J. Parallel Distrib. Comput. 70 (2010) 1241–1253

Contents lists available at ScienceDirect

J. Parallel Distrib. Comput.

journal homepage: www.elsevier.com/locate/jpdc

Identifying frequent items in a network using gossip ☆

Bibudh Lahiri *, Srikanta Tirthapura

Department of Electrical and Computer Engineering, Iowa State University, Ames, IA 50011, United States

ARTICLE INFO

Article history:
Received 18 November 2009
Received in revised form
29 June 2010
Accepted 23 July 2010
Available online 8 August 2010

Keywords:
Gossip
Frequent items
Sketch

ABSTRACT

We present algorithms for identifying frequently occurring items in a large distributed data set. Our algorithms use gossip as the underlying communication mechanism, and do not rely on any central control, nor on an underlying network structure, such as a spanning tree. Instead, nodes repeatedly select a random partner and exchange data with that 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 identifying frequent items is built by layering a novel small space "sketch" of data over a gossip-based data dissemination mechanism. We prove that the algorithm identifies the frequent items with high probability, and provides bounds on the time till convergence. To our knowledge, this is the first work on identifying frequent items using gossip.

Published by Elsevier Inc.

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, and hence, there is no overhead of sub-network formation and maintenance. A gossip protocol proceeds in many "rounds". 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 in distributed systems was first proposed by Demers et al. [6].

We consider the problem of identifying *frequent items* in a distributed data set, 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 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 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 algorithm can be used to track the most frequently accessed items in a low-overhead, decentralized manner, without having to aggregate the frequencies of accesses at any central location. 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 items 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 versions of the problem, one with a relative threshold on the frequency, and the other with an absolute threshold on the frequency. In the relative threshold version, the task is to identify all items whose frequency of occurrence is more than a certain fraction of the total size of the data, where the fraction (the relative threshold) is a user-defined parameter. In the absolute threshold version, the task is to identify all items whose frequency of occurrence is at least an absolute number (the absolute threshold), which is a 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 items at any single place in the network. Instead,

☆ A preliminary version of this paper appeared in the *Proceedings of the 15th International Colloquium on Structural Information and Communication Complexity, 2008, pages 119–130.*

* Corresponding author.
E-mail addresses: bibudh@iastate.edu, bibudhlahiri@gmail.com (B. Lahiri), snt@iastate.edu (S. Tirthapura).
URLs: http://www.ece.iastate.edu/~bibudh/ (B. Lahiri), http://www.ece.iastate.edu/~snt/ (S. Tirthapura).

0743-7315/$ – see front matter. Published by Elsevier Inc.
doi:10.1016/j.jpdc.2010.07.006

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 on the sketch is that it 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 items whose frequency exceeds the user defined threshold. At the same time, items whose popularity is significantly below the threshold will be omitted (again, with high probability).

Contributions. The contributions of this work are as follows:

- We present randomized algorithms for identifying frequent items using gossip, for both the relative and absolute threshold versions of the problem.
- For each algorithm, we present a rigorous analysis of the correctness, time till convergence, and the communication overhead. Our analysis shows that these algorithms converge quickly, and can maintain frequent items in a network with a reasonable communication overhead.
- We present results from our simulations on synthetic data sets. We observed in our simulations that the convergence time and the communication overhead were both much lower than the theoretically guaranteed predictions.

To our knowledge, this is the first work on identifying frequent items in a distributed data set using gossip.

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 and is maintaining statistics about frequently requested data items, or the most frequently observed data in a distributed system. In such an application, an exact answer may not be required, and an approximate answer may suffice.

1.1. Related work

Demers et al. [6] were the first to provide a formal treatment of gossip protocols (or "epidemic algorithms" as they called them) for data dissemination. Kempe and Kleinberg [14] 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 et al. [13] proposed algorithms for computing the sum, average, approximately uniform random sample and quantiles using uniform gossip. Their algorithm for quantiles is based on their algorithm for the sum—they choose a random element in the data, and count the number of elements that are greater and less than the chosen element, and recurse on smaller data sets until the quantile is found. Thus their algorithms need many instances of "sum" computations to converge before the quantile is found. A similar approach could potentially be used to find frequent items using gossip. In contrast, our algorithms are not based on repeated computation of the sum, and thus converge faster.

Much recent work [1,2,20] 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 items 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. [12] proposed algorithms for gossip with flexible tradeoffs between the number of rounds and the number of messages transmitted. Dimakis et al. [7] consider the problem of computing the average over *random geometric graphs* with location-aware nodes, combining uniform gossip with greedy geographic routing. Deb et al. [5] used network coding along with uniform gossip to speed up the dissemination of k messages in the network.

The problem of identifying frequent items in data has been extensively studied [18,17,11] in the database, data streams and network monitoring communities (where frequent items are often called "heavy-hitters"). The early work in this is due to Misra and Gries [18], who proposed a deterministic algorithm to identify frequent items in a stream in small space, followed by Manku and Motwani [17], who gave randomized and deterministic algorithms for the same problem. The above were algorithms for a centralized setting.

In a distributed setting, Cao and Wang [4] proposed an algorithm to find the top-k items, where they first made a lower-bound estimate for the kth value, and then used the estimate as a threshold to prune away items which should not qualify as top-k. Zhao et al. [22] proposed a sampling-based and a counting-sketch-based scheme to identify globally frequent items. Manjhi et al. [16] present an algorithm for finding frequent items on distributed streams through a tree-based aggregation. Venkataraman et al. [21] present an algorithm for identifying "superspreaders" or "heavy distinct hitters" in a network data stream. Keralapura et al. [15] proposed an algorithm for continuously maintaining the frequent items over a network of nodes. The above algorithms sometimes assume the presence of a central node, or an underlying network structure such as a spanning tree [16,15], and hence are not applicable where the underlying network does not guarantee reliability or robustness. Haridasan and van Renesse [10] proposed a gossip-based technique that allowed each node in a network to estimate the distribution of values held by other nodes, but their results did not offer any theoretical bounds.

Organization of the paper. In Section 2, we state our system model and give a precise definition of the problem. We first present the algorithm and analysis for the relative threshold version in Section 3, and then the absolute threshold version in Section 4. In Section 5 we discuss the simulation results for both absolute and relative thresholds with an asynchronous time model. In Section 6, we discuss the extension of these results to synchronous gossip.

2. Model

We consider a distributed system with N nodes numbered from 1 to N. The number of nodes N is not necessarily known to any participating node, and this information is not used by the algorithms. An "item" is an integer from the set $\{1, 2, \ldots, m\}$. An "element" is a single occurrence of an item at any node. Each node i holds a multiset of items M_i, or equivalently, a set of elements. Let $\mathcal{N}_i$ denote the size of M_i. Let $M = \bigcup_{i=1}^{N} M_i$ denote the set of all elements in the network.

For item $v \in [m]$, the frequency of v is denoted by f_v, and is defined as the number of occurrences of v in M. 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 items v such that f_v is large. Let the total number of elements be defined as $\mathcal{N} = \sum_{i=1}^{N} \mathcal{N}_i$.

We consider the scenario of *uniform gossip*, which is the most commonly used model of gossip. Whenever a node i is chosen to transmit, it chooses the destination of its message to be a node selected uniformly at random 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 assume that the participating nodes execute the algorithms faithfully, and do not maliciously attempt to influence the results of the computation by sending spurious/incorrect messages.

Problem definition. We consider two variants of the problem, depending on how the thresholds are defined.

- *Relative threshold*. The user may be interested in identifying items whose relative frequency in the data set 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 $f_v \geq \phi\mathcal{N}$, and v is considered an infrequent item if $f_v < (\phi - \psi)\mathcal{N}$. According to this definition, there may be no more than $1/\phi$ frequent items.
- *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 an infrequent item if $f_v < k - \lambda$. Note that there may be up to $\mathcal{N}/k$ frequent items according to this definition.

In a centralized setting, when all elements are being observed at the same location, the above formulations of relative and absolute thresholds are equivalent, since the number of elements $\mathcal{N}$ can be computed easily, 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 elements in the system accurately enough. Thus, we treat these two problems separately. The lack of knowledge of the network size N does not, though, prevent the system from choosing gossip partners uniformly at random. For example, Gkantsidis et al. [9] show how random walks can provide a good approximation to uniform sampling for networks where the gap between the first and the second eigenvalues of the transition matrix is constant.

Once the gossip has continued for long enough, the following probabilistic guarantees must hold, whether for absolute or relative thresholds. Let δ be a user-provided bound on the error probability ($0 < \delta < 1$).

- With probability at least $(1 - \delta)$, every node reports every frequent item.
- With probability at least $(1 - \delta)$, no node reports an infrequent item.

Note that we present randomized algorithms, where the probabilistic guarantees hold irrespective of the input.

Time model. Time is divided into non-overlapping rounds. We consider two types of models, asynchronous and synchronous, depending on whether or not the nodes proceed at the same rate.

- In the *asynchronous* model, in each round, a single source node, chosen uniformly at random out of all N nodes, transmits to another randomly chosen receiver. Thus, in each round in the asynchronous model, there is only one message.
- In the *synchronous* model of communication, in each round, every node in the network 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.

In Sections 3 and 4, we mostly focus on the asynchronous model. We discuss the extension of our results to the synchronous model in Section 6.

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 items with relative threshold

Given thresholds ϕ and ψ, where $\psi < \phi$, the goal is to identify all items v such that $f_v \geq \phi\mathcal{N}$ and no item u such that $f_u < (\phi - \psi)\mathcal{N}$.

