

Computing Frequent Elements using Gossip

Bibudh Lahiri and Srikanta Tirthapura *

Iowa State University, Ames, IA, 50011, USA
{bibudh,snt}@iastate.edu

Abstract. We present algorithms for identifying frequently occurring elements in a large distributed data set using *gossip*. Our algorithms do not rely on any central control, or on an underlying network structure, such as a spanning tree. Instead, nodes repeatedly select a random partner and exchange data with the partner – if this process continues for a (short) period of time, the desired results are computed, with probabilistic guarantees on the accuracy. Our algorithm for frequent elements is built by layering a novel small space “sketch” of data over a gossip-based data dissemination mechanism. We prove that the algorithm converges to the approximate frequent elements with high probability, and provide bounds on the time till convergence. To our knowledge, this is the first work on computing frequent elements using gossip.

1 Introduction

We are increasingly faced with data-intensive decentralized systems, such as large scale peer-to-peer networks, server farms with tens of thousands of machines, and large wireless sensor networks. With such large networks comes increasing unpredictability; the networks are constantly changing, due to nodes joining and leaving, or due to node and link failures. *Gossip* is a type of communication mechanism that is ideally suited for distributed computation on such unstable, large networks. Gossip-based distributed protocols do not assume any underlying structure in the network, such as a spanning tree, so, there is no overhead of sub-network formation and maintenance. A gossip protocol proceeds in many “rounds” and in each round, a node contacts a few randomly chosen nodes in the system and exchanges information with them. The randomization inherently provides robustness, and surprisingly, often leads to fast convergence times. The use of gossip-based protocols for data dissemination and aggregation was first proposed by Demers *et al.* [1].

We focus on the problem of identifying *frequent data elements* in a network using gossip. Consider a large peer-to-peer network that is distributing content, such as news or software updates. Suppose that the nodes in the network (or the network administrators) wish to track the identities of the most frequently accessed items in the network. The relevant data for tracking this aggregate are the frequencies of accesses of different items. However, this data is distributed

* Supported in part by NSF grant CNS 0520102 and by ICUBE, Iowa State University

throughout the network – in fact, even the number of accesses to a single item may not be available at any single point in the network. Our gossip-based algorithm for frequent elements can be used to track the most frequently accessed items in a low-overhead, decentralized manner. Another application of tracking frequent items is in the detection of a distributed denial of service (DDoS) attack, where many malicious nodes may team up to simultaneously send excessive traffic towards a single victim (typically a web server), so that legitimate clients are denied service. Detecting a DDoS attack is equivalent to finding that the total number of accesses to some server has exceeded a threshold. A distributed frequent elements algorithm can help by tracking the most frequently accessed web servers in a distributed manner, and noting if these frequencies are abnormally large. With a gossip-based algorithm this computation can proceed in a totally decentralized manner.

We consider two variants of the frequent elements problem, with absolute and relative thresholds. In the absolute threshold version, the task is to identify all elements whose frequency of occurrence is at least an absolute number (threshold), which is an user-defined parameter. In the relative threshold version, the task is to identify all elements whose frequency of occurrence is more than a certain fraction of the total size of the data, where the fraction (the relative threshold) is an user-defined parameter. In a distributed dynamic network, these two problems turn out to be rather different from each other.

Our algorithms work without explicitly tabulating the frequencies of different elements at any single place in the network. Instead, the distributed data is represented by a small space “sketch” that is propagated and updated via gossip. A sketch is a space-efficient representation of the input, which is specific to the aggregate being computed, and captures the essence of the data for our purposes. The space taken by the sketch can be tuned as a function of the desired accuracy. A complication with gossip is that since it is an unstructured form of communication, it is possible for the same data item to be inserted into the sketch multiple times as the sketch propagates. Due to this, a technical requirement is that the sketch should be able to handle duplicate insertions, i.e. it should be *duplicate-insensitive*. If the gossip proceeds long enough, the sketch can be used to identify all elements whose frequency exceeds the user defined threshold. At the same time, elements whose popularity is significantly below the threshold will be omitted (again, with high probability).

To summarize our contributions, we present the first work on computing frequent elements in a distributed data set using gossip. We present randomized algorithms for both the absolute threshold and the relative threshold versions of the problem. For each algorithm, we present a rigorous analysis of the correctness, and the time till convergence. Our analysis show that gossip-based algorithms converge quickly, and can maintain frequent elements in a network with a reasonable communication overhead. We also note that similar techniques can be used in estimating the number of distinct data items in the network.

With a gossip protocol, communication is inherently randomized, and a node can never be certain that the results on hand are correct. However, the longer

the protocol runs, the closer the results get to the correct answer, and we are able to quantify the time taken till the protocol converges to the correct answer, with high probability. Gossip algorithms are suitable for applications which can tolerate such relaxed consistency guarantees. Examples include a network monitoring application, which is running in the background, maintaining statistics about frequently requested data items, or the most frequently observed data in a distributed system. In such an application, a guaranteed accurate answer may not be required, and an approximate answer may suffice.

1.1 Related Work

Demers *et al.* [1] were the first to provide a formal treatment of gossip protocols (or “epidemic algorithms” as they called them) for data dissemination. Kempe and Kleinberg [2] analyzed the influence of the underlying gossip mechanism on the design of gossip-based protocols, and explored the limitations of uniform gossip in solving the *nearest resource location problem*. Kempe, Dobra and Gehrke [3] proposed algorithms for computing the sum, average, approximately uniform random sampling and quantiles using uniform gossip. Their algorithm for quantiles are based on their algorithm for the sum – they choose a random element and count the number of elements that are greater and lesser than the chosen element, and recurse on smaller data sets until the median is found. Thus their algorithms need many instances of “sum” computations to converge before the median is found. A similar approach could potentially be used to find frequent elements using gossip. In contrast, our algorithms for frequent elements are not based on repeated computation of the sum, and converge faster.

Much recent work [4–6] has focused on computing “separable functions” using gossip. A separable function is one that can be expressed as the sum of individual functions of the node inputs. For example, the function “count” is separable, and so is the function “sum”. However, the set of frequent elements is not a separable function. Hence, these techniques do not apply to our problem. There is much other work on the computation of basic aggregates, we list a few representative ones here. Kashyap *et al.* [7] proposed algorithms for gossip with flexible tradeoffs between the number of rounds and the number of messages transmitted. Dimakis, Sarwate and Wainwright [8] consider the problem of computing the average over *random geometric graphs* with location-aware nodes, combining uniform gossip with greedy geographic routing.

The problem of identifying frequent elements in data has been extensively studied [9–11] in the database, data streams and network monitoring communities (where frequent elements are often called “heavy-hitters”). The early work in this is due to Misra and Gries [9], who proposed a deterministic algorithm to identify frequent elements in a stream, followed by Manku and Motwani [10], who gave randomized and deterministic algorithms for tracking frequent elements in limited space. The above were algorithms for a centralized setting.

Cao and Wang [12] proposed an algorithm to find the top- k elements in a distributed setting, where they first made a lower-bound estimate for the k^{th} value, and then used the estimate as a threshold to prune away elements which

should not qualify as top- k . Zhao *et al.* [13] proposed a sampling-based and a counting-sketch-based scheme to identify globally frequent elements. Manjhi *et al.* [14] present an algorithm for finding frequent items on distributed streams, through a tree-based aggregation. Venkataraman *et al.* [15] present an algorithm for identifying “superspreaders” or “heavy distinct hitters” in a network data stream. Keralapura, Cormode and Ramamirtham [16] proposed an algorithm for continuously maintaining the frequent elements over a network of nodes. The above algorithms are not directly applicable to the problem of identifying frequent elements using gossip, because they sometimes assume the presence of a central node, or an underlying network structure such as a spanning tree [14, 16], and further, they do not consider the dissemination of sketch through gossip, or the cost of network-wide communication.

1.2 Organization of the Paper

In Section 2, we state our model and the problem more precisely. The algorithm and analysis for the case of absolute threshold in the asynchronous time model is presented in Section 3 and the case of relative threshold is presented in Section 4. In Section 5 we discuss the synchronous time model, and the computation of the number of distinct elements. Due to space constraints, we have deferred most of the proofs till the appendix, and present sketches of the proofs of some results.

