Near-Optimal Compression of Probabilistic Counting Sketches for Networking Applications

Björn Scheuermann
Computer Networks Research Group
Heinrich Heine University
Düsseldorf, Germany
scheuermann@cs.uni-duesseldorf.de

Martin Mauve
Computer Networks Research Group
Heinrich Heine University
Düsseldorf, Germany
mauve@cs.uni-duesseldorf.de

ABSTRACT

Sketches—data structures for probabilistic, duplicate insensitive counting—are central building blocks of a number of recently proposed network protocols, for example in the context of wireless sensor networks. They can be used to perform robust, distributed data aggregation in a broad range of settings and applications. However, the structure of these sketches is very redundant, making effective compression vital if they are to be transmitted over a network. Here, we propose lossless compression schemes for two types of sketches, Flajolet-Martin sketches and HyperLogLog sketches. They use arithmetic coding as a basis. Analysis and simulations show that compression very close to the entropy limit can be achieved, with an algorithm that is simple enough to be employed even on very resource-constrained hardware. The proposed method outperforms existing compression schemes for Flajolet-Martin sketches by far. Furthermore, we point out some surprising parallels between compressed Flajolet-Martin and HyperLogLog sketches.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General—*Data Communications*; E.4 [**Data**]: Coding and Information Theory—*Data Compaction and Compression*; E.2 [**Data**]: Data Storage Representations

General Terms

Algorithms, Design, Performance

Keywords

Probabilistic Counting, Compression, Flajolet-Martin Sketches, HyperLogLog Sketches, Arithmetic Coding

1. INTRODUCTION

Sketches are data structures for probabilistic counting. They allow for the estimation of the number of distinct elements in a multiset, in a single pass, linear time, and very limited memory. The first such data structure that has been proposed were Flajolet-Martin sketches [6], more recently followed by a number of alternative designs, like, for example, Multiresolution Bitmaps [3], LogLog sketches [2], MinCount [7], and HyperLogLog sketches [5]; a good introduction and survey can be found in [4].

Because they can be used for duplicate insensitive distributed aggregation, sketches have recently gained some attention in the networking area. Their use has been proposed, e. g., for sensor networks [1, 11] and in vehicular ad-hoc networks (VANETs) [10]. A discussion of a broader range of applications can be found in [9], where the gathering of approximate global state in a general distributed system is considered.

Originally, however, sketches have been designed with database applications in mind. They allow for high algorithmic performance, but carry substantial redundancy. When sketches are transmitted over resource-constrained networks, it is immediately evident that keeping the size of the transmissions as small as possible is not only desirable, but often vital to achieve acceptable performance. Therefore, compressing the sketches for the transmission is highly desirable. In addition to saving channel bandwidth, this may also result in large energy savings: communication is much more energy-consuming than computation, so that some additional computation effort to reduce the size of the transmissions easily pays off [8].

Here, we focus on Flajolet-Martin (FM) and HyperLogLog (HLL) sketches. The former are well-known and often used, because of their simple structure, easy tractability, and high accuracy. HLL sketches, on the other hand, are a very recent development. Though slightly more complex, they can be considered the most efficient existing data structure for probabilistic duplicate-insensitive counting, especially when it comes to estimating large cardinalities. Thus, they constitute an interesting alternative to FM sketches for certain applications.

Based on probabilistic models for these two data structures, we introduce near-entropy-optimal, but nevertheless surprisingly simple lossless compression schemes. This compression is immediately applicable wherever the sketches are transmitted over a resource-constrained communication channel. Our approach is based on arithmetic coding [13], requiring constant memory and linear time. We will present evaluation results which do not only underline the effectiveness of the proposed method, but also show interesting parallels between the two considered data structures.

The remainder of this paper is structured as follows. We first introduce FM and HLL sketches in the following section. Then we give an overview of some networking applications that use sketches and also discuss related work in the area of sketch compression in Section 3. Thereafter, in Section 4, we model the data structures probabilistically, and, based on these insights, introduce our compression schemes. A discussion and evaluation follows in Section 5, before we finally summarize and conclude our paper in Section 6.

2. INTRODUCTION TO SKETCHES

2.1 Flajolet-Martin Sketches

An FM sketch is a data structure that represents an estimate of the number of distinct elements in a multiset. It is a bit array $S = s_1, \ldots, s_w$ of fixed length $w \geq 1$. All bits are initially set to zero. To add (or "count") an element x, it is hashed by a hash function h with geometrically distributed positive integer output, such that

$$P(h(x) = i) = 2^{-i}. \quad (1)$$

The entry $s_{h(x)}$ is then set to one. (With probability 2^{-w} we have $h(x) > w$; then, no operation is performed.) A hash function with the necessary properties can be derived from a binary random hash function by using the position of the first 1-bit in the output string as the hash value.