We first describe the intuition. Our algorithm is based on random sampling. The idea is that if an item occurs frequently in the original data, it is likely to occur at approximately the same relative frequency in an appropriately sized random sample as well. Hence, if we choose those items which have occurred frequently in the random sample, we are also likely to choose the frequent items in the input. To give guaranteed accuracy, we need a large enough sampling probability. However, this sampling probability cannot be decided in advance since the size of the dataset is not known beforehand. Hence, our algorithm works with an adaptive sampling probability, based on the idea of *min-wise independent permutations* [3]. For $i \in [N]$ and $\ell \in [\mathcal{N}_i]$, let the tuple (i, ℓ) denote the ℓth element within M_i. Thus the tuple (i, ℓ) uniquely identifies an element within M, by first identifying a node id i, and then an element within M_i. Let m_i^ℓ denote the value of element (i, ℓ). The algorithm assigns each element (i, ℓ) a weight w_i^ℓ, which is a random number in the unit interval (0, 1).

The algorithm maintains a sketch S of $(i, \ell, m_i^\ell, w_i^\ell)$ tuples, where (i, ℓ) identifies the element, m_i^ℓ is the value of the element, and w_i^ℓ is the weight. The sketch has no more than t elements, and only the tuples that have the smallest weights are included in S. The intuition is that if an item v has a large relative frequency, then v must occur frequently among the tuples with the t smallest weights, and hence in the sketch. Maintaining these t elements with the smallest weights through gossip is easy, just as it is easy to maintain the smallest weight element through gossip. If we choose a large enough sketch size t, the likelihood of a frequent item appearing in the sketch a sufficient number of times is very high.

The algorithm for the asynchronous model is described in Fig. 1. The threshold t is determined through the analysis to be $O\left(\frac{1}{\psi^2}\ln\left(\frac{1}{\delta}\right)\right)$. There are three parts to this algorithm (and all others that we describe). The first part is the Initialization, where each node initializes its own sketch. The next part is the Gossip, where the nodes exchange sketches with each other according to the communication model. 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 a query for frequent items is answered using the sketch. The accuracy of the result improves as further rounds of gossip occur.

The sketch at any node can be stored by any data structure that implements an associative array with keys in a sorted order, the key here being w_i^ℓ, and the value being a combination of the other three. In our simulation, we created a Java object of class `Tuple` with $(i, \ell, m_i^\ell, w_i^\ell)$, and each node maintains the sketch as a `TreeSet` collection of these `Tuple` objects. The `TreeSet` Java class allows the objects to be stored in a user-defined order, so we kept the objects sorted as per the weight w_i^ℓ. The sorting order was defined by a class called `WeightComparator`, which implements the `Comparator` interface of Java. When two sketches (`TreeSet` objects) were merged in our simulation, the `TreeSet` class ensured that when an object is added to the sketch of the local node, the user-defined ordering on weights was preserved. The implementation of the `TreeSet` class ensures that the time taken to add, remove or check the existence of an item to a `TreeSet` collection is logarithmic in the number of objects in the collection. After merging two sketches, we checked whether the total number of objects exceeded the maximum size of the sketch (t), and if it did, we created a `SortedSet` object with the t objects with the lowest weights, and re-initialized the `TreeSet` object (the sketch) with this `SortedSet` object.

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

1. **Initialization:**
   (a) $t \leftarrow \frac{128}{\psi^2}\ln(\frac{3}{\delta})$; $S_i \leftarrow \Phi$
   (b) for $\ell = 1$ to $\mathcal{N}_i$
      i. Choose w_i^ℓ as a uniformly distributed random number in $(0, 1)$
      ii. Set $S_i \leftarrow S_i \cup \{(i, \ell, m_i^\ell, w_i^\ell)\}$
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 the 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 items, report every value v such that at least $(\phi - \frac{\psi}{2})t$ (nodeID, elementID, value, weight) tuples exist in S_i with value equal to v

Fig. 1. Gossip algorithm at node *i* for finding the frequently occurring items with a relative threshold.

3.1. Analysis

Let $\mathcal{W}$ denote the multi-set of weights $\bigcup_{i=1}^{N}\bigcup_{\ell=1}^{\mathcal{N}_i}\{w_i^\ell\}$. Clearly, $\mathcal{N} = |\mathcal{W}|$. Let τ denote the tth smallest element in $\mathcal{W}$. Let M^t be the set of elements $\{(i, \ell)\}$ such that the $w_i^\ell \le \tau$, where ties are broken arbitrarily. In other words, M^t is the set of t input elements which have been assigned the smallest weights.

The analysis can be divided into two parts. We first show that, with high probability, each frequent item occurs with a sufficient frequency in M^t. Similarly, with high probability, the frequency in M^t of each infrequent item is small. As a result, if the sketch at a node equals M^t, then it can identify frequent items with a low probability of a false positive or a false negative. Note that this portion of the analysis is purely local, and has not yet dealt with the distributed algorithm directly.

Next, we analyze the distributed gossip process and prove that, with high probability, the set M^t is disseminated to all nodes within $O(N \log N)$ rounds. Combining the analysis of the gossip with the results about false positives and false negatives, we obtain the main result about the correctness of the algorithm, Theorem 3.1.

3.2. Analysis of M^t

We first show that τ is sharply concentrated around $\frac{t}{\mathcal{N}}$.

Lemma 3.1. *If* $t = \frac{128}{\psi^2}\ln\left(\frac{3}{\delta}\right)$, *then:* (1) $\Pr\left[\tau < \frac{t}{\mathcal{N}}\left(1 - \frac{\psi}{4}\right)\right] < \frac{\delta}{3}$ *and* (2) $\Pr\left[\tau > \frac{t}{\mathcal{N}}\left(1 + \frac{\psi}{4}\right)\right] < \frac{\delta}{3}$.

Proof. Let X be a random variable equal to the number of elements in $\mathcal{W}$ that are less than $\frac{t}{\mathcal{N}}\left(1 - \frac{\psi}{4}\right)$. Since the weights are chosen independently of each other, X follows a binomial distribution with $\mathcal{N}$ trials and probability of success in each trial $\frac{t}{\mathcal{N}}\left(1 - \frac{\psi}{4}\right)$. This gives $E[X] = t\left(1 - \frac{\psi}{4}\right)$. Using Chernoff bounds, we get

$$\Pr\left[\tau < \frac{t}{\mathcal{N}}\left(1 - \frac{\psi}{4}\right)\right]$$
$$= \Pr[X \ge t] = \Pr\left[X \ge E[X]\left(\frac{1}{1 - \frac{\psi}{4}}\right)\right]$$
$$\le \Pr\left[X \ge E[X]\left(1 + \frac{\psi}{4}\right)\right] \quad \left[\text{since } \frac{1}{1 - \frac{\psi}{4}} > 1 + \frac{\psi}{4}\right]$$
$$\le e^{-\frac{E[X]\psi^2}{48}} = e^{-\frac{t\left(1 - \frac{\psi}{4}\right)\psi^2}{48}}. \tag{1}$$

Using $t = \frac{128}{\psi^2}\ln\left(\frac{3}{\delta}\right)$,

$$\frac{t\left(1 - \frac{\psi}{4}\right)\psi^2}{48} = \frac{8}{3}\ln\left(\frac{3}{\delta}\right)\left(1 - \frac{\psi}{4}\right) \ge \ln\left(\frac{3}{\delta}\right). \tag{2}$$

Note that $\frac{8}{3}\left(1 - \frac{\psi}{4}\right) \ge 1$ since $\psi \le 1$. Substituting (2) in (1) yields:

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

which completes the proof of the first part.

For the second part, let Y be a random variable equal to the number of elements in $\mathcal{W}$ that are less than $\frac{t}{\mathcal{N}}\left(1 + \frac{\psi}{4}\right)$. Y follows a binomial distribution with $\mathcal{N}$ trials and probability of success in each trial equal to $\frac{t}{\mathcal{N}}\left(1 + \frac{\psi}{4}\right)$. This gives $E[Y] = t\left(1 + \frac{\psi}{4}\right)$.

$$\Pr\left[\tau > \frac{t}{\mathcal{N}}\left(1 + \frac{\psi}{4}\right)\right]$$
$$= \Pr[Y < t] = \Pr\left[Y < E[Y]\left(\frac{1}{1 + \frac{\psi}{4}}\right)\right].$$

Note that $\frac{1}{1+\frac{\psi}{4}} \le 1 - \frac{\psi}{8}$ since $\left(1 + \frac{\psi}{4}\right)\left(1 - \frac{\psi}{8}\right) = 1 + \frac{\psi}{8} - \frac{\psi^2}{32} \ge 1$. This yields

$$\Pr\left[Y < E[Y]\left(\frac{1}{1 + \frac{\psi}{4}}\right)\right] \le \Pr\left[Y < E[Y]\left(1 - \frac{\psi}{8}\right)\right]$$
$$\le e^{-\frac{E[Y]\psi^2}{64}\left(\frac{1}{2}\right)} \quad \text{[Chernoff bound].}$$

Substituting $t = \frac{128}{\psi^2}\ln\left(\frac{3}{\delta}\right)$, we get:

$$\frac{E[Y]\psi^2}{64} = t\left(1+\frac{\psi}{4}\right)\frac{\psi^2}{64} = \frac{128}{64}\ln\left(\frac{3}{\delta}\right)\left(1+\frac{\psi}{4}\right) \geq 2\ln\left(\frac{3}{\delta}\right).$$

Thus,

$$\Pr\left[\tau > \frac{t}{\mathcal{N}}\left(1+\frac{\psi}{4}\right)\right] \leq e^{-\ln\left(\frac{3}{\delta}\right)} = \frac{\delta}{3}. \quad \square$$

We next prove results about the false negatives and false positives. In order to do so, we need the following corollaries of the Chernoff bound. Let X be any binomial random variable, i.e. $X = \sum_{i=1}^{n} X_i$ where the X_i are independent 0–1 random variables. The common form of the Chernoff bound expresses the tail probabilities of X as a function of the expectation $\mu = E[X]$. In our cases, $E[X]$ is not known exactly, but a range $[\mu_L, \mu_H]$ is known such that $\mu_L \leq \mu \leq \mu_H$. In such a case, the following inequalities are useful.