2 Model

We consider a distributed system with N nodes numbered from 1 to N . The number of nodes is not necessarily known to any participating node, and this information is not used by the algorithms. Each node i holds a single data item m_i . Without loss of generality, we assume that $m_i \in \{1, 2, \dots, m\}$ is an integer representing an item identifier. For data item $v \in \{1, \dots, m\}$, the frequency of v is denoted by f_v , and is defined as the number of nodes that have data item v , i.e. $f_v = |\{j \in [N] : m_j = v\}|$. Note that f_v may not be available locally at any node, in fact determining f_v itself requires a distributed computation. The task is to identify those elements that have large frequencies. We note that though we describe our algorithms for the case of one item per node, they can be easily extended to the case when each node has not only a single item, but perhaps a (multi)set of items.

We consider the scenario of *uniform gossip*, which is the basic, and most commonly used model of gossip. Whenever a node i transmits, it chooses the destination of the message to be a random node from among all the current nodes in the system. The selection of the transmitting node is done by the distributed scheduler, described later in this section. We consider two variants of the problem, depending on how the thresholds are defined.

Absolute Threshold. The user gives an absolute frequency threshold $k > 1$ and approximation error λ ($\lambda < k$). An item v is considered a frequent item if

$f_v \geq k$, and v is an infrequent item if $f_v < k - \lambda$. Note that with a data set of N elements there may be up to N/k frequent elements according to this definition.

Relative Threshold. In some cases, the user may not be interested in an absolute frequency threshold, but may only be interested in identifying items whose relative frequency exceeds a given threshold. More precisely, given a relative threshold ϕ ($0 < \phi < 1$), approximation error ψ ($0 < \psi < \phi$), an item v is considered to be a frequent item if $f_v \geq \phi N$, and v is considered an infrequent item if $f_v < (\phi - \psi)N$. According to this definition, there may be no more than $1/\phi$ frequent items.

In a centralized setting, when all items are being observed at the same location, the above formulations of relative and absolute thresholds are equivalent, since the number of items N is known, and any absolute threshold can be converted into a relative threshold, or vice versa. However, in a distributed setting, a threshold for relative frequency cannot be locally converted by a node into a threshold on the absolute frequency, since the user in a large distributed system may not know the number of nodes or the number of data items in the system accurately enough. Thus, we treat these two problems separately.

At the end of the gossip, the following probabilistic guarantees must hold, whether for absolute or relative thresholds. Let δ be an user provided bound on the error probability ($0 < \delta < 1$).

(1) *Frequent Items are Identified.* With probability at least $(1 - \delta)$, every node reports every frequent item.

(2) *Infrequent Items are Discarded.* With probability at least $(1 - \delta)$, no node reports an infrequent item. In other words, the probability that an infrequent item is incorrectly reported by a node in the system is less than δ . Note that we present randomized algorithms, where the probabilistic guarantees hold irrespective of the input.

Time Model. We consider both asynchronous and synchronous models. In the *asynchronous* model, time is divided into non-overlapping rounds. In each round, a single source node, chosen uniformly at random out of all N nodes, transmits to another randomly chosen receiver. The time complexity is the number of rounds, or equivalently, the number of transmissions, since in each round there is only one transmission. Sections 3 and 4 discuss the asynchronous model. In the *synchronous* model, in each round, *every* node sends a message to a receiver chosen uniformly at random from among all nodes. Thus, in a single round of synchronous communication, N messages are exchanged among the nodes. We consider the synchronous model in Section 5.

Performance Metrics. We evaluate the quality of our protocols via the following metrics: the *convergence time*, which is defined as the number of rounds of gossip till convergence, and the *communication complexity*, which is defined as the number of bytes exchanged till convergence.

3 Frequent Elements with an Absolute Threshold

We now present an algorithm in the asynchronous model for identifying elements whose frequency is greater than a user specified absolute threshold k . The algorithms for the synchronous model are presented in Section 5. Let $S = \{m_i : i \in [N]\}$ denote the multi-set of all input values. The goal is to output all elements v such that $f_v \geq k$ without outputting any element v such that $f_v < k - \lambda$. We first describe the high level intuition.

Our algorithm is based on random sampling. The elements of S are sampled in a distributed manner, and the sampled elements are disseminated to all nodes using gossip – the cost of doing so is small, since the random sample is typically much smaller than the size of the population. The sampling also ensures that frequent elements are exchanged more often during the later gossip phase. Intuitively, suppose we sample each element from S into a set T with probability $1/k$. For a frequent element v with $f_v \geq k$, we (roughly) expect one or more copies of v to be included within T . Similarly, for an infrequent element u with $f_u < k - \lambda$, we expect that no copy of u will be included in T . However, some infrequent elements may get “lucky” and may be included in T and similarly, some frequent elements may not make it to T . The probabilities of these events decrease as the sample size increases.

To refine this sampling scheme, we sample with a probability that is a little larger than $1/k$, say c/k for some parameter c . Finally, we select those elements that occur at least r times within T , for some parameter $r < c$ that will be decided by the analysis. It turns out that c and r will need to depend on the approximation error λ as well as the threshold k . The smaller λ is, the greater should be the sampling probability, since we need to make a more precise distinction between the frequencies of frequent and infrequent elements. In the actual algorithm, we use a sampling probability of $\frac{12k}{\lambda^2} \ln \frac{2}{\delta}$ – note that this is $\Omega(\frac{1}{k})$ since $\lambda < k$ and hence $\frac{k}{\lambda^2} > \frac{1}{k}$.

The precise algorithm for sampling and gossip is shown in Figure 1. There are three parts to this algorithm (and all others that we describe). The first part is the *Initialization*, where each node initialized its own sketch, which is usually through drawing a random sample. The next part is the *Gossip* portion, where the nodes in the system exchange sketches with each other. The algorithm only describes what happens during each round of gossip – it is implicit that such computations repeat forever. The third part is the *Query*, where we describe how a query for frequent elements is answered using the sketch. The accuracy of the result improves as further rounds of gossip occur. Through our analysis, we give a bound on the number of rounds after which frequent elements are likely to be found at all nodes.

3.1 Analysis

We now analyze the correctness and the time complexity of the algorithm in Figure 1. Due to space constraints, all proofs are presented in the appendix, and we present a few sketches of the proofs here.

Input: Data item m_i , error probability δ , frequency threshold k , approximation error λ

1. **Initialization**

- (a) Choose ρ as a uniformly distributed random number in $(0, 1)$.
- (b) If $\rho < \frac{12k}{\lambda^2} \ln \frac{2}{\delta}$ then $S_i \leftarrow \{(i, m_i)\}$, else $S_i \leftarrow \Phi$ /* null set */

2. **Gossip**

In each round of gossip:

- (a) If sketch S_j received from node j then $S_i \leftarrow S_i \cup S_j$
- (b) If node i is selected to transmit, then select node j uniformly at random from $\{1, \dots, N\}$ and send S_i to j

3. **Query**

When asked for the frequent elements, report all data items which occur more than $r = \frac{12k^2}{\lambda^2} (1 - \frac{\lambda}{2k}) \ln \frac{2}{\delta}$ times in S_i as frequent elements.

Fig. 1. Gossip algorithm at node i for finding the frequently occurring elements with an absolute threshold k

Lemma 1. False Negative. *If v is an element with $f_v \geq k$, then with probability at least $1 - \delta$, v is returned as a frequent element by every node after $20N \ln N$ rounds.*

Sketch of proof: A false negative can occur in one of two ways. (1) Either too few copies of v were sampled during initialization or (2) The sampled copies of v were not disseminated to all nodes during the gossip. We show that the first event is unlikely by an analysis of the sampling process using Chernoff bounds. We show that the second event is also very unlikely through an analysis of the asynchronous gossip in Lemma 4. \square

Lemma 2. False Positive. *If u is an element with $f_u \leq k - \lambda$, where $k^{\frac{3}{4}} \leq \lambda < k$, then the probability that u is returned by some node as a frequent element is no more than δ .*

Sketch of proof: A false positive can occur if both the following events occur: (1) r or more copies of u were sampled initially and (2) all r copies of u reach some node in the network through gossip. We show that the first event is very unlikely, if $f_u \leq k - \lambda$, and hence the intersection of the events is also unlikely. \square

We have shown that (with probability at least $1 - \delta$), a data item with k or more occurrences is returned as a frequent element by every node; and a data item with less than $k - \lambda$ occurrences is not reported by any node as a frequent item. It is now natural to ask what happens to an element whose frequency of occurrence falls in the range $[k - \lambda, k)$. Let us refer to the frequency range of length λ , $[k - \lambda, k)$ as the “grey range”, to portray the uncertainty. For elements with frequency in the grey range, our algorithm provides no guarantee - they could be reported, or not. Clearly, a smaller value of λ is desirable, but this comes

at the cost of increased sampling probability, and hence greater communication complexity of gossip. For example, suppose that $k = 10^8$ and $\lambda = 5 \times 10^6$. This implies a 5 percent approximation error with respect to k . All elements with frequency greater than 10^8 will be reported (w.h.p) and all elements with frequency below 9.5×10^7 will not be reported, and the sampling probability is approximately $4.8 \times 10^{-5} \times \ln \frac{2}{\delta}$. This is the fraction of input items that are gossiped through the network in finding the frequent elements in the distributed data set.