The central result of [6] is that an estimate for the number of distinct elements added to the sketch—i. e., the cardinality of the multiset—can be obtained from the length of the initial uninterrupted block of ones in the bit field. This *indicator* is here denoted by

$$Z(S) := \min(\{i \mid 0 < i \leq w \wedge s_i = 0\} \cup \{w+1\}) - 1. \quad (2)$$

Based on $Z(S)$, the estimate of the number of distinct elements added to the sketch can be obtained by calculating

$$C(S) := 2^{Z(S)}/\rho, \quad (3)$$

where $\rho \approx 0.775351$. From this formula it also becomes clear that the size of the bit field w needs to grow logarithmically with the range of cardinalities that can be estimated by the sketch.

As an example, let us use a sketch to estimate the cardinality of the multiset $\{b, c, b, a, b, e, d\}$. The correct cardinality is 5. Assume that $h(b) = h(c) = h(e) = 1$, $h(d) = 2$, and $h(a) = 4$. Consequently, after all elements have been hashed, the first, second, and fourth bit of the sketch will be set, i. e., the sketch is $1101000\ldots$. Therefore, the length of the uninterrupted block of ones on the left hand side is $Z(S) = 2$, and the cardinality estimate is $2^2/\rho = 5.159 \approx 5$.

Although it worked out well in our example, in general, the variance of $Z(S)$ is quite high, and thus the accuracy of the estimate is not very good. Trading off accuracy against memory, a set of sketches can be used to improve the estimate's quality. This technique is called Probabilistic Counting with Stochastic Averaging (PCSA) in [6]. Each element is added to one of the sketches, determined by an equidistributed hash function. Let m sketches be used, $S_1, \ldots, S_m$, and let the indicator Z be defined as

$$Z := \sum_{j=1}^{m} Z(S_j). \quad (4)$$

Then the estimate for the cardinality—the total number of distinct items—is

$$C_{FM}(S_1, \ldots, S_m) := m \cdot 2^{Z/m}/\rho. \quad (5)$$

A single sketch is the special case with $m = 1$. Therefore, when we talk about FM sketches in the following, we mean PCSA sets of FM sketches, unless otherwise stated.

PCSA yields a standard error of approximately $0.78/\sqrt{m}$ for a sufficiently large number of elements (in [6] it is stated that C should be at least $10 \cdot m$). Hence, with $m = 64$ an accuracy of approximately 10 % is achieved, or 5 % with $m = 256$. Depending on the requirements of the application, the number of used sketches may be substantial, making effective compression vital.

Unfortunately, however, (5) gives bad estimates for small Z/m. This becomes immediately clear when looking at the minimum value: Z is non-negative, thus C_{FM} is never below m/ρ. We thus use the following modified version of (5):

$$C(S_1, \ldots, S_m) := m \cdot \frac{2^{Z/m} - 2^{-\kappa \cdot Z/m}}{\rho}. \quad (6)$$

We obtained very good results with $\kappa \approx 1.75$. Observe that for $Z \gg m$, (6) converges quickly to (5). Hence the asymptotic properties of (5) shown in [6] likewise hold for (6).

Two FM sketches can be merged, yielding a sketch for the union of the multisets. This property is exploited for duplicate insensitive distributed aggregation in the proposed networking applications. The merging is simply accomplished by calculating the bit-wise logical OR operation of the sketches.

2.2 HyperLogLog Sketches

HyperLogLog sketches [5] are an improved variant of LogLog sketches [2]. In contrast of FM sketches, the size of which grows logarithmically with the maximum cardinality that can be estimated, HyperLogLog sketches grow with its log log.

HLL sketches use *m registers* $M_1, \ldots, M_m$, each w bits long. w is typically very small, around 5. Similar to the respective parameters for FM sketches, m determines the accuracy of the estimation, while the covered cardinality range essentially depends on w. Where in FM sketches every sketch in a PCSA set is a bit field, in HLL sketches every register stores an integer in the range between 0 and $2^w - 1$. They are all initialized to 0. To add an element, the basic procedure is initially similar to a PCSA set of FM sketches. One of the registers is chosen by an equidistributed hash function, and a geometrically distributed value between 1 and $2^w - 1$ is determined by another hash function like in (1). The selected register is then set to the maximum of its current and this new value.

To read the estimate from a HyperLogLog sketch, again an indicator is calculated. Here, it is

$$Z := \left(\sum_{j=1}^{m} 2^{-M_j} \right)^{-1}. \quad (7)$$

The cardinality estimate is then given by

$$C := \alpha_m \cdot m^2 \cdot Z, \quad (8)$$

where

$$\alpha_m := \left(m \int_0^\infty \left(\log_2 \frac{2+u}{1+u} \right)^m du \right)^{-1}. \quad (9)$$

The values of α_m are all significantly below 1; for all reasonable values of m, it is between 0.7 and 0.73.