Lemma 3.2. *For any* $0 < \delta \leq 1$,

$$\Pr[X \geq (1+\delta)\mu_H] \leq \exp\left(\frac{-\mu_H\delta^2}{3}\right).$$

Proof. Let $M_X(\gamma)$ be the moment generating function of X.

$$\begin{aligned}
&\Pr[X \geq (1+\delta)\mu_H] \\
&= \Pr[e^{\gamma X} \geq e^{\gamma(1+\delta)\mu_H}] \quad \text{for any } \gamma > 0 \\
&\leq \frac{E[e^{\gamma X}]}{e^{\gamma(1+\delta)\mu_H}} \quad \text{[by Markov inequality]} \\
&= \frac{M_X(\gamma)}{e^{\gamma(1+\delta)\mu_H}} \\
&\leq \frac{e^{(e^\gamma-1)\mu}}{e^{\gamma(1+\delta)\mu_H}} \quad [\text{since } M_X(\gamma) \leq e^{(e^\gamma-1)\mu}, \text{ as proved in [19, page 64]}] \\
&\leq \frac{e^{(e^\gamma-1)\mu_H}}{e^{\gamma(1+\delta)\mu_H}} \\
&\quad [\text{since } \mu_H \geq \mu, \text{ and } \gamma > 0 \Rightarrow (e^\gamma - 1) > 0 \Rightarrow e^{(e^\gamma-1)} > 1].
\end{aligned}$$

Substituting $\gamma = \ln(1+\delta) > 0$, we get (for any $\delta > 0$),

$$\Pr[X \geq (1+\delta)\mu_H] \leq \left(\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}\right)^{\mu_H}. \tag{3}$$

It is proved in [19, page 65], that for any $0 < \delta \leq 1$,

$$\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}} \leq \exp\left(\frac{-\delta^2}{3}\right).$$

Combining this with inequality (3), the claim follows. $\square$

Lemma 3.3. *For any* $0 < \delta < 1$,

$$\Pr[X \leq (1-\delta)\mu_L] \leq \exp\left(\frac{-\mu_L\delta^2}{2}\right).$$

Proof.

$$\begin{aligned}
&\Pr[X \leq (1-\delta)\mu_L] \\
&= \Pr[e^{\gamma X} \geq e^{\gamma(1-\delta)\mu_L}] \quad \text{for any } \gamma < 0 \\
&\leq \frac{E[e^{\gamma X}]}{e^{\gamma(1-\delta)\mu_L}} \quad \text{[by Markov inequality]} \\
&= \frac{M_X(\gamma)}{e^{\gamma(1-\delta)\mu_L}} \quad [M_X(\gamma) \text{ is the moment generating function of } X] \\
&\leq \frac{e^{(e^\gamma-1)\mu}}{e^{\gamma(1-\delta)\mu_L}} \quad [\text{since } M_X(\gamma) \leq e^{(e^\gamma-1)\mu}] \\
&\leq \frac{e^{(e^\gamma-1)\mu_L}}{e^{\gamma(1-\delta)\mu_L}} \quad [\text{since } \mu_L \leq \mu, \text{ and } \gamma < 0 \\
&\qquad \Rightarrow (e^\gamma - 1) < 0 \Rightarrow 0 < e^{(e^\gamma-1)} < 1].
\end{aligned}$$

Substituting $\gamma = \ln(1-\delta) < 0$, we get, for any $0 < \delta < 1$,

$$\Pr[X \leq (1-\delta)\mu_L] \leq \left(\frac{e^{-\delta}}{(1-\delta)^{(1-\delta)}}\right)^{\mu_L}. \tag{4}$$

It is proved in [19, page 66], that for any $0 \leq \delta < 1$,

$$\frac{e^{-\delta}}{(1-\delta)^{(1-\delta)}} \leq \exp\left(\frac{\delta^2}{2}\right).$$

Combining this with inequality (4), the claim follows. $\square$

The following lemmas provide upper bounds on the probabilities of finding a false negative and a false positive respectively, in a centralized setting.

Lemma 3.4. *If v is a frequent item, i.e. $f_v \geq \phi\mathcal{N}$, then with probability at least $1-\delta$, v occurs at least $\left(\phi - \frac{\psi}{2}\right)t$ times in M^t.*

Proof. Let E_1 be the event that v occurs at least $\left(\phi - \frac{\psi}{2}\right)t$ times in M^t. Let E_2 be the event $\tau \geq \frac{t}{\mathcal{N}}\left(1-\frac{\psi}{4}\right)$. Let Z be a random variable indicating the number of copies of v with weight $\frac{t}{\mathcal{N}}\left(1-\frac{\psi}{4}\right)$ or smaller. Let E_3 be the event that $Z \geq \left(\phi - \frac{\psi}{2}\right)t$. We observe that if E_2 and E_3 are true, then E_1 is also true. From Lemma 3.1, we know that $\Pr[\bar{E_2}] < \frac{\delta}{3}$. Using these:

$$\begin{aligned}
\Pr[E_1] &\geq \Pr[E_2 \wedge E_3] = 1 - \Pr[\bar{E_2} \vee \bar{E_3}] \\
&\geq 1 - \Pr[\bar{E_2}] - \Pr[\bar{E_3}] \\
&\geq 1 - \frac{\delta}{3} - \Pr\left[Z < \left(\phi - \frac{\psi}{2}\right)t\right].
\end{aligned} \tag{5}$$

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

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

Applying Lemma 3.3 with $\mu_L = \left(\phi - \frac{\psi}{4}\right)t$ (note that $\mu = E[Z] \geq \mu_L$), we get

$$\begin{aligned}
&\Pr\left[Z < \left(\phi - \frac{\psi}{2}\right)t\right] \\
&= \Pr\left[Z < \left(\phi - \frac{\psi}{4}\right)t\left(1 - \frac{\psi}{4\phi - \psi}\right)\right] \\
&\leq \exp\left(\frac{-\left(\phi - \frac{\psi}{4}\right)t\left(\frac{\psi}{4\phi-\psi}\right)^2}{2}\right) \\
&\leq \exp\left(\frac{-4\ln\left(\frac{3}{\delta}\right)}{\phi - \frac{\psi}{4}}\right) \quad \left[\text{substituting } t = \frac{128}{\psi^2}\ln\left(\frac{3}{\delta}\right)\right] \\
&< \exp\left(-4\ln\left(\frac{3}{\delta}\right)\right) \quad \left[\text{since } 0 < \phi - \frac{\psi}{4} < 1\right] \\
&= \left(\frac{\delta}{3}\right)^4.
\end{aligned} \tag{6}$$

Substituting (6) in (5):

$$\Pr[E_1] \geq 1 - \frac{\delta}{3} - \left(\frac{\delta}{3}\right)^4 \geq 1 - \delta. \quad \square$$

Lemma 3.5. *If u is an infrequent item, i.e. $f_u < (\phi - \psi)\mathcal{N}$, then, with probability at least $1 - \delta$, u occurs less than $\left(\phi - \frac{\psi}{2}\right) t$ times in M^t.*

Proof. Let Y denote the number of copies of u with weight $\leq \frac{t}{\mathcal{N}}\left(1 + \frac{\psi}{4}\right)$. Let E denote the event that u occurs less than $\left(\phi - \frac{\psi}{2}\right) t$ times in M^t. As in the proof of Lemma 3.4, using Lemma 3.1, we get:

$$\begin{aligned}\Pr[E] &\geq \Pr\left[\left(\tau \leq \frac{t}{\mathcal{N}}\left(1 + \frac{\psi}{4}\right)\right) \wedge \left(Y \leq \left(\phi - \frac{\psi}{2}\right) t\right)\right] \\ &\geq 1 - \Pr\left[\tau > \frac{t}{\mathcal{N}}\left(1 + \frac{\psi}{4}\right)\right] - \Pr\left[Y > \left(\phi - \frac{\psi}{2}\right) t\right] \\ &\geq 1 - \frac{\delta}{3} - \Pr\left[Y > \left(\phi - \frac{\psi}{2}\right) t\right]. \qquad (7)\end{aligned}$$

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

$$\begin{aligned}E[Y] &\leq (\phi - \psi)\mathcal{N}\left(\frac{t}{\mathcal{N}}\left(1 + \frac{\psi}{4}\right)\right) \\ &\leq t\left[\phi - \frac{3\psi}{4} - \frac{\psi^2}{4}\right] \leq t\left[\phi - \frac{3\psi}{4}\right].\end{aligned}$$