Analysis of the Gossip. We now shift our attention to the gossip mechanism itself. Let \mathcal{T} denote the multi-set of all items sampled during initialization $\mathcal{T} \subseteq S$ and $|\mathcal{T}| \leq N$. Consider a single sampled item $\theta \in \mathcal{T}$. Let T_θ be defined as the number of rounds till θ has been disseminated to all nodes in the network.

Lemma 3. $E[T_\theta] = 2N \ln N + O(N)$.

Sketch of proof: Let ξ_t be the set of nodes that have θ after t rounds. Thus ξ_0 has only one node (the one that sampled θ during the initialization step). For $t = 1 \dots N - 1$, let random variable X_t be the number of rounds required to increase $|\xi|$ from t to $t + 1$. We can write $T_\theta = X_1 + X_2 + \dots + X_{N-1}$. By noting that each X_t is a geometric random variable and using linearity of expectation we can arrive at the desired result. Further details are in the appendix. \square

Our proof for high-probability bounds on T_θ use the following result about a sharp concentration for the *coupon collector* problem. Suppose there are coupons of M distinct types, and one has to draw coupons (with replacement) at random until at least one coupon of each type has been collected. Initially, it is very easy to select a type not yet chosen, but as more and more types get chosen, it becomes increasingly difficult to get a coupon of a type not yet chosen. The following result can be found in standard textbooks (for example, Motwani and Raghavan [17]).

Theorem 1 (Folklore). *Let the random variable \mathcal{C} denote the number of trials to collect at least one coupon of each of M types. Then, for any constant $c \in \mathcal{R}$, $\lim_{M \rightarrow \infty} \Pr[\mathcal{C} > M \ln M + cM] = 1 - e^{-e^{-c}}$.*

Lemma 4. $\lim_{N \rightarrow \infty} \Pr(T_\theta > 20N \ln N) = O(\frac{1}{N^2})$

Sketch of proof: The dissemination of θ by gossip can be divided into two phases. The first phase starts when θ is at a single node and continues until it has reached $\frac{N}{2}$ distinct nodes. The second phase starts after θ has reached $\frac{N}{2}$ nodes and continues until it reaches N nodes. In the first phase, in each round of gossip, it is less likely to find a source node that has θ and at the same time, it is more likely to find a destination that does not have θ . Once θ has reached $\frac{N}{2}$ nodes, the situation reverses. We analyze the number of rounds required for these two phases separately. For each phase, we bound the random variable that defines the number of rounds in the phase by a simpler random variable that can be analyzed with the help of a coupon-collector type of argument. Combining the results from the two phases yields the desired result. \square

For an item v , let T_v denote the number of rounds required to disseminate all copies of v to all nodes.

Lemma 5. $\lim_{N \rightarrow \infty} \Pr[T_v > 20N \ln N] = O(\frac{1}{N})$

Proof. The proof follows from Lemma 4 using a union bound, and the fact that there are no more than N copies of v .

Lemmas 1, 2 and 5 together lead to the following theorem about the correctness of the algorithm.

Theorem 2. *Suppose the distributed algorithm in Figure 1 is run for $20N \ln N$ rounds. Then, with probability at least $1 - \delta$, any data item with k or more occurrences will be identified as a frequent element at every node. With probability at least $1 - \delta$, any data item with less than $k - \lambda$ occurrences will not be identified as a frequent element at any node.*

Communication Complexity. We next analyze the communication complexity, i.e. the number of bytes transmitted during the gossip. During the algorithm, the sizes of the messages exchanged start from one item and grow as the algorithm progresses. To avoid the complexity of dealing with different message sizes, we separately analyze the total number of bytes contributed to gossip by each sampled item, and add these contributions together. Consider any sampled item θ . We assume that transmitting any item (i, m_i) takes a constant number of bytes. Let random variable \mathcal{B} denote the number of bytes it takes to disseminate θ among all nodes.

Lemma 6. $E[\mathcal{B}] = O(N \ln N)$

Sketch of proof: Let ξ_t be the set of nodes that have θ after t rounds. In each round of gossip, θ may or may not be transmitted. Further each time θ is transmitted, $|\xi|$ increases only if the destination of the message is a node which is not already in ξ_t . We analyze \mathcal{B} as the number of transmissions of θ till ξ includes all nodes. The details of the proof, using conditional probabilities, are presented in the appendix. \square

We can similarly get a high probability bound on \mathcal{B} (proofs in appendix).

Lemma 7. $\Pr[\mathcal{B} > 3N \ln N] = O(\frac{1}{N^2})$.

Let \mathcal{Y} denote the total number of bytes that need to be exchanged for the whole protocol until the frequent elements have been identified. By combining Lemma 7 with an estimate on the number of sampled items, we get the following result about the communication complexity of the algorithm in Figure 1.

Theorem 3 (Communication Complexity for Absolute Threshold). *With high probability,*

$$\mathcal{Y} = O\left(\frac{N^2 k}{\lambda^2} \ln \frac{1}{\delta} \ln N\right)$$

4 Frequent elements with Relative Threshold

Given thresholds ϕ and ψ , where $\psi < \phi$, the goal is to identify all elements v such that $f_v \geq \phi N$ and no element u such that $f_u < (\phi - \psi)N$. Unlike the case of absolute threshold, there is no fixed probability that a node can use to sample data elements locally. For the same relative frequency threshold, the absolute frequency threshold (ϕN), as well as the approximation error (ψN) increases with N . Thus if ϕ and ψ are kept constant and N increases, then a smaller sampling probability will suffice, because of the analysis in 3. Since we do not have prior knowledge of N , we need a more “adaptive” method of sampling where the sampling probability decreases as more elements are encountered during gossip.

To design our sketch, we use an idea similar to *min-wise independent permutations* [18]. Each data item $m_i, i = 1 \dots N$ is assigned a weight w_i , which is a random number in the unit interval $(0, 1)$. The algorithm maintains a sketch T of (m_i, w_i) tuples that have the t smallest weights w_i . The value of t can be decided independent of the population size N . The intuition is that if an element v has a large relative frequency, then v must occur among the tuples with the smallest weight. Maintaining these minimum-weight elements through gossip is easy, and if we choose a large enough sketch, the likelihood of a frequent element appearing in the sketch a sufficient number of times is very high. We identify a value m as a frequent item if there are at least $(\phi - \frac{\psi}{2})t$ tuples in T with $m_i = m$; otherwise, m is not identified as a frequent element. The algorithm for the asynchronous model is described in Figure 2. The threshold t is chosen to be $O(\frac{1}{\psi^2} \ln(\frac{1}{\delta}))$.

4.1 Analysis

The proofs of most of the following lemmas appear in the appendix. Let τ denote the t^{th} minimum element among the N random values $\{w_i, i = 1 \dots N\}$. The next lemma shows that τ is sharply concentrated around $\frac{t}{N}$.

Lemma 8. For $t = \frac{128}{\psi^2} \ln(\frac{3}{\delta})$, τ satisfies the following properties: (1) $\Pr[\tau < \frac{t}{N}(1 - \frac{\psi}{4})] < \frac{\delta}{3}$ and
(2) $\Pr[\tau > \frac{t}{N}(1 + \frac{\psi}{4})] < \frac{\delta}{3}$

We now present a bound on the dissemination time of the smallest weights. Let T_t denote the time taken for the t smallest weights to be disseminated to all nodes.

Lemma 9. $\Pr[T_t > 20N \ln N] \leq O(\frac{1}{N})$.

Proof. The proof follows by using the union bound along with Lemma 4.

The following lemmas provide upper bounds on the probabilities of finding a false negative and a false positive respectively, by the algorithm described in Figure 2.