For small cardinalities in the order of m or below, HLL sketches exhibit a similar problem as FM sketches. Observe that the smallest value Z can take is $1/m$, and therefore the estimate will always be greater or equal to $\alpha_m m$; since α_m is typically around 0.7, this is in the order of m. This problem has also been recognized in [5], where correction formulas for both very small and very large cardinalities are given. For the small range correction, [5] suggests to use the number of registers whose value is zero as a second, additional indicator. This number is denoted by V. If the estimate C from (8) is less than $5m/2$, the corrected estimate

$$C^* := m \cdot \log(m/V) \tag{10}$$

is used instead of C. We will use this correction mechanism here.[1]

The (asymptotic) relative error of HyperLogLog sketches according to [5] is $1.04/\sqrt{m}$. An HLL sketch thus needs an approximately $(1.04/0.78)^2 \approx 1.78$ times higher m to achieve the same accuracy as FM sketches. While this seems worse at a first glance, it has to be taken into account that the respective values of m and w are not immediately comparable: the m registers in case of a HyperLogLog sketch take significantly less space than the the same number of sketches in an FM sketch PCSA set would.

Like FM sketches, HLL sketches can be used for duplicate insensitive distributed aggregation. Two HLL sketches are merged by calculating the register-wise maximum.

3. RELATED WORK

3.1 Networking Applications of Sketches

A number of applications have been proposed that involve the transmission of sketches over resource-constrained communication channels. The first such use of FM sketches has been proposed in the context of wireless sensor networks. A scheme applying Flajolet-Martin sketches is described in [1], similar ideas in a more general discussion can be found in [11]. Both consider a situation where sensor readings are collected towards a sink with in-network data aggregation. A typical example is a query for the average measured temperature in an area.

The classical approach is to construct an aggregation tree, i. e., a routing tree rooted at the sink. The sensor readings are communicated towards the source along the edges of this tree. Where multiple branches meet in an intermediate node, this node combines the values from all these branches into one single, aggregated value. In [1, 11], it is pointed out that a communication failure along one link results in losing the data from a whole subtree. Their basic idea is to make the data collection more robust by using multiple, redundant paths from the source nodes to the sink. The value read from each sensor may then arrive via multiple different paths at the source. In-network aggregation with duplicate sensitive aggregates like sums or averages is then, however, no longer possible: many sensor readings would end up being incorporated multiple times into the final result.

Using aggregates based on duplicate insensitive sketches overcomes this problem in a very elegant way: the sensors, instead of transmitting their measured value directly, generate a sketch representing the value (essentially by adding a respective number of random elements). These sketches, on their way towards the sink, can then be arbitrarily combined. This technique can be used for a variety of aggregates, including, e. g., sums, averages, variance and standard deviation.

In [10], we propose a hierarchical data aggregation scheme for vehicular ad-hoc networks, based on FM sketches. It is applicable, for example, in distributed, cooperative traffic information systems or for the dissemination of information on free parking places. In the considered applications, data like the current traffic situation or the number of free parking places on a road segment is gathered cooperatively by the vehicles themselves. It is then shared in the network. To respect the capacity constraints of such a network, the data is aggregated hierarchically: fine-grained information is made available in the near vicinity, while coarser aggregates are distributed within a larger surrounding. In such a setting, duplicate insensitivity is again vital, and it can again be achieved by using sketch-based aggregates. But in contrast to the sensor network applications, the data is here not transmitted towards a sink: there are no "queries" and thus also no "replies" to queries. The information is instead proactively disseminated in the network. Thus, the problem of removing old information arises. To solve this, the FM sketches are modified to obtain a soft-state variant, where the contained information decays over time.[2]

3.2 Compression of Sketches

Reducing the size of sketches is generally perceived as a vital building block for applications that involve their transmission. However, not much effort has so far been put into their compression. Two heuristic schemes for FM sketches have been proposed. We discuss them here, and use them for comparison purposes later.

In [12], it is suggested to write the PCSA set as a matrix, where each sketch forms one row. This matrix is then read column-wise, which results in the sketches being interleaved: first the leading bits of all sketches, followed by all the second bits, and so on. A long sequence of ones can be expected in the beginning. The length of this sequence of ones followed by the (uncompressed) remainder of the sketch are then transmitted.

In [1], the idea from [12] has been taken up and is adapted. The same interleaving scheme is combined with run-length encoding of the whole set of sketches. This scheme is therefore able to deal quite well with the outer regions of the sketch, where long sequences of ones and zeroes dominate, but it is not well suited for the middle region.

4. SKETCH COMPRESSION

We will now look at the structure and the probability distributions of the sketch entries in detail. This will lead to analytical expressions for the entropy of FM and HLL sketches on the one hand, and to near-entropy-optimal compression schemes for the two data structures on the other hand. We start with FM sketches, and subsequently show how similar concepts can be applied also to HLL sketches.