Applying Lemma 3.2 with $\mu_H = \left(\phi - \frac{3\psi}{4}\right) t$ (note that $\mu = E[Z] \leq \mu_H$), we get

$$\begin{aligned}&\Pr\left[Y > \left(\phi - \frac{\psi}{2}\right) t\right] \\ &= \Pr\left[Y > \left(\phi - \frac{3\psi}{4}\right) t \left(1 + \frac{\psi}{4\phi - 3\psi}\right)\right] \\ &\leq \exp\left(\frac{-\left(\phi - \frac{3\psi}{4}\right) t \left(\frac{\psi}{4\phi - 3\psi}\right)^2}{3}\right) \\ &\leq \exp\left(\frac{-8\ln\left(\frac{3}{\delta}\right)}{3\left(\phi - \frac{3\psi}{4}\right)}\right) \quad \left[\text{substituting } t = \frac{128}{\psi^2}\ln\left(\frac{3}{\delta}\right)\right] \\ &< \exp\left(-\frac{8}{3}\ln\left(\frac{3}{\delta}\right)\right) \quad \left[\text{since } 0 < \phi - \frac{3\psi}{4} < 1\right] \\ &= \left(\frac{\delta}{3}\right)^{\frac{8}{3}}. \qquad (8)\end{aligned}$$

Substituting (8) in (7), we get

$$\Pr[E] \geq 1 - \frac{\delta}{3} - \left(\frac{\delta}{3}\right)^{\frac{8}{3}} \geq 1 - \delta. \quad \square$$

3.3. Analysis of gossip

Consider an item θ that is disseminated through gossip in the asynchronous model. At the start, θ is with one node, and in subsequent rounds it is disseminated to the other nodes. Let T^N be the number of rounds till θ is disseminated to all the N nodes.

Lemma 3.6. $E[T^N] = 2N\ln N + O(N)$.

Proof. Let ξ_i be the set of nodes that have θ after i rounds. Thus ξ_0 has only one node (the one that sampled θ during the initialization step). For $j = 1 \ldots N - 1$, let random variable X_j be the number of rounds required to increase the number of nodes that have θ from j to $j + 1$.

$$T^N = \sum_{j=1}^{N-1} X_j.$$

For $i \geq 1$, in round i, a new node receives θ if a gossip message is transmitted from node α to node β, where $\alpha \in \xi_{i-1}$ and $\beta \notin \xi_{i-1}$. Thus X_j 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 $\alpha \in \xi_{i-1}$ to a node $\beta \notin \xi_{i-1}$. The probability of a success is thus $\left(\frac{j}{N}\right)\left(1 - \frac{j}{N}\right) = \frac{j(N-j)}{N^2}$. A geometric random with probability of success p has expectation $1/p$, so we get $E[X_j] = \frac{N^2}{j(N-j)}$. Using linearity of expectation, we get:

$$\begin{aligned}E[T^N] &= \sum_{j=1}^{N-1} E[X_j] = \sum_{j=1}^{N-1} \frac{N^2}{j(N-j)} = N\sum_{j=1}^{N-1}\left(\frac{1}{j} + \frac{1}{N-j}\right) \\ &= 2N\sum_{j=1}^{N-1}\frac{1}{j} = 2NH_{N-1} = 2N\ln N + O(N)\end{aligned}$$

where H_k denotes the kth Harmonic number. $\square$

Our proof for high-probability bounds on T^N uses a result about the *coupon collector* problem. Suppose there are coupons of Λ distinct types, labeled $\{1, 2, \ldots, \Lambda\}$, and one has to draw coupons at random (with replacement) 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. We present a high probability bound on the number of trials to collect all Λ coupons.

Lemma 3.7. *Let the random variable $\mathcal{C}_\Lambda$ denote the number of trials to collect at least one coupon of each of Λ types. Then,*

$$\Pr[\mathcal{C}_\Lambda > 3\Lambda\ln\Lambda] \leq \frac{1}{\Lambda^2}.$$

Proof. Let E_i denote the event that the coupon with label $i \in \{1, 2, \ldots, \Lambda\}$ did not get drawn at all in $3\Lambda\ln\Lambda$ trials. Then,

$$\begin{aligned}\Pr[\mathcal{C}_\Lambda > 3\Lambda\ln\Lambda] &= \Pr\left(\bigcup_{i=1}^{\Lambda} E_i\right) \leq \sum_{i=1}^{\Lambda}\Pr(E_i) \quad \text{(union bound)} \\ &= \sum_{i=1}^{\Lambda}\left(1 - \frac{1}{\Lambda}\right)^{3\Lambda\ln\Lambda} \leq \sum_{i=1}^{\Lambda}\left(e^{-\frac{1}{\Lambda}}\right)^{3\Lambda\ln\Lambda} \\ &\leq \sum_{i=1}^{\Lambda}\frac{1}{\Lambda^3} = \frac{1}{\Lambda^2}. \quad \square\end{aligned}$$

Lemma 3.8.

$$\Pr[T^N > 12N\ln 2N] \leq \frac{1}{2N^2}.$$

Proof. The dissemination of θ can be divided into two phases. The first phase starts with the first transmission and 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 easy to find a destination that does not have θ. Once θ has reached $\frac{N}{2}$ nodes, the situation reverses. Let T_1 and T_2 be the number of rounds taken by the two phases, respectively.

More formally, let X_j be defined as in the proof of Lemma 3.6. Let $T_1 = \sum_{j=1}^{N/2} X_j$, and $T_2 = \sum_{j=\frac{N}{2}+1}^{N-1} X_j$. Clearly, we have $T^N = T_1 + T_2$.

To bound T^N, we note that if $T^N > 12N \ln 2N$, then at least one of T_1 or T_2 should be greater than $6N \ln 2N$. In Lemma 3.9, we show that T_1 and T_2 are bounded by $6N \ln 2N$, with high probability. Thus,

$$\begin{aligned}\Pr[T^N > 12N \ln 2N] &\le \Pr[(T_1 > 6N \ln 2N) \cup (T_2 > 6N \ln 2N)] \\ &\le \Pr[T_1 > 6N \ln 2N] + \Pr[T_2 > 6N \ln 2N] \\ &\quad \text{(union bound)} \\ &\le \frac{1}{4N^2} + \frac{1}{4N^2} \quad \text{(Lemma 3.9)} \\ &= \frac{1}{2N^2}. \quad \square\end{aligned}$$

Lemma 3.9.

$$\Pr[T_1 > 6N \ln 2N] \le \frac{1}{4N^2}$$

$$\Pr[T_2 > 6N \ln 2N] \le \frac{1}{4N^2}.$$

Proof. For $0 < p \le 1$, let $G(p)$ denote a geometric random variable with parameter p. From the proof of Lemma 3.6, we know that $X_j = G\left(\frac{j(N-j)}{N^2}\right)$. Thus, T_1 is the sum of independent geometric random variables.

$$T_1 = \sum_{j=1}^{N/2} G\left(\frac{j(N-j)}{N^2}\right).$$

Consider the random variable $\mathcal{C}_{2N}$, the number of trials needed to collect $2N$ coupons.

$$\mathcal{C}_{2N} = \sum_{j=1}^{2N} G\left(\frac{2N-j+1}{2N}\right).$$

Let random variable $\mathcal{C}'$ consist of the last few terms of $\mathcal{C}_{2N}$:

$$\mathcal{C}' = \sum_{j=1}^{N/2} G\left(\frac{j}{2N}\right).$$

Note that for $j = 1, \ldots, N/2$, we have $(N-j) \ge \frac{N}{2}$, and hence $\frac{j(N-j)}{N^2} \ge \frac{j}{2N}$. Thus we can write $\mathcal{C}'$ and T_1 as follows:

$$\mathcal{C}' = \sum_{j=1}^{N/2} G(y_j)$$

$$T_1 = \sum_{j=1}^{N/2} G(x_j)$$

such that $x_j \ge y_j$, for all $j = 1, \ldots, \frac{N}{2}$. We also know that if $0 < y \le x \le 1$, then for any $\gamma > 0$, $\Pr[G(y) \ge \gamma] \ge \Pr[G(x) \ge \gamma]$. From Lemma 3.10, we have that $\mathcal{C}'$ stochastically dominates T_1, i.e. for each $\gamma \ge 0$, $\Pr[\mathcal{C}' \ge \gamma] \ge \Pr[T_1 \ge \gamma]$. Since $\mathcal{C}_{2N} \ge \mathcal{C}'$, we have:

$$\begin{aligned}\Pr[T_1 > 6N \ln 2N] &\le \Pr[\mathcal{C}' > 6N \ln 2N] \\ &\le \Pr[\mathcal{C}_{2N} > 6N \ln 2N] \le \frac{1}{4N^2}\end{aligned}$$

where we have used Lemma 3.7. The proof for T_2 follows similarly. $\square$

Lemma 3.10. *Suppose Y and Z are random variables defined as follows. $Y = \sum_{i=1}^{k} Y_i$, and $Z = \sum_{i=1}^{k} Z_i$, where the Y_i s are mutually independent, the Z_i s are mutually independent, and for every $i = 1, \ldots, k$, Y_i stochastically dominates Z_i, i.e. for every $\gamma \ge 0$, $\Pr[Y_i \ge \gamma] \ge \Pr[Z_i \ge \gamma]$. Then, Y stochastically dominates Z, i.e. for each $\gamma \ge 0$, $\Pr[Y \ge \gamma] \ge \Pr[Z \ge \gamma]$.*

Proof. For $i = 1 \ldots k$, let f_i and g_i be the cumulative distribution functions of Y_i and Z_i respectively.

$$f_i(\gamma) = \Pr[Y_i \le \gamma]$$
$$g_i(\gamma) = \Pr[Z_i \le \gamma].$$

Consider any $i \in \{1, 2, \ldots, k\}$. We know that for each $\gamma \ge 0$, $f_i(\gamma) = 1 - \Pr[Y_i > \gamma] \le 1 - \Pr[Z_i > \gamma] = g_i(\gamma)$. Thus, $f_i(\gamma) \le g_i(\gamma)$.