Input: Data item m_i ; error probability δ , relative frequency threshold ϕ , approximation error $\psi < \phi$

1. **Initialization:**

- (a) $t \leftarrow \frac{128}{\psi^2} \ln(\frac{3}{\delta})$
- (b) Choose w_i as a uniformly distributed random number in the real interval $(0, 1)$;
set $S_i \leftarrow \{(m_i, w_i)\}$

2. **Gossip**

In each round of gossip:

- (a) If sketch S_j is received from node j then
 - i. $S_i \leftarrow S_i \cup S_j$
 - ii. If $|S_i| > t$ then retain t elements of S_i with smallest weights
- (b) If node i is selected to transmit, then select node j uniformly at random and send S_i to j

3. **Query**

When queried for the frequent elements, report every value v such that at least $(\phi - \frac{\psi}{2})t$ (value, weight) tuples exist in S_i with value equal to v

Fig. 2. Gossip algorithm at node i for finding the frequently occurring elements with a relative threshold

Lemma 10. *Suppose the distributed algorithm in Figure 1 is run for $20N \ln N$ rounds. Then, if v is a frequent element, i.e. $f_v \geq \phi N$, then with probability at least $1 - \delta$, v is identified by every node as a frequent element.*

Sketch of proof: Two events need to happen for v to be recognized as a frequent element. (1) Enough copies of v must occur among the t smallest weights, and (2) The t smallest weight elements must be disseminated to all nodes via gossip. In the full proof, we show that both these events are very likely. \square

Lemma 11. *Suppose the distributed algorithm in Figure 1 is run for $20N \ln N$ rounds. If u is an infrequent element, i.e. $f_u < (\phi - \psi)N$, then, with probability at least $1 - \delta$, u is not identified by any node as a frequent element.*

Sketch of proof: A false positive can happen if both the following events occur. (1) There are an unusually high number of copies of u among the elements with the τ smallest weights, and (2) all these copies are disseminated to all nodes. We show that the first event is highly unlikely, and so is the probability of a false positive. \square

Combining Lemmas 11, 10 and 9 we get the following theorem.

Theorem 4. *Suppose the distributed algorithm in Figure 2 is run for $20N \ln N$ rounds. Then, with probability at least $1 - \delta$, any data item with ϕN or more occurrences will be identified as a frequent item at every node. Similarly, with probability at least $1 - \delta$, any data item with less than $(\phi - \psi)N$ occurrences will not be identified as a frequent item at any node.*

Since the size of the sketch at any time during gossip is at most t , we get the following result on the communication complexity, using Lemma 9.

Theorem 5. *The number of bytes exchanged by the algorithm in Figure 2 till the frequent elements are identified is at most $O(\frac{1}{\sqrt{\delta}} \ln(\frac{1}{\delta})N \ln N)$, with probability at least $1 - O(\frac{1}{N})$.*

5 Extensions

5.1 Synchronous Model

We now focus on the synchronous communication model. In the synchronous model, all nodes transmit equally often, and in each communication round, every node can send a message to one other (randomly chosen) node. We use a result due to Frieze and Grimmett [19], who considered the time to spread a rumor in a network. In their model, there is a rumor message that has to spread to everyone in a population of size N . Initially, a single person has the rumor. In every communication round, each person who already has the rumor conveys it to another randomly chosen person in the population, and we are interested in the number of rounds taken for the rumor to spread to all N nodes. Note the similarity to our model of synchronous gossip.

Theorem 6 (Frieze and Grimmett 1985). *Let S_N denote the number of rounds required to spread a rumor among a population of size N . Then, (1) $\lim_{N \rightarrow \infty} \frac{S_N}{\log N} = \ln 2$ with high probability and (2) For $\gamma > 0$, $\Pr[S_N > (1 + (\gamma + 1) \ln 2) \log N] = o(N^{-\gamma})$*

Suppose that instead of a single rumor, there were m different rumors originating at different nodes, and all these rumors were being disseminated simultaneously among the N nodes. Let T_m be the number of rounds required for all the nodes to receive all m rumors.

Lemma 12. *With probability $1 - o(\frac{1}{N})$, $T_m \leq (1 + 2 \ln 2) \log N + \ln m$.*

Proof. For $i = 1 \dots m$, let t_m denote the number of rounds required to disseminate rumor i . Since all the m rumors are being disseminated simultaneously, we have $T_m = \max_{i=1}^m t_i$. Using the union bound:

$$\Pr[T_m > x] = \Pr\left[\bigcup_{i=1}^m (t_i > x)\right] \leq \sum_{i=1}^m \Pr[t_i > x] = m \Pr[S_N > x]$$

Using $\gamma = 1 + \log_N m$ in Theorem 6, we get $\Pr[S_N > (1 + 2 \ln 2) \log N + \ln m] = o(\frac{1}{Nm})$, and the result follows.

Our algorithms for the synchronous time model for absolute and relative thresholds, are described in the appendix (in Figures 4 and 5 respectively). These differ from the algorithms for the asynchronous models (Figures 1 and 2)

in that in every round of communication, every node sends a message. Note that the sampling probability, the sketch size and the thresholds for identification of frequent elements in the algorithms for the asynchronous model also suffice for the synchronous model, so the analysis of the random sampling is the same as in the asynchronous model. The only change is in the gossip mechanism. We arrive at the following result, the proofs are in the appendix.

Theorem 7 (Synchronous gossip). *If the synchronous algorithms in Figures 4 and 5 are run for $(1 + 3 \ln 2) \log N$ rounds, then all frequent items (with absolute and relative thresholds, respectively) will be identified with probability at least $1 - \delta$, and no infrequent item will be identified, with probability at least $1 - \delta$.*

Note that for both absolute and relative thresholds, the number of rounds required in the synchronous model is less than that required by the asynchronous model by a factor of $\theta(N)$ – this is to be expected, since in each round in the asynchronous model, a single message is exchanged while in each round in the asynchronous model, N messages are exchanged.

5.2 The Number of Distinct Elements

The number of distinct elements in a set of values, sometimes called the zeroth frequency moment of the set [20], is an important statistic, useful in various applications in network monitoring and data mining (see [21] for a survey of its applications). We now note that it is possible to apply an algorithm due to Bar-Yossef et al. [22] to get an algorithm for estimating the number of distinct elements in a distributed data set through gossip. The algorithm provides the following guarantee. For user provided approximation error $\epsilon \in (0, 1)$ and error probability $\delta \in (0, 1)$, every node i should have an (ϵ, δ) -approximate estimate $\tilde{F}_0^{(i)}$ of F_0 , such the condition $\Pr[\frac{|F_0 - \tilde{F}_0^{(i)}|}{F_0} \leq \epsilon] \geq 1 - \delta$ is satisfied for every i . The algorithm is described for the asynchronous model in Figure 3.

We use a hash function $h(\cdot)$ which returns an output that is assumed to be uniformly distributed in $(0, 1)$; further the outputs of the hash function on different inputs are assumed to be independent of each other. It is possible to modify the analysis to work under more realistic assumptions of hash functions with limited independence (see [20–22]). The intuition is that if there are F_0 distinct elements, then we can expect the minimum of the mapped hash values to be close to $\frac{1}{F_0}$ with high probability. Bar-Yossef et al [23] showed that if v is the k^{th} minimum among all the hashed values for $k = O(\frac{1}{\epsilon^2})$, then $\tilde{F}_0 = \frac{k}{v}$ is a good estimator of F_0 . By taking the median of $O(\log \frac{1}{\delta})$ such estimators, it is possible to get an (ϵ, δ) estimator for F_0 . The algorithm for node i is described in Figure 3. For asynchronous gossip, $20N \ln N$ rounds of gossip suffice, with high probability (Lemma 5). For synchronous gossip, $(1 + 3 \ln 2) \log N$ rounds of gossip suffice, with high probability (Theorem 7).