4.1 Compressing Flajolet-Martin Sketches

A closer look at FM sketches reveals that much more is known about their structure than the high probabilities of ones on the left and zeroes on the right hand side, as it has been exploited by the existing schemes. Based on the number of added distinct elements, one can calculate the probability of a bit being zero explicitly, for every bit position. Since the hash function is pseudo-random, the bits chosen for distinct elements can be considered independent. The probability of a bit being zero is the probability that all elements have been mapped to other bits. For a PCSA set of m sketches, the probability of an element being mapped to the i-th bit of one specific sketch is $2^{-i}/m$. Consequently, if C distinct ele-

[1]The basic ideas behind the small range correction for HLL sketches can also be transferred to FM sketches, yielding an alternative to (6). However, (6) has the benefit of not requiring a second statistic on the sketch, since it is based purely on Z.

[2]While the structure of these modified sketches does not allow for an immediate application of our compression technique, they are sufficiently similar that an adaption seems possible.

ments have been added, we get

$$P(s_i = 0) = \left(1 - 2^{-i}/m\right)^C \tag{11}$$

for the i-th bit in any of the sketches.

A PCSA set can represent $mw+1$ different values of C—each corresponding to one value of the indicator Z, which is in the range $0, \ldots, mw$. This corresponds to $\log_2(mw+1)$ bits of information (assuming that all values are equally likely). For each of these values, however, there are many different sketches: many bit patterns result in the same indicator value, and thus the same estimated element count. This additional information, in the form of the exact distribution of ones and zeroes, can be interpreted as the information that allows for the duplicate insensitivity of the sketch.

Let us now get an idea about the amount of this "duplicate insensitivity information": how many bits of information are contained in a sketch, beyond the $\log_2(mw+1)$ bits for its value? This number is the Shannon entropy of the sketch given its value. It also constitutes a lower bound on the size that can be achieved by a lossless compression scheme.

For this analysis, let us assume that the bits are independent. Although there are some minor correlations (e. g., Z and the first $m-1$ sketches determine the sequence of leading ones in the m-th sketch), this is a good approximation. The probabilities of a bit being one or zero are known by (11). Thus, given C, m, and w, the entropy of a PCSA set given the number of added distinct elements C can be obtained by summing up the binary entropy functions for all single bits:

$$\begin{aligned} H_{\text{FM}} = -m \sum_{i=1}^{w} \big(& P(s_i = 0) \log_2 P(s_i = 0) \\ & + P(s_i = 1) \log_2 P(s_i = 1) \big). \end{aligned} \tag{12}$$

Evaluating H_{FM} for some values shows that the amount of additional information for duplicate insensitivity by far exceeds the few bits that are necessary to convey the value of the sketch alone. For example for $w = 16$ and $m = 256$, Z can take 4097 distinct values, and the value of the sketch can thus be conveyed in 13 bits. For the same parameters, the entropy given C can be up to about 1200 bits. Nevertheless, this is still significantly less than the uncompressed size of the sketches, which is 4096 bits in our example. This shows that there is potential for substantial compression.

The compression scheme we propose can be understood as being based on a separation of the represented value (which is, as seen above, only $\log_2(mw+1)$ bits) and the significantly larger duplicate insensitivity information. The former is negligibly small, so it is not worth spending significant effort on its compression. We simply prepend $\lceil \log_2(mw+1) \rceil$ bits to the compressed PCSA set to convey the indicator Z. Based on Z, both sender and receiver can calculate a model for the data, i. e., the probabilities (11) for each bit. A perfect compression scheme would then need H_{FM} additional bits for the duplicate insensitivity information.

We then compress the sketches bit-wise using arithmetic coding as presented in [13]. If the input probabilities are known, arithmetic coding achieves entropy-optimal compression. Furthermore, it is able to deal very well with an input probability distribution that changes for every bit.

The basic idea of arithmetic coding is to divide the interval $[0,1]$ first into sub-intervals according to the probabilities of the first input character, i. e., the first bit in our case. The sub-interval corresponding to the bit's value is then chosen. It is again subdivided, matching the probability distribution for the second bit, and so on. The sender on-the-fly generates a binary representation of a value within the finally resulting (tiny) interval. Since Z is transmitted first, the decoder also knows the used probability distribution. It can thus reconstruct the encoder's steps and thereby recovers the data.

A big benefit in particular on resource constained hardware like wireless sensor nodes is that both encoder and decoder can be implemented very efficiently with integer arithmetic. Both require only a constant, very small amount of memory. Encoding and decoding complexity are linear in the input length. The calculation of the probabilities (11) is the most complex necessary operation. Note, however, that any reasonable approximation of the exact probabilities will do fine, as long as sender and receiver use the same approximation.

4.2 Compressing HyperLogLog Sketches

The above ideas to compress FM sketches can be transferred also to HLL sketches. Here, however, register-wise compression instead of the bit-wise procedure used above is more appropriate.