We can view Y_i and Z_i as random variables in the same sample space as follows. For $i = 1 \ldots k$, let U_i be a number chosen uniformly at random from $(0, 1)$. Let random variables $Y_i' = f_i^{-1}(U_i)$, and $Z_i' = g_i^{-1}(U_i)$. It is easy to see that for every outcome for U_i, $f_i^{-1}(U_i) \ge g_i^{-1}(U_i)$. Thus random variables Y_i' and Z_i' satisfy $Y_i' \ge Z_i'$. If the outcomes U_i, $i = 1, \ldots, k$ are all independent, then the Y_i's are mutually independent, and the Z_i's are mutually independent. We observe that for every $\gamma \ge 0$,

$$\begin{aligned}\Pr[Y_i' \le \gamma] &= \Pr[f_i^{-1}(U_i) \le \gamma] \\ &= \Pr[U_i \le f_i(\gamma)] = f_i(\gamma) = \Pr[Y_i \le \gamma].\end{aligned}$$

Hence, Y_i' and Y_i have identical distributions. Similarly, Z_i' and Z_i have identical distributions. Now, consider Y' and Z' defined as follows:

$$Y' = \sum_{i=1}^{k} Y_i'$$

$$Z' = \sum_{i=1}^{k} Z_i'.$$

Since the Y_i's are mutually independent, Y' has the same distribution as Y, and similarly Z' has the same distribution as Z. Further, for each outcome in the above sample space, $Y' \ge Z'$. This implies that for each $\gamma \ge 0$, $\Pr[Y \ge \gamma] \ge \Pr[Z \ge \gamma]$. $\square$

We now present a bound on the dissemination time of the smallest weights. Let $\mathcal{T}^t$ denote the time taken for all items in M^t to be disseminated to all nodes.

Lemma 3.11. $\Pr[\mathcal{T}^t > 12N \ln 2N] \le \frac{1}{2N}$.

Proof. For $i = 1 \ldots N$, let $M_i^t = M_i \cap M^t$ i.e. the set of all elements at node i which have been assigned weights among the smallest t weights. Note that in the algorithm in Fig. 1, all elements in M_i^t are transmitted together, i.e., in each round, either all elements in M_i^t are transmitted, or none of them are; thus, the upper bound on T^N also applies to the dissemination time of M_i^t. Let E_i denote the event that M_i^t is not disseminated to all nodes in $12N \ln 2N$ rounds. From Lemma 3.8, we have $\Pr[E_i] \le \frac{1}{2N^2}$.

$$\begin{aligned}\Pr[\mathcal{T}^t > 12N \ln 2N] &= \Pr\left[\bigcup_{i=1}^{N} E_i\right] \\ &\le \sum_{i=1}^{N} \Pr[E_i] \quad \text{(union bound)} \\ &\le N \cdot \frac{1}{2N^2} = \frac{1}{2N}. \quad \square\end{aligned}$$

We now present the main theorem on the correctness of the algorithm in Fig. 1.

Input at node i**:** Data set M_i, error probability δ, frequency threshold k, approximation error λ

1. **Initialization**
   (a) $S_i \leftarrow \Phi$
   (b) for $\ell = 1$ to $\mathcal{N}_i$
      i. Choose ρ as a uniformly distributed random number in $(0, 1)$.
      ii. If $\rho < \frac{12k}{\lambda^2} \ln \frac{2}{\delta}$ then $S_i \leftarrow S_i \cup \{(i, \ell, m_i^l)\}$
2. **Gossip**
   In each round:
   (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 $\{1, \ldots, N\}$ and send S_i to j
3. **Query**
   When asked for the frequent items, report all items which occur more than $r = \frac{12k^2}{\lambda^2}(1 - \frac{\lambda}{2k}) \ln \frac{2}{\delta}$ times in S_i.

Fig. 2. Gossip algorithm at node *i* for finding the frequently occurring items with an absolute threshold *k*.

Theorem 3.1. *Suppose the distributed algorithm in Fig. 1 is run for* $12N \ln 2N$ *rounds. Then, with probability at least* $1 - \delta$*, an item of frequency* $\phi\mathcal{N}$ *or more in* M *will be identified as a frequent item at every node. Similarly, with probability at least* $1 - \delta$*, an item with frequency less than* $(\phi - \psi)\mathcal{N}$ *will not be identified as a frequent item at any node.*

Proof. From Lemma 3.11, we have that all elements in M^t are disseminated to every node in the network after $12N \ln 2N$ rounds. The theorem follows from Lemmas 3.5 and 3.4. □

Since the size of the sketch at any time during gossip is at most $t = \frac{128}{\psi^2} \ln\left(\frac{3}{\delta}\right)$, the number of bytes exchanged in each round is $O\left(\frac{1}{\psi^2} \ln\left(\frac{1}{\delta}\right)\right)$. Hence, we get the following result on the communication complexity, using Lemma 3.11.

Theorem 3.2. *The number of bytes exchanged by the algorithm in Fig. 1 till the frequent items are identified is at most* $O(\frac{1}{\psi^2} \ln(\frac{1}{\delta}) N \ln N)$*, with probability* $1 - O\left(\frac{1}{N}\right)$*.*

4. Frequent items with an absolute threshold

We now present an algorithm in the asynchronous model for identifying items whose frequency is greater than a user-specified absolute threshold k. As in Section 3, let $M = \bigcup_{i=1}^{N} M_i$ denote the multi-set of all input values. The goal is to output all items v such that $f_v \geq k$ without outputting any item v such that $f_v < k - \lambda$.

The intuition is as follows. Similar to the algorithm for relative threshold, this algorithm is based on random sampling. Unlike the algorithm for relative threshold, the sampling probability can be statically decided by the nodes, based on k and λ. The elements of M are sampled in a distributed manner, and the sampled elements are disseminated using gossip. Intuitively, suppose we sample each element from M into a set S with probability $1/k$. For a frequent item v with $f_v \geq k$, we (roughly) expect one or more copies of v to be present in S. Similarly, for an infrequent item u with $f_u < k - \lambda$, we expect that no copy of u will be included in S. However, some infrequent items may get "lucky" and may be included in S and similarly, some frequent items may not make it to S. The probabilities of these events depend on the sample size.

To refine this sampling scheme, we sample with a probability that is slightly larger than $1/k$, say c/k for some parameter c. Finally, we select those items that occur at least r times within S, for some parameter $r < c$; the value of r will be determined by the analysis. The smaller the value of λ, the greater should be the sampling probability, since we need to make a more precise distinction between the frequencies of frequent and infrequent items. In the actual algorithm, we use a sampling probability of $\frac{12k}{\lambda^2} \ln \frac{2}{\delta}$—note that this is $\Omega\left(\frac{1}{k}\right)$ since $\lambda < k$ and hence $\frac{k}{\lambda^2} > \frac{1}{k}$.

The algorithm for the absolute threshold is shown in Fig. 2. Through our analysis, we give a bound on the number of rounds after which frequent items are likely to be found at all nodes.

4.1. Analysis

We now analyze the correctness and the time complexity of the algorithm in Fig. 2. The plan is as follows. Let M^s be the set of elements in M that are sampled by the nodes, and hence get disseminated through gossip. We first show in Lemma 4.1 that within a small number of rounds, all elements in M^s are disseminated to all nodes. We then show in Lemma 4.2 that for each frequent item, M^s contains sufficient copies of the item (with high probability), thus showing that the probability of a false negative is small. Then, we show in Lemma 4.3 that for each infrequent item, M^s does not contain enough copies of the item to be identified as a frequent item at any node (with high probability), showing that the probability of a false positive is small. Let $\mathcal{T}^s$ denote the time taken for all items in M^s to be disseminated to all nodes.

Lemma 4.1.

$$\Pr[\mathcal{T}^s > 12N \ln 2N] \leq \frac{1}{2N}.$$

Proof. For a single element $\theta \in M^s$ that gets disseminated through gossip, the results of Lemmas 3.6 and 3.8 from the analysis for relative threshold hold, because the underlying gossip mechanism is the same for both the algorithms. For $i = 1 \ldots N$, let $M_i^s = M_i \cap M^s$, i.e., the set of all elements at node *i* which were sampled by node *i*, and hence included in the sketch at node *i* when it was initialized. Note that in the algorithm in Fig. 2, all elements in M_i^s are transmitted together, i.e., in each round, either all the elements in M_i^s are transmitted, or none of them are. Thus, the upper bound on T^N from Lemma 3.8 also applies to the dissemination time of M_i^s. Let E_i denote the event that M_i^s is not disseminated to all nodes in $12N \ln 2N$ rounds. From Lemma 3.8, we have $\Pr[E_i] \leq \frac{1}{2N^2}$.

$$\Pr[\mathcal{T}^s > 12N\ln 2N] = \Pr\left[\bigcup_{i=1}^{N} E_i\right] \le \sum_{i=1}^{N}\Pr[E_i] \le N\cdot\frac{1}{2N^2} = \frac{1}{2N}. \quad \square$$

Lemma 4.2. False negative. *If v is an item with $f_v \ge k$, then with probability at least $1-\delta$, v is returned as a frequent item by every node after $12N\ln 2N$ rounds.*

Proof. Let $r = \frac{12k^2}{\lambda^2}\left(1-\frac{\lambda}{2k}\right)\ln\frac{2}{\delta}$. If v is such that $f_v \ge k$, then v is not reported by a node in the following two situations.