Input: Data item m_i ; threshold parameters ϵ, δ ; random hash function h

1. **Initialization**
 $S_i \leftarrow \{h(m_i)\}$
 2. **Gossip**
Repeat forever
 - (a) If sketch S_j received from node j then
 - i. $S_j \leftarrow k$ minimum elements of $S_i \cup S_j$ ($k = O(1/\epsilon^2)$)
 - (b) If the clock ticks at this node, then select node j uniformly at random and send S_i to j
 3. **Query**
If asked for F_0 , report $\tilde{F}_0^{(i)} = \frac{k}{\text{maximum element in } S_i}$ as estimate of F_0
-

Fig. 3. Gossip algorithm for finding F_0 from a network

References

1. Demers, A.J., Greene, D.H., Hauser, C., Irish, W., Larson, J., Shenker, S., Sturgis, H.E., Swinehart, D.C., Terry, D.B.: Epidemic algorithms for replicated database maintenance. In: Proceedings of the Principles of Distributed Computing (PODC). (1987) 1–12
2. Kempe, D., Kleinberg, J.M.: Protocols and impossibility results for gossip-based communication mechanisms. In: Proceedings of the 43rd Symposium on Foundations of Computer Science (FOCS). (2002) 471–480
3. Kempe, D., Dobra, A., Gehrke, J.: Gossip-based computation of aggregate information. In: Proceedings of the 44th Symposium on Foundations of Computer Science (FOCS). (2003) 482–491
4. Boyd, S.P., Ghosh, A., Prabhakar, B., Shah, D.: Gossip algorithms: design, analysis and applications. In: Proceedings of the IEEE Conference on Computer Communications (INFOCOM). (2005) 1653–1664
5. Boyd, S.P., Ghosh, A., Prabhakar, B., Shah, D.: Randomized gossip algorithms. *IEEE Transactions on Information Theory* **52**(6) (2006) 2508–2530
6. Mosk-Aoyama, D., Shah, D.: Computing separable functions via gossip. In: Proceedings of the Twenty-Fifth Annual ACM Symposium on Principles of Distributed Computing (PODC). (2006) 113–122
7. Kashyap, S.R., Deb, S., Naidu, K.V.M., Rastogi, R., Srinivasan, A.: Efficient gossip-based aggregate computation. In: Proceedings of the Twenty-Fifth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS). (2006) 308–317
8. Dimakis, A.G., Sarwate, A.D., Wainwright, M.J.: Geographic gossip: efficient aggregation for sensor networks. In: Proceedings of the Fifth International Conference on Information Processing in Sensor Networks (IPSN). (2006) 69–76
9. Misra, J., Gries, D.: Finding repeated elements. *Science of Computer Programming* **2**(2) (1982) 143–152
10. Manku, G.S., Motwani, R.: Approximate frequency counts over data streams. In: Proceedings of 28th International Conference on Very Large Data Bases (VLDB). (2002) 346–357

11. Karp, R.M., Shenker, S., Papadimitriou, C.H.: A simple algorithm for finding frequent elements in streams and bags. *ACM Trans. Database Syst.* **28** (2003) 51–55
12. Cao, P., Wang, Z.: Efficient top-k query calculation in distributed networks. In: *Proceedings of the Twenty-Third Annual ACM Symposium on Principles of Distributed Computing (PODC)*. (2004) 206–215
13. Zhao, Q., Ogihara, M., Wang, H., Xu, J.: Finding global icebergs over distributed data sets. In: *Proceedings of the Twenty-Fifth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS)*. (2006) 298–307
14. Manjhi, A., Shkapenyuk, V., Dhamdhere, K., Olston, C.: Finding (recently) frequent items in distributed data streams. In: *Proceedings of the 21st International Conference on Data Engineering (ICDE)*. (2005) 767–778
15. Venkataraman, S., Song, D.X., Gibbons, P.B., Blum, A.: New streaming algorithms for fast detection of superspreaders. In: *Proceedings of the Network and Distributed System Security Symposium (NDSS)*. (2005)
16. Keralapura, R., Cormode, G., Ramamirtham, J.: Communication-efficient distributed monitoring of thresholded counts. In: *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*. (2006) 289–300
17. Motwani, R., Raghavan, P.: *Randomized Algorithms*. Cambridge University Press (1995)
18. Broder, A.Z., Charikar, M., Frieze, A.M., Mitzenmacher, M.: Min-wise independent permutations (extended abstract). In: *STOC*. (1998) 327–336
19. Frieze, A., Grimmett, G.: *The Shortest-Path Problem for Graphs with Random Arc-lengths*. Volume 10. Elsevier Science Publishers B.V. (1985)
20. Alon, N., Matias, Y., Szegedy, M.: The space complexity of approximating the frequency moments. *Journal of Computer and System Sciences* **58**(1) (1999) 137–147
21. Gibbons, P., Tirthapura, S.: Estimating simple functions on the union of data streams. In: *Proc. ACM Symp. on Parallel Algorithms and Architectures (SPAA)*. (2001) 281–291
22. Bar-Yossef, Z., Jayram, T., Kumar, R., Sivakumar, D., Trevisan, L.: Counting distinct elements in a data stream. In: *Proc. 6th International Workshop on Randomization and Approximation Techniques (RANDOM)*. (2002) 1–10
23. Bar-Yossef, Z., Jayram, T., Kumar, R., Sivakumar, D., Trevisan, L.: Counting distinct elements in a data stream. In: *Proceedings of the 6th International Workshop on Randomization and Approximation Techniques (RANDOM)*. (2002) 1–10

A Proofs from Section 3

Proof of Lemma 1 (False Negative).

Statement: If v is an element with $f_v \geq k$, then with probability at least $1 - \delta$, v is returned as a frequent element by every node after $20N \ln N$ rounds.

Proof. A false negative can occur in one of two ways (1) Less than $r = \frac{12k^2}{\lambda^2} (1 - \frac{\lambda}{2k}) \ln \frac{2}{\delta}$ copies of v were sampled in the initialization phase (step 1(b)) of the algorithm described in Figure 1, or (2) r or more copies of v were sampled during initialization, but some copies did not make it to all nodes during the gossip. Let E_1 denote the event that less than r copies of v were sampled. Let E_2 denote the event that after $20N \ln N$ transmissions, there was a sampled object (with r or more sampled copies) that was not disseminated to all nodes using gossip. Let \mathcal{E} denote the event that a false negative occurred, then $\Pr[\mathcal{E}] = \Pr[E_1 \cup E_2] \leq \Pr[E_1] + \Pr[E_2]$.

Consider some k copies of v in the input. Let X_v be a random variable that denotes the total number of these k copies of v that were sampled. X_v is a binomial random variable with f_v trials and the probability of success in each trial being $\frac{12k}{\lambda^2} \ln \frac{2}{\delta}$ (this is the probability with which a node samples its own data item). It follows that $E[X_v] = \frac{12k^2}{\lambda^2} \ln \frac{2}{\delta}$. Using Chernoff bounds, we get:

$$\begin{aligned} \Pr[E_1] &= \Pr[X_v < r] \\ &= \Pr[X_v < \frac{12k^2}{\lambda^2} (1 - \frac{\lambda}{2k}) \ln \frac{2}{\delta}] \\ &= \Pr[X_v < E[X_v] - \frac{6k}{\lambda} \ln \frac{2}{\delta}] \\ &\leq e^{-\frac{3}{2} \ln \frac{2}{\delta}} \text{ [using Chernoff bounds]} \\ &= (\frac{\delta}{2})^{\frac{3}{2}} < \frac{\delta}{2} \end{aligned}$$

In Lemma 5 we show that after $20N \ln N$ transmissions, the probability that all nodes have all sampled copies is at least $1 - 1/N$. Thus, $\Pr[E_2] < \frac{1}{N}$.

Let \mathcal{E} denote the event that a false negative has occurred. Then

$$\Pr[\mathcal{E}] = \Pr[E_1 \cup E_2] \leq \Pr[E_1] + \Pr[E_2] < \frac{\delta}{2} + \frac{1}{N} < \delta$$

Proof of Lemma 2 (False Positive).

Statement: If u is an element with $f_u \leq k - \lambda$, where $k^{\frac{3}{4}} \leq \lambda < k$, then the probability that u is returned by some node as a frequent element is no more than δ .

Proof. A false positive can occur if both these events happen (1) r or more copies of u are sampled initially (2) all r copies reach some node in the network through

gossip. Let these two events be E_1 and E_2 respectively, and let \mathcal{F} denote the event that a false positive occurred. Note that $\Pr[\mathcal{F}] = \Pr[E_1 \cap E_2] \leq \Pr[E_1]$.

Let X_u denote the number of copies of u that were sampled. Consider the “best case” scenario for a false positive, when $f_u = k - \lambda$. Then X_u is a binomial random variable with $E[X_u] = (k - \lambda) \frac{12k}{\lambda^2} \ln \frac{2}{\delta} = \frac{12k^2}{\lambda^2} (1 - \frac{\lambda}{k}) \ln \frac{2}{\delta}$. Note that

$$\begin{aligned} \Pr[E_1] &= \Pr[X_u > r] = \Pr[X_u > \frac{12k^2}{\lambda^2} (1 - \frac{\lambda}{2k}) \ln \frac{2}{\delta}] \\ &= \Pr[X_u > E[X_u] + \frac{6k}{\lambda} \ln \frac{2}{\delta}] \\ &\leq \exp(-(\ln \frac{2}{\delta}) (\frac{1}{1 - \frac{\lambda}{k}})) \text{ [using Chernoff bounds]} \\ &= (\frac{\delta}{2})^{\frac{1}{1 - \frac{\lambda}{k}}} \\ &< \frac{\delta}{2} \text{ [since } \frac{1}{1 - \frac{\lambda}{k}} > 1] \end{aligned}$$

Since $\Pr[\mathcal{F}] \leq \Pr[E_1]$, the lemma follows.