We also separate the value represented by the sketch from what again can be considered the duplicate insensitivity information. There is, however, a difference to the situation with FM sketches. For HLL sketches, the indicator Z from (7) has a much larger variety of possible values. In contrast to FM sketches, transmitting the value of the indicator is therefore not significantly better than transmitting the estimate itself. Since the maximum estimate is $\alpha_m \cdot m \cdot 2^{2^w}$, this can be done using $\lceil \log_2(\alpha_m m) \rceil + 2^w$ bits. Since even for very large cardinality ranges w is small (for instance, $w = 5$ suffices for cardinalities up to $\sim 10^9$), this is a reasonable size.

In extreme cases with higher w doing this naively may, however, result in using large number of bits for conveying the cardinality. For instance, for the rather extreme case of $w = 8$, more than $2^8 = 256$ bits would be used. Note, however, that it is not useful to transmit a more accurate representation of the estimate than the precision used when calculating the probabilities of the model—thus, there is a natural limit on the number of bits that should reasonably be spent for its transmission. Instead of an exact (integer) representation of the estimate, an approximation, e. g., through a floating point number can be used. In this case, the encoder of course also has to use the probabilities that result from the approximated estimate—otherwise the receiver's probability model deviates from the sender's, which may lead to decoding errors.

Note that sending the estimate instead of the indicator for HLL sketches also avoids another problem, which would otherwise result from the small range correction according to (10). As discussed above, this correction mechanism requires information on the sketch that goes beyond the indicator Z: the number of registers with value zero. Sending Z alone would thus not suffice where the small range correction applies: the receiver would not then not always be able to reconstruct the probability model that has been used for encoding. This is inherently overcome by transmitting the estimate instead of the indicator.

Assuming—like above the independence of the FM sketches in the PCSA set—that the registers of the HLL sketch are independent, we can obtain the probability distribution of a register's value after adding C distinct elements to the sketch. The probability that a specific one of the m registers is "hit" by an added element is $1/m$. For any register M_j and $0 \le k \le 2^w - 1$ we get

$$P(M_j \le k) = \begin{cases} \left(1 - 2^{-k}/m\right)^C & \text{for} \quad 0 \le k < 2^w - 1 \\ 1 & \text{for} \quad k = 2^w - 1 \end{cases} \tag{13}$$

The similarities to (11) are obvious.

Knowing the probability distribution allows to apply arithmetic coding. It also allows for calculating the entropy given then number

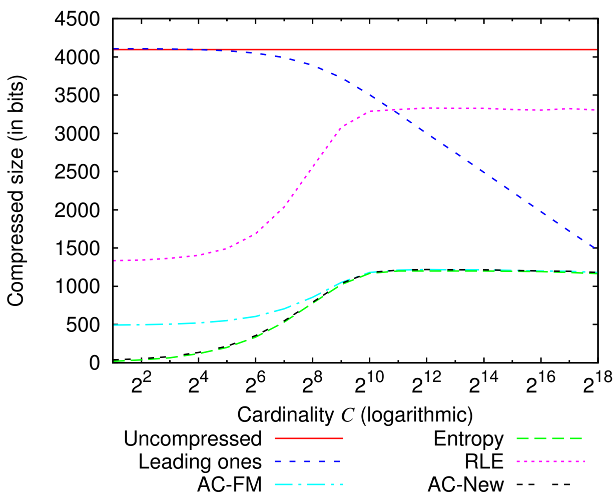

Figure 1: Compressed size of FM sketches with increasing cardinality. ($w = 16$, $m = 256$)

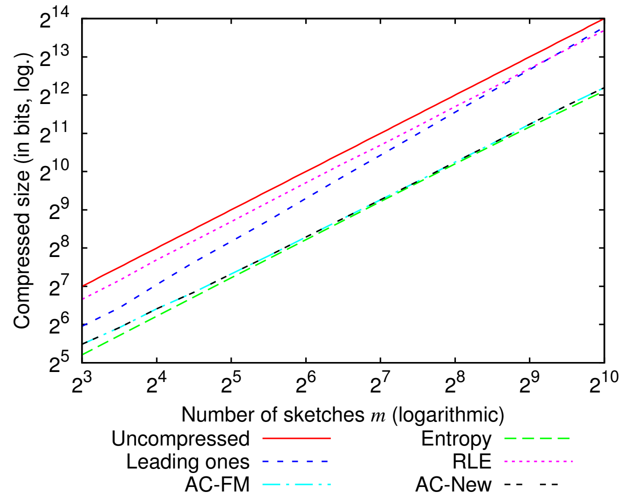

Figure 2: Compressed size of PCSA sets with an increasing number of FM sketches. ($w = 16$, $C = 2^{12}$)

of distinct added elements, which can again serve as a benchmark for the efficiency of the compression:

$$H_{\mathrm{HLL}} = -m \sum_{k=0}^{2^w-1} P(M_j = k) \log_2 P(M_j = k). \qquad (14)$$

The difference between the entropy and the uncompressed size of HLL sketches is smaller than that for FM sketches, but it is still substantial, indicating that for typical parameter choices a size reduction by at least one third should be possible.

