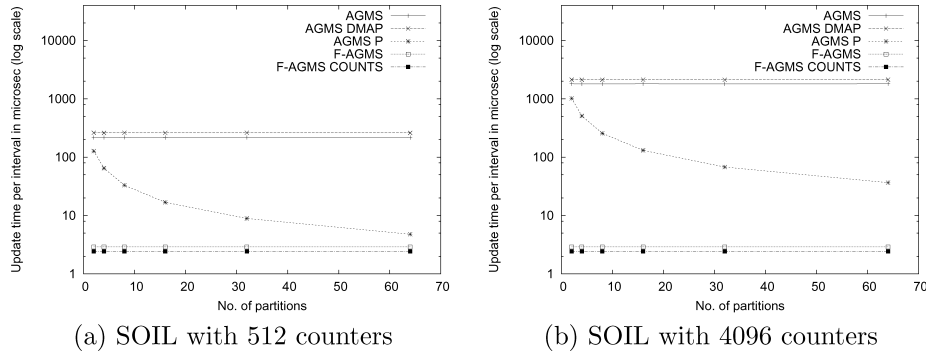


Fig. 17. Update time per interval as a function of the number of partitions for the SOIL data set. The sketch structure consists of a single row with either 512 counters (a) or 4096 counters (b).

is available. Our second goal was to detect the effect partitioning has on the accuracy of fast range-summation AGMS. From the right plot in Figure 16, we observe that partitioning has almost no influence on the accuracy of AGMS, the errors of the two methods being almost identical even when a significant number of partitions is used. Clearly, we expect the accuracy to drop after a certain level of partitioning, when the number of counters corresponding to a partition is small. The error of the other methods is plotted in (b) only for completeness. The fluctuations are only due to the randomness present in the methods, since the experiments were repeated with the same parameters for each different configuration of AGMS P.

Update Time. Our objective is to measure the time to update the sketch structure for the presented sketching methods. For a sketch consisting of only one row of counters, we know that the time is linear for AGMS sketches since all the counters have to be updated. This is reflected in Figure 17 that depicts the update time per interval for two sketch structures, one with 512 counters (left), and one with 4096 counters (right).⁶ Notice that Figure 17 actually plots the update time per interval as a function of the number of partitions, thus the constant curves for all the methods except AGMS P. As expected, the update time for AGMS P decreases as the number of partitions increases, since the number of counters in a partition decreases. The decrease is substantial for a sketch with 512 counters, to the point where the update time is almost identical with the time for DMAP over F-AGMS sketches, the fastest method. Notice the significant gap of 2 to 4 orders of magnitude between the methods based on AGMS and those based on F-AGMS sketches, with the update time for F-AGMS being in the order of a few micro-seconds, while the update time for AGMS is in the order of milli-seconds.

6.5 Discussion

As it was already known [Rusu and Dobra 2007], DMAP is inferior both in accuracy and update time to fast range-summation for AGMS sketches, facts

⁶We used the same machine as in Section 5.

reconfirmed by our experimental results. While DMAP can be used in conjunction with any type of sketching technique, fast range-summation is immediately applicable only to AGMS sketches. In order to improve the update time of AGMS, we propose AGMS P, a method that reduces the number of counters that need to be updated by partitioning the domain of the key and distributing the counters over the partitions. Even with a simple partitioning that splits the domain into buckets of the same size, as is done for equi-width histograms, the improvement we obtain is remarkable; the update time becoming comparable with that for F-AGMS sketches, while the error remains as good as the error of fast range-summation AGMS. The only improvement gained by using DMAP over F-AGMS sketches is in update time. With a simple implementation modification that keeps exact counts for large dyadic intervals (F-AGMS COUNTS), the error drops significantly and becomes comparable with the error of fast range-summation AGMS. The roots of this modified method lie in the statistical analysis presented in Section 4.

Overall, to obtain methods for sketching intervals that have both small error and efficient update time, the basic techniques (DMAP and fast range-summation) have to be modified. F-AGMS COUNTS is a modification of DMAP over F-AGMS sketches with extremely efficient update time and with error approaching the standard given by AGMS for large-enough memory. AGMS P is a modification of fast range-summation AGMS that has excellent error and with update time close to that of F-AGMS sketches when the number of partitions is large enough. In conclusion, we recommend the use of F-AGMS COUNTS when the update time is the bottleneck and AGMS P when the available space is a problem.

7. CONCLUSIONS

In this article 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 that are known about these sketches. The analysis reveals that Fast-AGMS and Count-Min sketches have much better performance than the theoretical prediction for skewed data by a factor as much as 10^6 to 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. At the same time, Fast-AGMS sketches seem to represent the preferred choice as a basic block in more complex sketching techniques such as *skimmed sketches* [Ganguly et al. 2004] and *red-sketches* [Ganguly et al. 2005].

Fast range-summation remains the most accurate method to sketch interval data. Unfortunately, it is applicable only to AGMS sketches and, thus, it is

not practical due to the high update time. The solution we propose in this article is based on the partitioning of the domain and of the counters in the sketch structure in order to reduce the number of counters that need to be updated. The improvement in update time is substantial, getting close to DMAP over Fast-AGMS sketches, the fastest method studied. Moreover, by applying a simple modification inspired by statistical analysis and the empirical study of the sketching techniques, the accuracy of DMAP over F-AGMS can be increased significantly, to the point where it is almost equal to the accuracy of fast range-summation over AGMS for large-enough space. Considering the overall results for sketching interval data, we recommend the use of the fast range-summation method with domain partitioning whenever the accuracy is critical and the use of DMAP COUNTS method over F-AGMS sketches in situations where the time to maintain the sketch is critical.