- Less than r copies of v are present in M^s.
- r or more copies of v were sampled into M^s during the 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 are present in M^s. Let E_2 denote the event that after $12N\ln 2N$ rounds, all of M^s was not disseminated to all the nodes in the network. Let E denote the event that there was some node that did not report v as a frequent item.

$$\Pr[E] \le \Pr[E_1\cup E_2] \le \Pr[E_1] + \Pr[E_2]. \tag{9}$$

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 are in M^s. X_v is a binomial random variable with f_v trials and the probability of success in each trial being $\frac{12k}{\lambda^2}\ln\frac{2}{\delta}$. It follows that $E[X_v] = \frac{12k^2}{\lambda^2}\ln\frac{2}{\delta}$. Using Chernoff bounds:

$$\begin{aligned}\Pr[E_1] &= \Pr[X_v < r] = \Pr\left[X_v < \frac{12k^2}{\lambda^2}\left(1-\frac{\lambda}{2k}\right)\ln\frac{2}{\delta}\right]\\ &= \Pr\left[X_v < E[X_v]\left(1-\frac{\lambda}{2k}\right)\right]\\ &\le e^{-\frac{3}{2}\ln\frac{2}{\delta}} = \left(\frac{\delta}{2}\right)^{\frac{3}{2}} < \frac{\delta}{2}.\end{aligned}$$

From Lemma 4.1, we have $\Pr[E_2] < \frac{1}{N}$. Using the bounds on $\Pr[E_1]$ and $\Pr[E_2]$ in inequality (9), and assuming $N > \frac{2}{\delta}$, we get:

$$\Pr[E] \le \frac{\delta}{2} + \frac{1}{N} < \delta. \quad \square$$

Lemma 4.3. False positive. *If u is an item with $f_u \le k-\lambda$, where $k^{\frac{3}{4}} \le \lambda < k$, then the probability that u is returned by some node as a frequent item is no more than δ.*

Proof. A false positive can occur if both these events happen (1) r or more copies of u are present in M^s; let E_1 denote this event and (2) all r copies reach some node in the network through gossip; let E_2 denote this event. Let E denote the event that a false positive occurred.

$$\Pr[E] = \Pr[E_1\cap E_2] \le \Pr[E_1].$$

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 $E[X_u] = (k-\lambda)\frac{12k}{\lambda^2}\ln\frac{2}{\delta} = \frac{12k^2}{\lambda^2}\left(1-\frac{\lambda}{k}\right)\ln\frac{2}{\delta}$. Using Chernoff bounds:

$$\begin{aligned}\Pr[E_1] &= \Pr[X_u > r] = \Pr\left[X_u > \frac{12k^2}{\lambda^2}\left(1-\frac{\lambda}{2k}\right)\ln\frac{2}{\delta}\right]\\ &= \Pr\left[X_u > E[X_u]\left(1+\frac{\frac{\lambda}{2k}}{1-\frac{\lambda}{k}}\right)\right]\\ &\le \exp\left(-\left(\ln\frac{2}{\delta}\right)\left(\frac{1}{1-\frac{\lambda}{k}}\right)\right)\\ &= \left(\frac{\delta}{2}\right)^{\frac{1}{1-\frac{\lambda}{k}}} < \frac{\delta}{2} \quad \left[\text{since } \frac{1}{1-\frac{\lambda}{k}} > 1\right]. \quad \square\end{aligned}$$

Lemmas 4.2, 4.3 and 4.1 together lead to the following theorem about the correctness of the algorithm.

Theorem 4.1. *Suppose the distributed algorithm in Fig. 2 is run for $12N\ln 2N$ rounds. Then, with probability at least $1-\delta$, any item with k or more occurrences in M will be identified as a frequent item at every node. With probability at least $1-\delta$, any item with less than $k-\lambda$ occurrences in M will not be identified as a frequent item at any node.*

We next analyze the communication complexity of gossip. Since each node initializes its sketch with the sampled elements, but accumulates more elements as the gossip proceeds, the sizes of the messages exchanged grow as the algorithm progresses. We note that the number of elements exchanged between two nodes in any round is no more than the total number of sampled elements, hence, the number of rounds of gossip required, times the maximum message size is an upper bound on the communication complexity. Let $\mathcal{Y}$ denote the total number of bytes that need to be exchanged in the network until the frequent items have been identified.

Theorem 4.2 (*Communication Complexity for Absolute Threshold*)**.** *With high probability, $\mathcal{Y} = O\left(\frac{N\mathcal{N}k}{\lambda^2}\ln\frac{1}{\delta}\ln N\right)$.*

Proof. Let Z denote the number of elements sampled during initialization, i.e. $Z = |M^s|$. Z follows a binomial distribution with $\mathcal{N}$ trials, and probability of success in each trial equal to $\frac{12k}{\lambda^2}\ln\left(\frac{2}{\delta}\right)$. Thus $E[Z] = \frac{12\mathcal{N}k}{\lambda^2}\ln\left(\frac{2}{\delta}\right)$. Using Chernoff bounds:

$$\begin{aligned}\Pr\left[Z > \frac{18\mathcal{N}k}{\lambda^2}\ln\left(\frac{2}{\delta}\right)\right] &= \Pr\left[Z > \left(1+\frac{1}{2}\right)\frac{12\mathcal{N}k}{\lambda^2}\ln\left(\frac{2}{\delta}\right)\right]\\ &\le e^{-\frac{\mathcal{N}k}{\lambda^2}\ln\left(\frac{2}{\delta}\right)} = \left(\frac{\delta}{2}\right)^{\frac{\mathcal{N}k}{\lambda^2}}\\ &< \frac{\delta}{2} \quad [\text{since } \mathcal{N} > \lambda, k > \lambda].\end{aligned}$$

We note that $Z\cdot\mathcal{T}^s$ is an upper bound on $\mathcal{Y}$. Thus, if $\mathcal{Y}$ is large, then either Z must be large, or $\mathcal{T}^s$ must be large. More precisely, using the above bound on Z, and Lemma 4.1, we get:

$$\begin{aligned}&\Pr\left[\mathcal{Y} > \frac{216N\mathcal{N}k}{\lambda^2}\ln\left(\frac{2}{\delta}\right)\ln 2N\right]\\ &\le \Pr\left[\left(Z > \frac{18\mathcal{N}k}{\lambda^2}\ln\left(\frac{2}{\delta}\right)\right)\bigcup\left(\mathcal{T}^s > 12N\ln 2N\right)\right]\\ &\le \Pr\left[Z > \frac{18\mathcal{N}k}{\lambda^2}\ln\left(\frac{2}{\delta}\right)\right] + \Pr\left[\mathcal{T}^s > 12N\ln 2N\right]\\ &< \frac{\delta}{2} + \frac{1}{2N} < \delta. \quad \square\end{aligned}$$

5. Simulation results

We used simulation to understand the following aspects of the algorithms that we developed. First, we wanted to know how easy it was to implement these algorithms. Next, since theoretical analysis is a pessimistic worst case analysis, we can expect the performance observed during simulation to be better than the theoretical predictions. We set out to measure by how much is the measured performance better than the theoretical predictions. Specifically, we measured the convergence time, and the error rate of the algorithm (these terms are described precisely below).

5.1. Input data and metrics used

We used two types of datasets for the simulations.

Data set I: Pareto-like distribution. The first one was generated by a Pareto-like distribution. Given the domain of items $[m] = \{1, 2, \ldots, m\}$, to make the frequent-items algorithms applicable, we needed a data distribution that can be made arbitrarily skewed, i.e., we wanted n ($n \ll m$) of the data items from $[m]$ to occur very frequently in the data set. During generating the data, we kept a parameter to specify the number of frequent items. We made all the frequent items equally probable, and the total probability of the frequent items was specified by yet another parameter. For example, if we want the data to have 10 frequent items, and the sum of the probabilities of these 10 items is 0.5, then the probability of generating each of these 10 frequent items would be 0.05. Hence, if we have 5000 nodes, and 100 elements per node, then the expected number of occurrences of each item that we desire to be frequent is $5000 \times 100 \times 0.05 = 25{,}000$. We tried three different dataset sizes: 500,000; 1,000,000 and 1,500,000. For each dataset size, we generated 10 different datasets, and for each of these 10 different datasets (of the same size), we formed the sketch and gossiped it 10 times, to average out the errors due to randomization. So each reading using this distribution (in Figs. 4 and 5) is an average of 100 readings.