5. EVALUATION

We have implemented the proposed arithmetic coding based compression schemes for FM and HLL sketches. For comparison purposes, we have also implemented the FM sketch compression scheme from [12] and run-length encoding over interleaved PCSA sets as used in [1]. Now, we will present some results of the practical application of these methods and discuss their implications. We start with FM sketches, and then compare the results to those obtained for HLL sketches.

5.1 Flajolet-Martin Sketches

Figure 1 shows the average compressed size of FM sketches with $w = 16$ and $m = 256$ for an increasing number of added elements. The uncompressed size is 4096 bits. The shown value range is what can reasonably be represented by sketches with these parameters; for higher cardinalities, their accuracy decreases quickly.

The compression scheme from [12] is called "leading ones" here. It obviously grows better for an increasing cardinality; this is the expected behavior, since for increasing C the number of leading ones of the interleaved sketch set also increases. For very low values in the sketch, the additional bits needed to transmit the length of the (then very short) initial sequence of ones is not made up for by the savings, so that at the left hand side of the graph the compressed size in fact minimally exceeds the uncompressed size for this scheme. "RLE" denotes the run-length encoding scheme. Performing well for small cardinalities, it degrades when the number of elements increases, because then the size of the middle region grows. The space savings in the outer regions are then partially countervailed by additional space needed where both ones and zeroes occur frequently.

"AC-FM" and "AC-New" transmit Z using $\lceil \log_2(mw+1) \rceil$ bits first and then use arithmetic coding, as proposed above. AC-FM uses the original estimator (5), while AC-New uses (6). Since for smaller element counts (6) is substantially more accurate than (5), the probabilities used for the arithmetic coding are then also much more accurate for AC-New than for AC-FM. Better probability estimates result in better compression, hence the significantly better performance of AC-New in this value range matches our expectations.

The entropy according to (12) is also shown in the graph. It is a lower bound on the achievable size with any lossless compression scheme. It is evident that the scheme proposed here comes extremely close to this theoretical limit. Where, due to the inaccurate estimates, AC-FM exceeds the entropy substantially for smaller values of C, AC-New is virtually identical to the entropy limit plus the 13 prepended bits used to convey the value of Z to the receiver.

Figure 2 shows the compressed sizes of PCSA sets along a different dimension, with increasing m (note the log-log scale). It is evident that, regardless of the compression, the size of the PCSA set increases approximately linearly with the number of sketches. This also holds for the entropy, which shows that a higher accuracy (which is achieved by increasing m) inevitably comes at the cost of an increase in size.

Since $C \gg m$ here, AC-FM and AC-New are practically indistinguishable. Again, both are very close to the entropy, pointing out once more the extremely high efficiency of the compression. Just for very small m, and thus small sizes of the uncompressed PCSA sets, the prepended bits for transmitting Z become visible.

Finally, Figure 3 shows how the size develops if both the number of elements and the number of sketches stay constant, but the number of bits used for each sketch increases. More bits in each single sketch allow for a larger value range to be represented. The uncompressed size of the PCSA set increases linearly, and likewise do the sizes of the leading ones and RLE compressed sketches, where the slope of the RLE compressed sketches is smaller. All these results could be expected considering the definitions of the respective schemes.

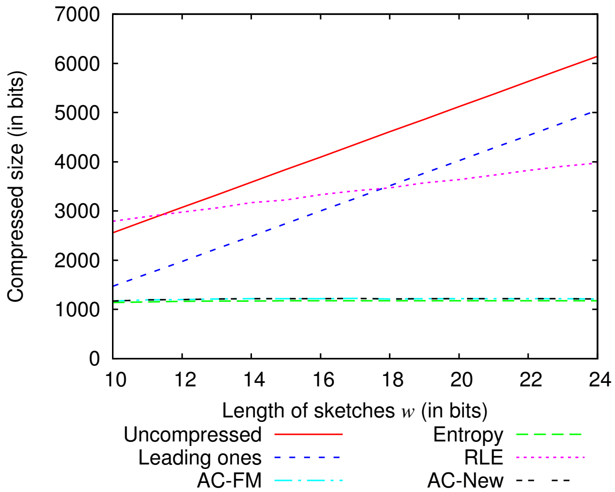

Figure 3: Compressed size of FM sketches with increasing sketch length. ($m = 256$, $C = 2^{12}$)

A look at the entropy in this graph is a little surprising at first, because it does not significantly increase with the FM sketches' length. In fact, the entropy (12) converges quickly with increasing w, so that only the number of bits required for the prepended value of Z increases if a larger value range is to be covered. The latter grows only like $\log w$. In effect, while the size of uncompressed FM sketches is in $\Theta(\log C_{\max})$ (for cardinalities up to $C_{\max}$), they need only $\Theta(\log \log C_{\max})$ space when compressed with our scheme. This is known to be the theoretically optimal scaling behavior [4].

