Tracking Quantiles of Network Data Streams with Dynamic Operations

Jin Cao, Li Erran Li, Aiyou Chen and Tian Bu
{jincao, erranlli, aychen, tbu}@alcatel-lucent.com

***Abstract*— Quantiles are very useful in characterizing the data distribution of an evolving dataset in the process of data mining or network monitoring. The method of Stochastic Approximation (SA) tracks quantiles *online* by incrementally deriving and updating local approximations of the underly distribution function at the quantiles of interest. In this paper, we propose a generalization of the SA method for quantile estimation that allows not only data insertions, but also dynamic data operations such as deletions and updates.**

I. Introduction

In many network monitoring applications, fast quantile tracking is very useful to detect any abnormal behaviour. However this is no easy matter due to the extreme traffic volume of today's high speed network that often requires algorithms to work in a stream setting, i.e., no data storage and each record can only been seen once. Another difficulty is that the arrived data may contain not only new records, but also updates of the old data as they become obsolete. For instance, there could be a complete removal of the old record (a deletion), or a updated data value of an old record (a updates). An example of this dynamic data is in network flow monitoring where our task is tracking the size of active flows. If there is a new packet arrival from an existing flow, then the flow packet counts will be incremented by 1, or, if there are no packet arrivals from an existing flow for a certain period of time (say 64s), then the flow is subject to deletion.

Stochastic approximation (SA) methods, introduced in a seminal paper by Robbins and Monro [7], are a family of iterative stochastic optimization algorithms that attempts to find zeroes or extrema of functions which cannot be computed directly, but only estimated via noisy observations. Applications of SA to the online quantile estimation problem have been developed in several papers [7], [8], [3], [4], and recently by [2] for simultaneous estimation of multiple quantiles. By viewing the data as generated random from an unknown distribution, the SA algorithm derives the quantile estimate from a local linear approximation of the underlying distribution function in the neighborhood of the quantile.

There are two main advantages of the SA algorithm for quantile estimation that are especially amenable to today's large volume fast changing network data. First, it uses negligible memory. To compute K quantiles, it uses only a space of at most $3K$, no matter how long the streaming data lasts. This memory saving is due to a very simple local approximation that SA employs as compared to other online methods that relies on a representative sample or summary information. [5], [1], Another feature is that the quantiles are *incrementally* updated using a weight assigned to the new data arrival. This allows fast quantile updates and adaptation to non-stationary data with an appropriate choice of weights. As demonstrated by [4], with a constant weight, the SA method for quantile estimation follows the trend of the data and works in a fashion that is similar to Exponentially Weighted Moving Averages.

Unfortunately, the existing SA algorithm for quantile estimation is designed for data that are added one by one. It cannot directly apply to the case when there is dynamic data operations such as erroneous data to be deleted or out-of-date data to be updated. The goal of this paper is to generalize the SA method to allow not only data insertions, but also dynamic data operations such as deletions and updates.

We consider four types of data operations: insertions, deletions, corrections and updates. Both insertions and deletions are self-explanatory. We make a distinction between corrections and updates, where both represent adjustments to old data. However, a corrected data record is still an old record, but a updated record is a new record. This difference is not important for stationary data where all data records are considered equal, but important for time sensitive applications where recent data receives more weight.

We also consider both cases when data are generated randomly from the same distribution (stationary), or data has a time varying distribution (non-stationary). We extensively validate our algorithm using both synthetic and real data, and demonstrate that for both stationary and non-stationary data our algorithm gives accurate quantile estimates. For stationary data, we also demonstrate empirically that our SA quantile estimate converges to the true quantile, similar to the original SA method for data with insertions only.

For the rest of the paper, we review the basic SA algorithm in Section II. We formulate and present our solution in Section III, and evaluate our algorithm in Section IV.

II. Stochastic Approximation: A Preliminary

In this section, we present the basic form of the Stochastic Approximation (SA) algorithm as it applies to quantile estimation with insertion only.

1) Basic Algorithm: Let $\{x_t\}$ be an incoming data stream with a distribution $\mathcal{F}_t$. Let p be a probability whose quantile is of interest, and let θ_t be the true quantile of $\mathcal{F}_t$ w.r.t. p. It is important to understand that the SA quantile estimation is essentially derived from a local linear approximation of the $\mathcal{F}_t$ at θ_t (see [2] Section 2.1.1). Let the distribution (CDF) approximation at time $t-1$ be $\hat{\mathcal{F}}_{t-1}$. At time t, with a new data insertion x_t associated with weight w_t, we obtain an adjusted CDF approximation by the following weighted average

$$\hat{\mathcal{F}}_t(x) \leftarrow (1 - w_t)\hat{\mathcal{F}}_{t-1}(x) + w_t I(x \geq x_t). \qquad (1)$$