Dataset II: Zipfian + uniform distributions. The second dataset was generated from a mixture of Zipfian and uniform distributions. Once again, we fixed a small number (n) of items from $[m]$ that would be generated with high frequency. According to the Zipfian distribution, the probability of the rth most frequent item ($r \in \{1, 2, \ldots, n\}$) was assigned by the following probability mass function:

$$f(r) = \frac{\frac{1}{r}}{\sum_{i=1}^{n} \frac{1}{i}}.$$

The sum of the probabilities of these n frequent items was set to $\theta < 1$. The remaining $m - n$ items from the domain $[m]$ were all assigned equal probability (this is where the uniform distribution came in), which was $\frac{1-\theta}{m-n}$. For each of the relative and absolute error algorithms, we worked with datasets of three different sizes: 500,000; 1,500,000 and 4,000,000. To average out the errors due to randomization, we created 100 different datasets of each size, and for each dataset, we repeated the experiment 50 times, and averaged the error rate over these 5000 runs; so each point in the plots in Figs. 6 and 7 resulted from 5000 repetitions.

We analyze the *convergence time* and the *error rates* of the algorithms as functions of the network size and message cost of the gossip, respectively.

5.2. Convergence time

Informally, a system is defined to have converged if it is in a configuration where every pair of nodes have "seen" each other's sketch, either through direct or indirect communication. More precisely, we define that for two nodes i and j, node i has communicated directly with node j if i sent a message to j. We (recursively) define that node i has communicated indirectly with node j if there exists a node k such that i has communicated directly/indirectly with k and then k has communicated directly/indirectly with j.

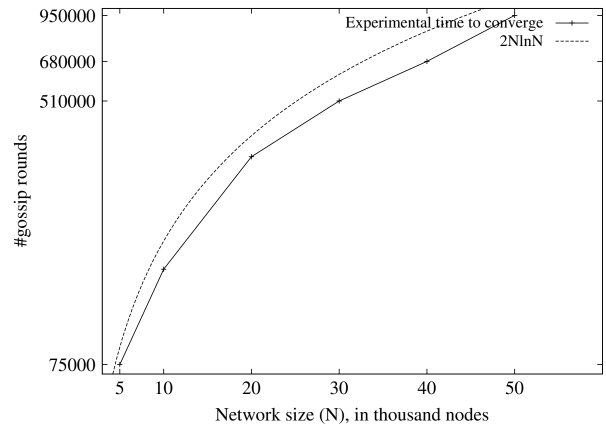


Fig. 3. The number of rounds till convergence versus network size N.

Note that according to the above definition, system convergence is sufficient to ensure that further communication will not lead to any changes in the state of the sketch at any node. In Fig. 3, we plot the number of rounds of gossip required for convergence as a function of the network size N. Because the convergence time for uniform gossip depends only on the network size, the same results apply for both absolute and relative thresholds. We note that all sampled elements at a node are disseminated together. Hence, the convergence time is a function only of the network size N, and does not depend on the size of the dataset, or on the size of the samples.

Error rate. Since the algorithm is a randomized approximation algorithm, there is a small, but non-zero probability that the algorithm will fail, i.e. it would report infrequent items as frequent (false positive) and/or would fail to identify frequent items (false negative). The user specifies the degree of accuracy desired through the approximation error ψ (for relative threshold) or λ (for absolute threshold) and the error probability δ.

We now describe the *error rate* metric that we used for measuring the observed error in our experiments. Note that while the algorithm guarantees a low probability of error, we measure the actual fraction of time that an error occurred during the simulations. The *false negative rate* is defined as the ratio of the number of false negatives reported by a node to the number of data items that are frequent, i.e., what fraction of the frequent items were not identified as frequent by the node. The *false positive rate* is defined as the ratio of the number of false positives reported by a node to the number of data items that are not frequent, but have occurred at least once in the input. The *error rate* is defined as the maximum of the false negative and the false positive rates. Since all nodes attain the same state once convergence occurs, the error rate can be recorded from an arbitrarily selected node (we recorded it from node 0). To see whether we really needed a sketch size as large as predicted by theory, we measured the observed error rate at various sketch sizes.

For relative threshold, the theoretical sketch size was $t = \frac{c_r}{\psi^2} \ln \frac{3}{\delta}$ (with $c_r = 128$ as revealed by the analysis), so we tried sketch sizes for various values of the constant c_r. Figs. 4 and 6 show the error rates as a function of the sketch size, for the Pareto-like distribution and the mixed distribution respectively.

For absolute threshold, the theoretical sampling probability was $\frac{c_a k}{\lambda^2} \ln \frac{2}{\delta}$ (with $c_a = 12$ as revealed by the analysis), so we tried sketch sizes for various values of the constant c_a. Figs. 5 and 7 show the error rates as a function of c_a, for different dataset sizes and corresponding different values of k and λ, for the Pareto-like distribution and the mixed distribution respectively.

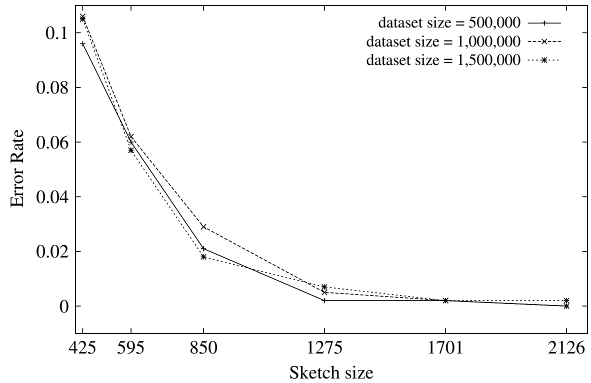


Fig. 4. The error rate as a function of the sketch size for the relative error algorithm, with the dataset generated by the Pareto-like distribution. $\phi = 0.081$, $\psi = 0.02$ and $\delta = 0.1$.

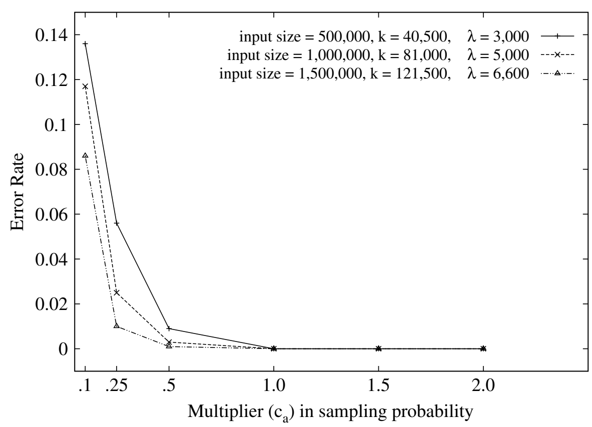


Fig. 5. The error rate as a function of c_a, a multiplier in the sampling probability, for the absolute error algorithm. The dataset is generated by the Pareto-like distribution. Note that the expected sketch size increases linearly with the sampling probability. $\delta = 0.1$.

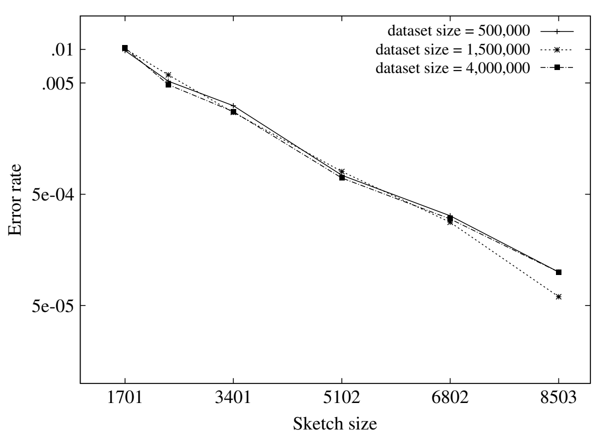


Fig. 6. The error rate as a function of the sketch size for the relative error algorithm, with the dataset generated by the mixed distribution.

5.3. Observations

We make the following observations from our experience with the simulations and the results in Figs. 3–7.

- The theoretical analysis predicted that for a system with N nodes, $12N \ln 2N$ rounds of gossip are sufficient for convergence with high probability. By carrying out simulations with upto 50,000 nodes, we found from simulations, that convergence

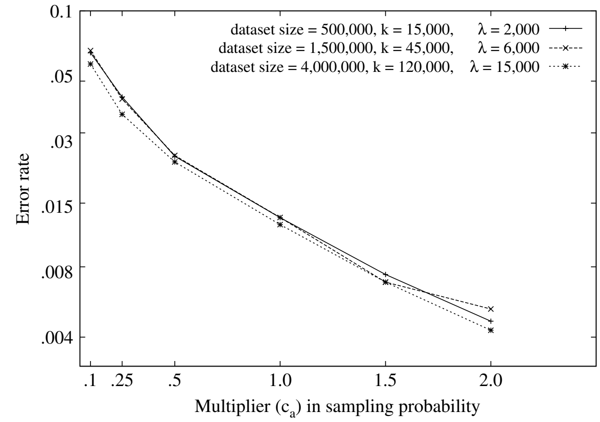


Fig. 7. The error rate as a function of c_a, a multiplier in the sampling probability, for the absolute error algorithm. The dataset is generated by the mixed distribution.

  was typically achieved with less than $2N \ln N$ rounds of gossip for all the values of N that we tried (see Fig. 3).
- For relative threshold, while the analysis showed that in the expression for the sketch size $\frac{c_r}{\psi^2} \ln \frac{3}{\delta}$, a constant factor $c_r = 128$ is necessary, we found in our experiments that a constant factor of $c_r = 0.25$ was sufficient in all cases to meet the desired error bounds. This indicates that, in practice, the required sketch sizes may be much smaller than predicted by theory.
- For absolute threshold, while the analysis showed that, in the expression for the sampling probability $\frac{c_a k}{\lambda^2} \ln \frac{2}{\delta}$, a constant factor $c_a = 12$ is necessary, we found in our experiments that a constant factor of $c_a = 2$ was sufficient in all cases to meet the desired error bounds. This indicates that in practice, the required sampling probability may be smaller than predicted by theory.