In summary, our evaluations of the entropy as well as our simulations with real sketches show that the size of compressed sketches depends almost exclusively on the required accuracy, and thus on m. Asymptotically, the size increases with the $\log \log$ of the required capacity; in practice, however, the component for which this applies is dwarfed by the duplicate insensitivity information part, the size of which turned out to be in $\Theta(m)$, and thus depends only on the required accuracy.

5.2 HyperLogLog Sketches

Let us now look at the situation for HyperLogLog sketches. The compression scheme used here transmits the estimate in $\min\{\lceil \log_2(\alpha_m m) \rceil + 2^w, 32\}$ leading bits, according to the considerations from Section 4.2 and assuming the use of single precision (32 bit) floats. This estimate is followed by the compressed duplicate insensitivity information.

Figure 4 is the HLL sketch equivalent to Figure 1. We use $w = 4$ and $m = 455$ here, since with these parameters both the covered value range and the accuracy are equivalent to those achieved by the FM sketches in Figure 1. The uncompressed size of HLL sketches is of course smaller than that of FM sketches, 1820 versus 4096 bits in this case. The figure shows that the compression works very well, just as its counterpart for FM sketches did: the compressed size is very close to the entropy of the duplicate insensitivity information.

The biggest surprise of this evaluation, however, is revealed when comparing the entropy of the HLL sketches to that of the equivalent FM sketches in Figure 1—they are very similar. This does not only hold for the parameters chosen here, but seems to be a general property: Figures 5 and 6 show the uncompressed and compressed sizes as well as the entropy along the m- and w-dimensions,

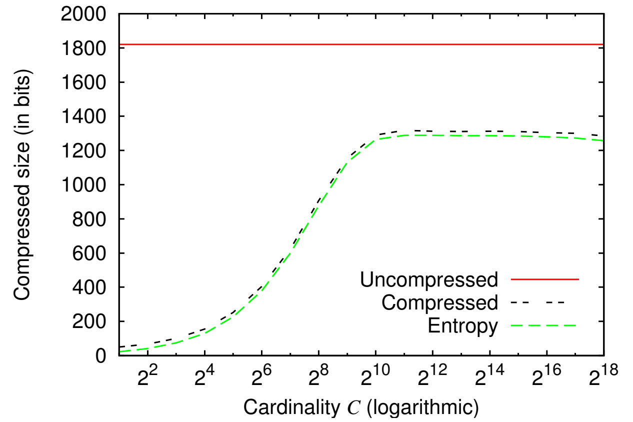

Figure 4: Compressed size of HLL sketches with increasing cardinality. ($w = 4$, $m = 455$)

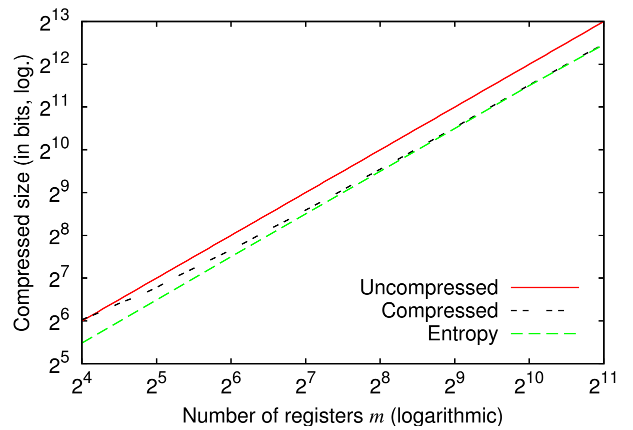

Figure 5: Compressed size of HLL sketches with an increasing number of registers. ($w = 4$, $C = 2^{12}$)

with parameters chosen for comparability with Figures 2 and 3, respectively. The entropy and thus the compressed sizes of FM and HLL sketches are, if the same value range is to be covered and the same accuracy is targeted, almost identical—with a slight size advantage of up to a few percent for the FM sketches. Therefore, if compression is applied, the choice between FM and HLL sketches is *not* influenced by the size of the transmissions.

6. CONCLUSION

In this paper we have discussed schemes for compressing two data structures for probabilistic, duplicate insensitive counting, Flajolet-Martin and HyperLogLog sketches. By calculating the entropy of the sketches explicitly and comparing it to the simulatively obtained compressed sizes we were able to show that in both cases the proposed methodology of transmitting the represented value first, using it to build a probability model of the bits (for FM sketches) or register values (for HLL sketches), and then applying arithmetic coding based on this probability model, achieves a compression ratio that practically gets extremely close to the theoretical optimum. Nevertheless, the scheme is very simple, and can be implemented in some ten lines of code. It also has very favorable complexity in terms of memory as well as computation.