Proof of Lemma 3.

Statement: $E[T_\theta] = 2N \ln N + O(N)$, where T_θ is defined as the number of rounds till θ is disseminated to all nodes in the network.

Proof. Let ξ_t be the set of nodes that have θ after t transmissions. Thus ξ_0 has only one node (the one that sampled θ during the initialization step). For $t = 1 \dots N - 1$, let random variable X_t be the number of transmissions required to increase $|\xi|$ from t to $t + 1$. We can write:

$$T_\theta = X_1 + X_2 + \dots + X_{N-1}$$

For $t \geq 1$, in time step t , the size of ξ increases only if a gossip message is transmitted from node i to node j where $i \in \xi_{t-1}$ and $j \notin \xi_{t-1}$. We note that X_t is a geometric random variable, i.e. the number of trials till the first “success” where a success is defined as a message from node $i \in \xi_{t-1}$ to a node $j \notin \xi_{t-1}$. The probability of a success for X_t is $(\frac{t}{N})(1 - \frac{t}{N}) = \frac{t(N-t)}{N^2}$. We know that if a random variable Y follows a geometric distribution with probability of success p then $E[Y] = \frac{1}{p}$, so we get $E[X_t] = \frac{N^2}{t(N-t)}$. Using linearity of expectation, we have:

$$\begin{aligned} E[T_\theta] &= \sum_{t=1}^{N-1} E[X_t] = \sum_{t=1}^{N-1} \frac{N^2}{t(N-t)} \\ &= N \sum_{t=1}^{N-1} (\frac{1}{t} + \frac{1}{N-t}) = 2N \sum_{t=1}^{N-1} \frac{1}{t} = 2NH_{N-1} = 2N \ln N + O(N) \end{aligned}$$

where H_k denotes the k th Harmonic number.

Proof of Lemma 4

Statement: $\lim_{N \rightarrow \infty} \Pr(T_\theta > 20N \ln N) = O(\frac{1}{N^2})$

Proof. The dissemination of the object θ can be divided into two distinct phases. The first phase continues until the object has reached $\frac{N}{2}$ distinct nodes. The second phase starts once it has reached $\frac{N}{2}$ nodes and continues until it reaches N nodes. Note that, in the first phase, it is more unlikely to find a source node that has θ and it is very likely to find a destination that does not have θ . Once θ has reached $\frac{N}{2}$ nodes, the situation reverses.

We define the random variable X_j as the number of transmissions taken by $|\xi_\theta|$ to increase from j to $j+1$. We define T_θ^1 and T_θ^2 as the random variables that stand for the number of transmissions required in the two aforementioned phases respectively. That makes

$$T_\theta^1 = X_1 + X_2 + \dots + X_{\frac{N}{2}} \quad (1)$$

$$T_\theta^2 = X_{\frac{N}{2}+1} + X_{\frac{N}{2}+2} + \dots + X_{N-1} \quad (2)$$

Clearly,

$$T_\theta = T_\theta^1 + T_\theta^2$$

We define a “success” as increment of $|\xi_\theta|$ from j to $j+1$. So X_j follows a geometric distribution with probability of success $= \frac{j}{N}(1 - \frac{j}{N}) = \frac{j(N-j)}{N^2}$. Note that for $j \leq \frac{N}{2}$, the probability of success of X_j is $\frac{j(N-j)}{N^2} \geq \frac{j(\frac{N}{2})}{N^2} = \frac{j}{2N}$. Let Y_j be a geometric random variable with probability of success $= \frac{j}{2N}$ for $j \leq \frac{N}{2}$. The probability of success of Y_j can be at most that of X_j , hence, had the gossip taken place with the probability of success of Y_j , T' - the number of transmissions required to disseminate an object θ to $\frac{N}{2}$ nodes - would have been at least as big as T_θ^1 . So, given a point in the range of all possible convergence times ($10N \ln N$ in this case), T' is more (or equally) likely to exceed it than T_θ^1 . In addition, let T'' be the number of transmissions to be made when there are $2N$ nodes and the data item has to reach each node at least once. Since T'' has a higher expected value than T' , it is even more likely than T' to be more than a given point ($10N \ln N$ here, once again) in the range of all possible convergence times. Formally,

$$\Pr(T_\theta^1 > 10N \ln(2N)) \leq \Pr(T' > 10N \ln N) \leq \Pr(T'' > 10N \ln N) \quad (3)$$

Note that

$$\Pr[T'' > 10N \ln N] \leq \Pr[T'' > 2N \ln(2N) + 2N \ln(2N)^2] \quad (4)$$

since $10N \ln N > 2N \ln(2N) + 2N \ln(2N)^2$.

In the limit of large N , Theorem 1 yields $\Pr(T'' > 2N \ln(2N) + c(2N)) = 1 - e^{-e^{-c}}$. Using $c = \ln(2N^2)$, we get

$$\Pr(T'' > 2N \ln(2N) + 2N \ln(2N^2)) = 1 - e^{-\frac{1}{2N^2}} \quad (5)$$

Combining inequalities 3, 4 and equation 5 and taking the complement of the event in inequality 3, we get

$$\Pr[T_\theta^1 \leq 10N \ln N] \geq e^{-\frac{1}{2N^2}} \quad (6)$$

In a similar way, we can prove

$$\Pr[T_\theta^2 \leq 10N \ln N] \geq e^{-\frac{1}{2N^2}} \quad (7)$$

Note that if each of T_θ^1 and T_θ^2 is at most $10N \ln N$ with high likelihood, then that is a *sufficient* condition to ensure that T_θ is at most $20N \ln N$ with high probability (it is not a *necessary* condition though). So using inequalities 6 and 7, we can write:

$$\begin{aligned} \Pr(T_\theta \leq 20N \ln N) &\geq \Pr((T_\theta^1 \leq 10N \ln N) \text{ and } (T_\theta^2 \leq 10N \ln N)) \\ &\geq (e^{-\frac{1}{2N^2}})(e^{-\frac{1}{2N^2}}) \\ &= e^{-\frac{1}{N^2}} \end{aligned} \quad (8)$$

Using the inequality $e^{-x} \geq 1 - x$, inequality 8 implies

$$\Pr(T_\theta \leq 20N \ln N) \geq 1 - \frac{1}{N^2} \quad (9)$$

Taking the complement of the event in inequality 9 establishes our claim.

Proof of Lemma 6.

Statement: $E[\mathcal{B}] = O(N \ln N)$, where \mathcal{B} is the number of bytes taken to disseminate θ among all nodes.

Proof. Let X_j denote the total number of transmissions (whether involving θ or not) taken for $|\xi|$ to increase from j to $j+1$. Let B_j denote the number of these X_j transmissions that contain θ . We have:

$$\mathcal{B} = \sum_{i=1}^{N-1} B_i$$

In each of the B_j transmissions except for the last, a message is sent from a node that has θ to another node that has θ . Only in the last of the B_j transmissions, the message is sent from a node that has θ to a node that does not have θ . Thus B_j is a geometric random variable with a probability of success being the *conditional probability* that the destination of the gossip is a node which does not yet have θ , given that the source of the gossip has θ . This probability is $\frac{N-j}{N}$, since the number of nodes which do not yet have θ is $N-j$. Thus we have $E[B_j] = \frac{N}{N-j}$. Using linearity of expectation, we get:

$$\begin{aligned}
E[\mathcal{B}] &= \sum_{j=1}^{N-1} E[B_j] = \sum_{j=1}^{N-1} \frac{N}{N-j} \\
&= N \sum_{j=1}^{N-1} \frac{1}{j} = NH_{N-1} = N \ln N + O(N)
\end{aligned}$$

Proof of Lemma 7.

Statement: $\Pr[\mathcal{B} > 3N \ln N] = O(\frac{1}{N^2})$.

Proof. Using Theorem 1, and the characterization of \mathcal{B} from the Lemma 6, we get for any $c \in \mathcal{R}$,

$$\Pr[\mathcal{B} > N \ln N + cN] = 1 - e^{-e^{-c}}$$

Using $c = 2 \ln N$, we get

$$\Pr[\mathcal{B} > N \ln N + 2N \ln N] = 1 - e^{-\frac{1}{N^2}} \leq 1 - (1 - \frac{1}{N^2}) = \frac{1}{N^2}$$

where we have used the inequality $e^x \geq (1+x)$ for all real x .

Proof of Theorem 3 (Communication Complexity for Absolute Threshold).