6. Synchronous model

In the synchronous communication model, all nodes transmit equally often. 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 [8], 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 6.1 (*[8]*)**.** *Let T^N denote the number of rounds required to spread a rumor among a population of size N. Then,* (1) $\lim_{N\to\infty} \frac{T^N}{\log_2 N} = \ln 2$ *with high probability and,* (2) *For* $\gamma > 0$, $\Pr[T^N > (1 + (\gamma + 1)\ln 2)\log_2 N] = o(N^{-\gamma})$.

Suppose that instead of a single rumor, there were α different rumors originating at different nodes, and all these rumors were being disseminated simultaneously among the N nodes. Let $\mathcal{T}^\alpha$ be the number of rounds required for all the nodes to receive all α rumors.

Lemma 6.1. *With probability* $1 - o\left(\frac{1}{N}\right)$, $\mathcal{T}^\alpha \leq (1 + 2\ln 2)\log_2 N + \ln \alpha$.

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

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

1. **Initialization:**
   (a) $t \leftarrow \frac{128}{\psi^2}\ln(\frac{3}{\delta})$; $S_i \leftarrow \Phi$
   (b) for $\ell = 1$ to $\mathcal{N}_i$
      i. Choose w_i^ℓ as a uniformly distributed random number in $(0,1)$
      ii. set $S_i \leftarrow S_i \cup \{(i, \ell, m_i^\ell, w_i^\ell)\}$
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) 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 $(\phi - \frac{\psi}{2})t$ (nodeID, elementID, value, weight) tuples exist in S_i with value equal to v

Fig. 8. Synchronous gossip algorithm at node *i* for frequent items with a relative threshold.

Input: Data sets M_i, error probability δ, frequency threshold k, approximation error λ

1. **Initialization**
   (a) $S_i \leftarrow \Phi$
   (b) for $l = 1$ to $\mathcal{N}_i$
      i. Choose ρ as a uniformly distributed random number in $(0,1)$.
      ii. If $\rho < \frac{12k}{\lambda^2}\ln\frac{2}{\delta}$ then $S_i \leftarrow S_i \cup \{(i, l, m_i^l)\}$
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) Select node j uniformly at random from $\{1, \ldots, N\}$
   (c) Send S_i to j
3. **Query**
   When asked for the frequent items, 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 items.

Fig. 9. Synchronous gossip algorithm at node *i* for frequent items with absolute threshold *k*.

$$\Pr[\mathcal{T}^\alpha > x] = \Pr\left[\bigcup_{i=1}^{\alpha}(t_i > x)\right] \le \sum_{i=1}^{\alpha}\Pr[t_i > x] = \alpha \Pr[T^N > x].$$

Using $\gamma = 1 + \log_N \alpha$ in Theorem 6.1, we get $\Pr[T^N > (1 + 2\ln 2)\log_2 N + \ln\alpha] = o\left(\frac{1}{N\alpha}\right)$, and the result follows. □

Our algorithms for the synchronous time model for relative and absolute thresholds, are described in Figs. 8 and 9 respectively. These differ from the algorithms for the asynchronous models (Figs. 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 items 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:

Theorem 6.2 (*Synchronous Gossip*)**.** *If the synchronous algorithms in Figs. 8 and 9 are run for* $4\log_2 N$ *rounds, then all frequent items (with relative and absolute thresholds, respectively) will be identified with probability at least* $1 - \delta$*, and no infrequent item will be identified, with probability at least* $1 - \delta$*.*

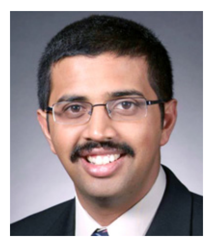
Proof. Similar to the asynchronous time model, in the synchronous model too, the analysis of gossip does not depend on how many elements each node begins with, or how many elements from each node find a place in the sketch; since all the elements from the (local) sketch of a single node get disseminated together. The number of such local sketches is trivially no more than N. If all these items are disseminated to all nodes, then the guarantees will be met. Substituting $\alpha \le N$ in Lemma 6.1 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. □

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.

References

[1] S.P. Boyd, A. Ghosh, B. Prabhakar, D. Shah, Gossip algorithms: design, analysis and applications, in: Proceedings of the IEEE Conference on Computer Communications, INFOCOM, 2005, pp. 1653–1664.
[2] S.P. Boyd, A. Ghosh, B. Prabhakar, D. Shah, Randomized gossip algorithms, IEEE Transactions on Information Theory 52 (6) (2006) 2508–2530.
[3] A.Z. Broder, M. Charikar, A.M. Frieze, M. Mitzenmacher, Min-wise independent permutations (extended abstract), in: Proceedings of the ACM Symposium on Theory of Computing, STOC, 1998, pp. 327–336.
[4] P. Cao, Z. Wang, Efficient top-*k* query calculation in distributed networks, in: Proceedings of the Twenty-Third Annual ACM Symposium on Principles of Distributed Computing, 2004, PODC, pp. 206–215.
[5] S. Deb, M. Médard, C. Choute, Algebraic gossip: a network coding approach to optimal multiple rumor mongering, IEEE Transactions on Information Theory 52 (6) (2006) 2486–2507.
[6] A.J. Demers, D.H. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H.E. Sturgis, D.C. Swinehart, D.B. Terry, Epidemic algorithms for replicated database maintenance, in: Proceedings of the Principles of Distibuted Computing, PODC, 1987, pp. 1–12.
[7] A.G. Dimakis, A.D. Sarwate, M.J. Wainwright, Geographic gossip: efficient aggregation for sensor networks, in: Proceedings of the Fifth International Conference on Information Processing in Sensor Networks, IPSN, 2006, pp. 69–76.
[8] A. Frieze, G. Grimmett, The Shortest-Path Problem for Graphs with Random Arc-Lengths, vol. 10, Elsevier Science Publishers, Besloten, Vennootschap, 1985.
[9] C. Gkantsidis, M. Mihail, A. Saberi, Random walks in peer-to-peer networks, in: Proceedings of the 23rd Conference of the IEEE Communications Society, INFOCOM, 2004.
[10] M. Haridasan, R. van Renesse, Gossip-based distribution estimation in peer-to-peer networks, in: Proceedings of the 7th International Workshop on Peer-to-Peer Systems, IPTPS, 2008.
[11] R.M. Karp, S. Shenker, C.H. Papadimitriou, A simple algorithm for finding frequent elements in streams and bags, ACM Transactions on Database Systems 28 (2003) 51–55.
[12] S.R. Kashyap, S. Deb, K.V.M. Naidu, R. Rastogi, A. Srinivasan, Efficient gossip-based aggregate computation, in: Proceedings of the Twenty-Fifth ACM SIGACT–SIGMOD–SIGART Symposium on Principles of Database Systems, PODS, 2006, pp. 308–317.
[13] D. Kempe, A. Dobra, J. Gehrke, Gossip-based computation of aggregate information, in: Proceedings of the 44th Symposium on Foundations of Computer Science, FOCS, 2003, pp. 482–491.
[14] D. Kempe, J.M. Kleinberg, Protocols and impossibility results for gossip-based communication mechanisms, in: Proceedings of the 43rd Symposium on Foundations of Computer Science, FOCS, 2002, pp. 471–480.
[15] R. Keralapura, G. Cormode, J. Ramamirtham, Communication-efficient distributed monitoring of thresholded counts, in: Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD, 2006, pp. 289–300.
[16] A. Manjhi, V. Shkapenyuk, K. Dhamdhere, C. Olston, Finding (recently) frequent items in distributed data streams, in: Proceedings of the 21st International Conference on Data Engineering, ICDE, 2005, pp. 767–778.
[17] G.S. Manku, R. Motwani, Approximate frequency counts over data streams, in: Proceedings of 28th International Conference on Very Large Data Bases, VLDB, 2002, pp. 346–357.
[18] J. Misra, D. Gries, Finding repeated elements, Science of Computer Programming 2 (2) (1982) 143–152.
[19] M. Mitzenmacher, E. Upfal, Probability and Computing: Randomized Algorithms and Probabilistic Analysis, Cambridge University Press, 2005.
[20] D. Mosk-Aoyama, D. Shah, Computing separable functions via gossip, in: Proceedings of the Twenty-Fifth Annual ACM Symposium on Principles of Distributed Computing, PODC, 2006, pp. 113–122.
[21] S. Venkataraman, D.X. Song, P.B. Gibbons, A. Blum, New streaming algorithms for fast detection of superspreaders, in: Proceedings of the Network and Distributed System Security Symposium, NDSS, 2005.
[22] Q. Zhao, M. Ogihara, H. Wang, J. Xu, 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, pp. 298–307.

Bibudh Lahiri has been a Ph.D. student in the Department of Electrical & Computer Engineering, Iowa State University since Fall 2006. He received his B.E. in Computer Science and Engineering from Jadavpur University in 2002 and worked as a software engineer from 2002 to 2006. His primary research interest is the design of efficient algorithms for processing large data streams and distributed datasets.



Srikanta Tirthapura received his Ph.D. in Computer Science from Brown University in 2002, and his B.Tech. in Computer Science and Engineering from IIT Madras in 1996. He is currently an Associate Professor in the department of Electrical and Computer Engineering at Iowa State University. His research is concerned with data-intensive computing and distributed computing.