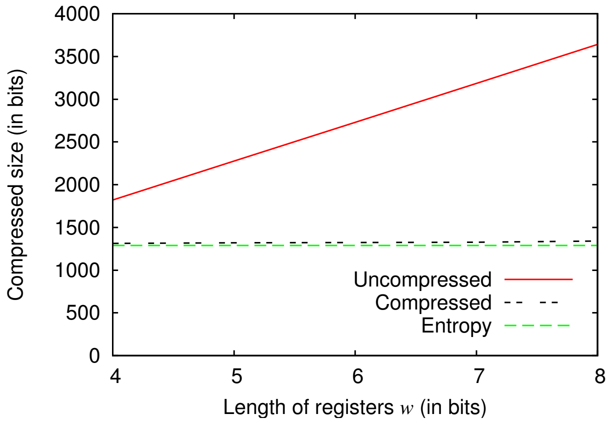

Figure 6: Compressed size of HLL sketches with increasing register length. ($m = 455$**,** $C = 2^{12}$)

Our scheme is thus of immediate benefit whenever the transmission capacity is constrained or transmissions are costly, e. g., in terms of energy. We therefore consider it an important building block for protocols using FM or HLL sketches.

Two additional interesting results—apart from the proposed compression method—could also be obtained. It turned out that even though the structure and the uncompressed sizes of FM and HLL sketches are very different, they share unexpected similarities. In particular, the entropies of FM and HLL sketches with comparable properties are almost identical. Furthermore, for both types of sketches, the amount of duplicate insensitivity information is practically independent from the covered value range of the sketch. Since its size dominates the amount of information for the actual value of the sketch by far, the covered cardinality range is *not* a limiting factor in practice, if the transmission bandwidth is the bottleneck.

Acknowledgments

We would like to thank Jeffrey Considine and Feifei Li from Boston University for their helpful feedback on the compression used in [1]. We also thank the authors of [1] as well as those of [13] for making their source code available.

7. REFERENCES

[1] J. Considine, F. Li, G. Kollios, and J. Byers. Approximate aggregation techniques for sensor databases. In *ICDE '04: Proceedings of the 20th International Conference on Data Engineering*, pages 449–460, Mar. 2004.

[2] M. Durand and P. Flajolet. Loglog counting of large cardinalities. In *ESA '03: Proceedings of the 11th Annual European Symposium on Algorithms*, pages 605–617, Sept. 2003.

[3] C. Estan, G. Varghese, and M. Fisk. Bitmap algorithms for counting active flows on high speed links. In *IMC '03: Proceedings of the 3rd ACM SIGCOMM Conference on Internet Measurement*, pages 153–166, Oct. 2003.

[4] P. Flajolet. Counting by coin tossings. In *ASIAN '04: Proceedings of the 9th Asian Computing Science Conference*, pages 1–12, Dec. 2004.

[5] P. Flajolet, É. Fusy, O. Gandouet, and F. Meunier. Hyperloglog: The analysis of a near-optimal cardinality estimation algorithm. In *AofA '07: Proceedings of the 2007 International Conference on Analysis of Algorithms*, June 2007.

[6] P. Flajolet and G. N. Martin. Probabilistic counting algorithms for data base applications. *Journal of Computer and System Sciences*, 31(2):182–209, Oct. 1985.

[7] F. Giroire. Order statistics and estimating cardinalities of massive data sets. In *AofA '05: Proceedings of the 2005 International Conference on Analysis of Algorithms*, pages 157–166, June 2005.

[8] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *HICSS '00: Proceedings of the 33rd Hawaii International Conference on System Sciences*, volume 8, page 8020, Jan. 2000.

[9] S. Keshav. Efficient and decentralized computation of approximate global state. *ACM SIGCOMM Computer Communication Review*, 36(1):69–73, Jan. 2006.

[10] C. Lochert, B. Scheuermann, and M. Mauve. Probabilistic aggregation for data dissemination in VANETs. In *VANET '07: Proceedings of the 4th ACM International Workshop on Vehicular Ad Hoc Networks*, Sept. 2007. To appear.

[11] S. Nath, P. B. Gibbons, S. Seshan, and Z. R. Anderson. Synopsis diffusion for robust aggregation in sensor networks. In *SenSys '04: Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, pages 250–262, Nov. 2004.

[12] C. R. Palmer, P. B. Gibbons, and C. Faloutsos. ANF: A fast and scalable tool for data mining in massive graphs. In *KDD '02: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 81–90, July 2002.

[13] I. H. Witten, R. M. Neil, and J. G. Cleary. Arithmetic coding for data compression. *Communications of the ACM*, 30(6):520–540, June 1987.