Statement: With high probability, $\mathcal{Y} = O(\frac{N^2 k}{\lambda^2} \ln \frac{1}{\delta} \ln N)$, where \mathcal{Y} is the total number of bytes that need to be exchanged for the whole protocol until the frequent elements have been identified.

Proof. From Lemma 7, we can infer

$$\Pr[\mathcal{B} \leq 3N \ln N] \geq 1 - \frac{1}{N^2} \tag{10}$$

In the initialization phase of the algorithm, each item is sampled with probability $\frac{12k}{\lambda^2} \ln(\frac{2}{\delta})$. We try to find an upper bound on the communication overhead for exchanging the sampled data items through gossip. Let Z be a random variable that stands for the number of data items that get sampled during the initialization. Z follows a binomial distribution with number of trials N and probability of success in each trial $\frac{12k}{\lambda^2} \ln(\frac{2}{\delta})$. Hence the expected number of data items that get sampled is $E[Z] = \frac{12Nk}{\lambda^2} \ln(\frac{2}{\delta})$. We try to find high-probability bounds for Z . Applying Chernoff bounds,

$$\begin{aligned}
\Pr[Z > \frac{18Nk}{\lambda^2} \ln(\frac{2}{\delta})] &= \Pr[Z > (1 + \frac{1}{2}) \frac{12Nk}{\lambda^2} \ln(\frac{2}{\delta})] \\
&\leq e^{-\frac{Nk}{\lambda^2} \ln(\frac{2}{\delta})} \\
&= (\frac{\delta}{2})^{\frac{Nk}{\lambda^2}} \\
&< \frac{\delta}{2} \text{ [since } N > \lambda \text{ and } k > \lambda]
\end{aligned}$$

This implies

$$\Pr[Z \leq \frac{18Nk}{\lambda^2} \ln(\frac{2}{\delta})] \geq 1 - \frac{\delta}{2} \quad (11)$$

Note that since \mathcal{Y} is the number of bytes to be exchanged for the whole protocol until *all* the frequent elements have been identified, if Z is bound within $\frac{18Nk}{\lambda^2} \ln(\frac{2}{\delta})$ and \mathcal{B} within $3N \ln N$, that can be *one* way to limit \mathcal{Y} within $\frac{54N^2k}{\lambda^2} \ln(\frac{2}{\delta}) \ln N$. To be precise, combining inequalities 10 and 11, we get

$$\begin{aligned} \Pr[\mathcal{Y} \leq \frac{54N^2k}{\lambda^2} \ln(\frac{2}{\delta}) \ln N] &\geq \Pr[Z \leq \frac{18Nk}{\lambda^2} \ln(\frac{2}{\delta})] \Pr[\mathcal{B} \leq 3N \ln N] \\ &\geq (1 - \frac{\delta}{2})(1 - \frac{1}{N^2}) \\ &= 1 - [\frac{\delta}{2} + \frac{1}{N^2} - (\frac{1}{N^2})(\frac{\delta}{2})] \geq 1 - \delta \end{aligned}$$

B Proofs from Section 4

Proof of Lemma 8.

Statement: For $t = \frac{128}{\psi^2} \ln(\frac{3}{\delta})$, τ , the t^{th} minimum element among N randomly generated values in $(0, 1)$, satisfies the following properties:

1. $\Pr[\tau < \frac{t}{N}(1 - \frac{\psi}{4})] < \frac{\delta}{3}$
2. $\Pr[\tau > \frac{t}{N}(1 + \frac{\psi}{4})] < \frac{\delta}{3}$

Proof. Let X be a random variable indicating the number of elements with weight less than $\frac{t}{N}(1 - \frac{\psi}{4})$. X follows a binomial distribution with number of trials = N and probability of success = $\frac{t}{N}(1 - \frac{\psi}{4})$. This gives $E[X] = t(1 - \frac{\psi}{4})$, and

$$\begin{aligned} \Pr[\tau < \frac{t}{N}(1 - \frac{\psi}{4})] &= \Pr[X \geq t] = \Pr[X \geq E[X](\frac{1}{1 - \frac{\psi}{4}})] \\ &\leq \Pr[X \geq E[X](1 + \frac{\psi}{4})] \quad [\text{since } \frac{1}{1 - \frac{\psi}{4}} > 1 + \frac{\psi}{4}] \\ &\leq e^{-\frac{E[X]\psi^2}{48}} \quad [\text{by Chernoff bound}] \\ &= e^{-\frac{t(1 - \frac{\psi}{4})\psi^2}{48}} \end{aligned} \quad (12)$$

Note that with $t = \frac{128}{\psi^2} \ln(\frac{3}{\delta})$,

$$\frac{t(1 - \frac{\psi}{4})\psi^2}{48} = \frac{8}{3} \ln(\frac{3}{\delta})(1 - \frac{\psi}{4}) \geq \ln(\frac{3}{\delta}) \quad (13)$$

Note that $\frac{8}{3}(1 - \frac{\psi}{4}) \geq 1$ since $\psi \leq 1$. Substituting the result of inequality 13 in inequality 12 yields

$$\Pr[\tau < \frac{t}{N}(1 - \frac{\psi}{4})] \leq e^{-\ln(\frac{3}{8})} = \frac{\delta}{3}$$

which completes the proof of the first claim.

For the second part, let Y be a random variable that indicates the number of elements with weight less than $\frac{t}{N}(1 + \frac{\psi}{4})$. Y follows a binomial distribution with number of trials = N and probability of success = $\frac{t}{N}(1 + \frac{\psi}{4})$. This gives $E[Y] = t(1 + \frac{\psi}{4})$, and

$$\Pr[\tau > \frac{t}{N}(1 + \frac{\psi}{4})] = \Pr[Y < t] = \Pr[Y < E[Y](\frac{1}{1 + \frac{\psi}{4}})] \quad (14)$$

Now, $\frac{1}{1 + \frac{\psi}{4}} \leq 1 - \frac{\psi}{8}$ since $(1 + \frac{\psi}{4})(1 - \frac{\psi}{8}) = 1 + \frac{\psi}{8} - \frac{\psi^2}{32} \geq 1$. This yields

$$\begin{aligned} \Pr[Y < E[Y](\frac{1}{1 + \frac{\psi}{4}})] &\leq \Pr[Y < E[Y](1 - \frac{\psi}{8})] \\ &\leq e^{-\frac{E[Y]\psi^2}{64}(\frac{1}{2})} \text{ [by Chernoff bound]} \end{aligned} \quad (15)$$

Now, with $t = \frac{128}{\psi^2} \ln(\frac{3}{8})$,

$$\frac{E[Y]\psi^2}{64} = t(1 + \frac{\psi}{4})\frac{\psi^2}{64} = \frac{128}{64} \ln(\frac{3}{8})(1 + \psi) \geq 2 \ln(\frac{3}{8}) \quad (16)$$

Substituting the results of inequality 16 in inequality 15, and then inequality 15 in inequality 14 yields

$$\Pr[\tau > \frac{t}{N}(1 + \frac{\psi}{4})] \leq e^{-\ln(\frac{3}{8})} = \frac{\delta}{3}$$

Proof of Lemma 10 (False Negative).

Statement: If a data item v is a frequent item, i.e. $f_v \geq \phi N$, then with probability at least $1 - \delta$, v is identified by every node as a frequent element.

Proof. Let Z be a random variable indicating the number of copies of v with weight $\frac{t}{N}(1 - \frac{\psi}{4})$ or less. Note that there are two sources of randomness in the entire process, and therefore identification of v as a frequent element on the basis of sketch requires two criteria to be satisfied. First, the t^{th} minimum weight τ should exceed $\frac{t}{N}(1 - \frac{\psi}{4})$, and then, at least $\phi - \frac{\psi}{2}$ occurrences of v should produce a weight which is less than or equal to $\frac{t}{N}(1 - \frac{\psi}{4})$, to ensure that make it within the t minimum hashed weights.