REFERENCES

- ADURI, P. AND TIRTHAPURA, S. 2005. Range efficient computation of F_0 over massive data streams. In *Proceedings of the 21st IEEE ICDE International Conference on Data Engineering*. IEEE Computer Society, 32–43.
- ALON, N., GIBBONS, P. B., MATIAS, Y., AND SZEGEDY, M. 2002. Tracking join and self-join sizes in limited storage. *J. Comput. Syst. Sci.* 64, 3, 719–747.
- ALON, N., MATIAS, Y., AND SZEGEDY, M. 1996. The space complexity of approximating the frequency moments. In *Proceedings of the 28th ACM Symposium on Theory of Computing*. ACM, New York, 20–29.
- AN, N., YANG, Z.-Y., AND SIVASUBRAMANIAM, A. 2001. Selectivity estimation for spatial joins. In *Proceedings of the 17th IEEE ICDE International Conference on Data Engineering*. IEEE Computer Society, 368–375.
- BALANDA, K. P. AND MACGILLIVRAY, H. L. 1988. Kurtosis: A critical review. *J. Amer. Statist.* 42, 2, 111–119.
- CHARIKAR, M., CHEN, K., AND FARACH-COLTON, M. 2002. Finding frequent items in data streams. In *Proceedings of the 29th International Colloquium on Automata, Languages and Programming*. Springer, 693–703.
- COLES, S. 2001. *An Introduction to Statistical Modeling of Extreme Values*. Springer, Berlin, Germany.
- CORMODE, G. AND GAROFALAKIS, M. 2005. Sketching streams through the net: distributed approximate query tracking. In *Proceedings of the 31st International Conference on Very Large Data Bases*. VLDB Endowment, 13–24.
- CORMODE, G. AND MUTHUKRISHNAN, S. 2005a. An improved data stream summary: the count-min sketch and its applications. *J. Algor.* 55, 1, 58–75.
- CORMODE, G. AND MUTHUKRISHNAN, S. 2005b. Summarizing and mining skewed data streams. In *Proceedings of SIAM Conference on Data Mining*. SIAM, Philadelphia, PA.
- CPS. 2006. <http://www.census.gov/cps>.
- DAS, A., GEHRKE, J., AND RIEDEWALD, M. 2004. Approximation techniques for spatial data. In *Proceedings of the 23rd ACM SIGMOD International Conference on Management of Data (SIGMOD'04)*. ACM, New York, 695–706.
- DOBRA, A., GAROFALAKIS, M., GEHRKE, J., AND RASTOGI, R. 2002. Processing complex aggregate queries over data streams. In *Proceedings of the 21st ACM SIGMOD International Conference on Management of Data (SIGMOD'02)*. ACM, New York, 61–72.
- ESTAN, C. AND NAUGHTON, J. F. 2006. End-biased samples for join cardinality estimation. In *Proceedings of the 22nd IEEE International Conference on Data Engineering*. IEEE Computer Society, 20–31.
- GANGULY, S., GAROFALAKIS, M., AND RASTOGI, R. 2004. Processing data-stream join aggregates using skimmed sketches. In *Proceedings of the 9th International Conference on Extending Database Technology*. Springer, 569–586.

- GANGULY, S., KESH, D., AND SAHA, C. 2005. Practical algorithms for tracking database join sizes. In *Proceedings of the 25th Conference on Foundations of Software Technology and Theoretical Computer Science*. Springer, 297–309.
- GILBERT, A. C., KOTIDIS, Y., MUTHUKRISHNAN, S., AND STRAUSS, M. J. 2005. Domain-driven data synopses for dynamic quantiles. *IEEE Transactions on Knowl. Data Engin.* 17, 7, 927–938.
- HAAS, P. J. AND HELLERSTEIN, J. M. 1999. Ripple joins for online aggregation. In *Proceedings of the eighteenth ACM SIGMOD International Conference on Management of Data (SIGMOD'99)*. ACM Press, New York, 287–298.
- KEMPE, D., DOBRA, A., AND GEHRKE, J. 2003. Gossip-based computation of aggregate information. In *Proceedings of the 44th IEEE Symposium on Foundations of Computer Science*. IEEE Computer Society, 482–491.
- MASSDAL. 2006. http://www.cs.rutgers.edu/_muthu/massdal.html.
- MOTWANI, R. AND RAGHAVAN, P. 1995. *Randomized Algorithms*. Cambridge University Press, Cambridge, UK.
- OLIVE, D. J. 2005. A simple confidence interval for the median. Manuscript. www.math.siu.edu/olive/ppmedci.pdf.
- PENNECCHI, F. AND CALLEGARO, L. 2006. Between the mean and the median: The L_p estimator. *Metrologia* 43, 3, 213–219.
- PRICE, R. M. AND BONETT, D. G. 2001. Estimating the variance of the sample median. *J. Statist. Comput. Simul.* 68, 3, 295–305.
- RUSU, F. AND DOBRA, A. 2007. Pseudorandom number generation for sketch-based estimations. *ACM Trans. Datab. Syst.* 32, 2, 11.
- SACHS, L. 1984. *Applied Statistics—A Handbook of Techniques*. Springer, Berlin, Germany.
- SHAO, J. 1999. *Mathematical Statistics*. Springer, Berlin, Germany.
- THORUP, M. AND ZHANG, Y. 2004. Tabulation based 4-universal hashing with applications to second moment estimation. In *Proceedings of the 15th ACM-SIAM Symposium on Discrete Algorithms*. 615–624.

Received July 2007; revised February 2008; accepted April 2008