Formally,

$$\Pr[v \text{ is identified as a frequent element}]$$

$$\begin{aligned}
&\geq \Pr[(\tau \geq \frac{t}{N}(1 - \frac{\psi}{4})) \text{ and } (Z \geq (\phi - \frac{\psi}{2})t)] \\
&= 1 - \Pr[(\tau < \frac{t}{N}(1 - \frac{\psi}{4})) \text{ or } (Z < (\phi - \frac{\psi}{2})t)] \\
&\geq 1 - \Pr[(\tau < \frac{t}{N}(1 - \frac{\psi}{4}))] - \Pr[(Z < (\phi - \frac{\psi}{2})t)] \text{ [by union bound]} \\
&\geq 1 - \frac{\delta}{3} - \Pr[Z < (\phi - \frac{\psi}{2})t] \tag{17}
\end{aligned}$$

To estimate $\Pr[Z < (\phi - \frac{\psi}{2})t]$, note that Z follows a binomial distribution with ϕN or more trials and a probability of success of $\frac{t}{N}(1 - \frac{\psi}{4})$. This makes

$$E[Z] \geq (\phi N)(\frac{t}{N}(1 - \frac{\psi}{4})) = \phi t(1 - \frac{\psi}{4}) \geq (\phi - \frac{\psi}{4})t$$

We take another random variable Z' that is a ‘‘pessimistic’’ version of Z in the sense that $E[Z'] = (\phi - \frac{\psi}{4})t$, which implies that Z' has a higher likelihood of being less than $(\phi - \frac{\psi}{2})t$. So we have

$$\Pr[Z < (\phi - \frac{\psi}{2})t] \leq \Pr[Z' < (\phi - \frac{\psi}{2})t] = \Pr[Z' < E[Z'](1 - \frac{\psi}{4\phi - \psi})] \leq e^{-E[Z'] \frac{\psi^2}{16(\phi - \frac{\psi}{4})^2}} \tag{18}$$

Substituting $t = \frac{128}{\psi^2} \ln(\frac{3}{\delta})$ in $E[Z']$,

$$E[Z'] \frac{\psi^2}{16(\phi - \frac{\psi}{4})^2} = (\phi - \frac{\psi}{4}) \frac{128}{\psi^2} \ln(\frac{3}{\delta}) \psi^2 \frac{1}{16(\phi - \frac{\psi}{4})^2} = \frac{8 \ln(\frac{3}{\delta})}{\phi - \frac{\psi}{4}} \geq 8 \ln(\frac{3}{\delta}) \tag{19}$$

Substituting inequality 19 in inequality 18 yields

$$\Pr[Z < (\phi - \frac{\psi}{2})t] \leq e^{-8 \ln(\frac{3}{\delta})} = (\frac{\delta}{3})^8 \tag{20}$$

Substituting the result of inequality 20 in inequality 17, we get

$$\Pr[v \text{ is identified as a frequent element}] \geq 1 - \frac{\delta}{3} - (\frac{\delta}{3})^8 \geq 1 - \delta$$

Proof of Lemma 11 (False Positive).

Statement: With probability at least $1 - \delta$, a data item u with a relative frequency of less than $(\phi - \psi)$ is not identified by any node as a frequent element.

Proof. Let W denote the number of copies of u with weight $\leq \frac{t}{N}(1 + \frac{\psi}{4})$. Note that we are conservative about the value of the t^{th} minimum weight here. We are using the upper bound $\frac{t}{N}(1 + \frac{\psi}{4})$ of the range in which we expect the t^{th} minimum weight to belong to. As in the proof of Lemma 10, we can write

$$\Pr[u \text{ is not identified as a frequent element}]$$

$$\begin{aligned}
&\geq \Pr[(\tau \leq \frac{t}{N}(1 + \frac{\psi}{4})) \text{ and } (W \leq (\phi - \frac{\psi}{2})t)] \\
&= 1 - \Pr[(\tau > \frac{t}{N}(1 + \frac{\psi}{4})) \text{ or } (W > (\phi - \frac{\psi}{2})t)] \\
&\geq 1 - \Pr[\tau > \frac{t}{N}(1 + \frac{\psi}{4})] - \Pr[W > (\phi - \frac{\psi}{2})t] \text{ [by union bound]} \\
&\geq 1 - \frac{\delta}{3} - \Pr[W > (\phi - \frac{\psi}{2})t] \tag{21}
\end{aligned}$$

To estimate $\Pr[W > (\phi - \frac{\psi}{2})t]$, note that W follows a binomial distribution with $(\phi - \psi)N$ or less trials and a probability of success of $\frac{t}{N}(1 + \frac{\psi}{4})$. This makes

$$E[W] \leq (\phi - \psi)N(\frac{t}{N}(1 + \frac{\psi}{4})) = t[\phi - \frac{3\psi}{4} - \frac{\psi^2}{4}] \leq t[\phi - \frac{3\psi}{4}]$$

We take another random variable W' that is a pessimistic version of W in the sense that $E[W'] = (\phi - \frac{3\psi}{4})t$, which implies that W' has a higher likelihood of being greater than $(\phi - \frac{\psi}{2})t$. So we have

$$\begin{aligned}
\Pr[W > (\phi - \frac{\psi}{2})t] &\leq \Pr[W' > (\phi - \frac{\psi}{2})t] \\
&= \Pr[W' > E[W'](1 + \psi')] \text{ where } \psi' = \frac{\psi}{4\phi - 3\psi} \\
&\leq e^{-\frac{E[W']\psi'^2}{3}} \text{ [by Chernoff bounds]} \tag{22}
\end{aligned}$$

Now,

$$E[W']\frac{\psi'^2}{3} = t(\frac{4\phi - 3\psi}{4})\frac{\psi^2}{(4\phi - 3\psi)^2}\frac{1}{3} = \frac{t\psi^2}{12(4\phi - 3\psi)} \geq \frac{t\psi^2}{(12)(4)} \tag{23}$$

Substituting $t = \frac{128}{\psi^2} \ln(\frac{3}{\delta})$ in inequality 23,

$$E[W']\frac{\psi'^2}{3} \geq \frac{8}{3} \ln(\frac{3}{\delta}) \geq \ln(\frac{3}{\delta}) \tag{24}$$

Substituting inequality 24 in inequality 22 yields

$$\Pr[W > (\phi - \frac{\psi}{2})t] \leq e^{-\ln(\frac{3}{\delta})} = \frac{\delta}{3} \tag{25}$$

Substituting inequality 25 in inequality 21, we get

$$\begin{aligned}
\Pr[u \text{ is not identified as a frequent element}] &\geq 1 - \frac{\delta}{3} - \frac{\delta}{3} \\
&\geq 1 - \delta
\end{aligned}$$

Input: Data item m_i , error probability δ , frequency threshold k , approximation error λ

1. **Initialization**

(a) Choose ρ to be a uniformly distributed random number in the real interval $(0, 1)$.

(b) If $\rho < \frac{12k}{\lambda^2} \ln \frac{2}{\delta}$ then $S_i \leftarrow \{(i, m_i)\}$, else $S_i \leftarrow \Phi$ /* null set */

2. **In each round of Gossip:**

(a) If sketch S_j is received from node j then $S_i \leftarrow S_i \cup S_j$

(b) Select node j uniformly at random from $1 \dots N$

(c) Send S_i to j

3. **Query:**

When queried for the frequent elements, report all data items which occur more than

$r = \frac{12k}{\lambda^2} (1 - \frac{\lambda}{2k}) \ln \frac{2}{\delta}$ times in S_i as frequent elements.

Fig. 4. Synchronous Gossip algorithm at node i for finding the frequently occurring elements with an absolute threshold k .

C Details from Section 5

C.1 Algorithms for Synchronous Gossip

Proof of Theorem 7

Statement: If the synchronous algorithms in Figures 4 and 5 are run for $(1 + 3 \ln 2) \log N$ rounds, then all frequent items (with absolute and relative thresholds, respectively) will be identified with probability at least $1 - \delta$, and no infrequent item will be identified, with probability at least $1 - \delta$.

Proof. The number of sampled items is trivially no more than N . If all these items are disseminated to all nodes, then the guarantees will be met. Substituting $m = N$ in Lemma 12 yields the desired result. We can get a slightly tighter result (but asymptotically still the same) by using a better bound on the number of sampled items.

Input: Data item m_i ; error probability δ , relative frequency threshold ϕ , approximation error ψ

1. **Initialization.** $t \leftarrow \frac{128}{\psi^2} \ln(\frac{3}{\delta})$; $w_i \leftarrow h(i)$; $S_i \leftarrow \{(m_i, w_i)\}$
2. **In each round of gossip:**
 - (a) If sketch S_j was received from node j then
 - i. $S_i \leftarrow S_i \cup S_j$
 - ii. If $|S_i| > t$ then retain t elements of S_i with smallest weights
 - (b) Select node j uniformly at random
 - (c) Send S_i to j
3. **Query:**

When queried for the frequent items, report every value v such that at least $(\phi - \frac{\psi}{2})t$ (value, weight) tuples exist in S_i with value = v

Fig. 5. Synchronous gossip algorithm at node i for finding the frequently occurring elements with a relative threshold