Evaluating this at the previous quantile S_{t-1} gives

$$P(X_t \le S_{t-1}) \approx (1-w_t)p + w_t I(x_t \le S_{t-1}) = p_t. \quad (2)$$

Suppose $f_t = \mathcal{F}_t'(\theta_t) > 0$ is the density of F_t at the true quantile (assumed known for now), we can further approximate $\mathcal{F}_t$ locally at (S_{t-1}, p_t) using a linear function with slope f_t, i.e.,

$$\hat{\mathcal{F}}_t(x) \approx (1-w_t)p + w_t I(x_t \le S_{t-1})) + (x - S_{t-1})f_t. \quad (3)$$

Setting the right side of the above to p, we obtained the SA quantile estimate,

$$S_t = S_{t-1} + f_t^{-1} w_t (p - I(x_t \le S_{t-1})). \quad (4)$$

In practice, the derivative f_t in (4) is unknown can only be estimated from the data. Algorithm I summarize the basic SA algorithm using estimates of f_t ([8], [4]).

Algorithm 1 Basic SA Algorithm for Estimating the Quantile of an Input Stream $\{x_t\}$ with Probability p.

1: At time 0, let the initial quantile estimate be S_0, the initial density estimate be f_0.
2: **for** each incoming data x_t **do**
3: Update the quantile probability of S_{t-1}: $p_t = (1-w_t)p + w_t I(S_{t-1} \le x_t)$;
4: Construct a local linear approximation $\hat{\mathcal{F}}_t$ of the distribution function F_t at point (S_{t-1}, p_t) using a line with slope f_t;
5: Update the quantile estimate S_t by the solution to $\hat{\mathcal{F}}_t(S_t) = p$.
6: Update the density estimate f_t: $f_t = (1-w_t)f_{t-1} + w_t(2c)^{-1} I(|x_t - S_t| \le c)$.

2) Choice of Weights and Convergence of SA: The SA algorithm applies to both stationary data, i.e., $\mathcal{F}_t = \mathcal{F}$, and non-stationary data. However, it is important to set w_t, the weight associated with new data arrival in (4), properly for each case. When the data is stationary, two common choices of the weights w_t are $w_t = 1/t$, or $w_t = w$. However, when $\mathcal{F}_t$ is non-stationary, as suggested by [6], [4], we will choose a constant weight $w_t = w$, as the choice of $w_t = 1/t$ is no longer appropriate since it cannot adapt to changes in the data distribution.

It is shown in ([7], [8], [6]) that for stationary data, quantile estimates with $1/t$ weights converge to the true quantile with a rate $O(t^{-\frac{1}{2}})$, but the estimates with constant weight only weakly converge to a distribution with the true quantile as its mean. However, the use of constant weights is still advocated in [4] as their simulation results suggest that it gives a good estimate and is less prone to bad initial values.

III. SA Algorithm with Dynamic Data Operations

We present our solutions for quantile estimation for a network data stream that allows dynamic updates. To simplify the presentation, we shall index the data stream in such a way that at time index t, there is always one and the only one data point x_t that gets inserted. At time t, in addition to insertion of x_t, the data stream is also subject to the following three possible dynamic updates to previous data:

1) *Deletion:* an old data $x_{t'}, t' < t$ is deleted, meaning that $x_{t'}$ is no longer considered a valid record at time t.
2) *Correction:* an old data $x_{t'}, t' < t$ is corrected with a new value x', meaning that the value of $x_{t'}$ is erroneous, and should be replaced with the correct value x'.
3) *Update:* the inserted record x_t is in fact a replacement of an old data $x_{t'}, t' < t$, meaning that the value of $x_{t'}$ at time t' should be deleted.

It is obvious that from the definition 3) above, an data update at index t is equivalent to a deletion of an old data and an insertion of a new data record, and thus does not need special consideration. It is important to note however, that while both correction and update are adjustments to an old data, a corrected data record is still an old record but is considered anew for a updated record. Although such a distinction between a correction and a update is not important for the case when all valid data records at time t are considered equivalent, it matters a lot for time sensitive applications where the most recent data are considered more important. As far as we know, no earlier work has addressed this difference.

Let $\mathcal{F}_t$ be the distribution of $\{x_t\}$ subject to these dynamic adjustments. Our goal is to track the quantile of $\mathcal{F}_t$ w.r.t. a probability p via an online algorithm. To prevent technical difficulties, we shall assume that $\mathcal{F}_t(\cdot)$ is a strictly monotone, and has positive derivatives on its domain. This constraint can be alleviated for discrete distributions by adding a small random noise to data. We present our solution in the following.

A. Basic Idea

We illustrate the basic idea of our algorithm using data with insertions and deletions only. The main difficulty here is how to reverse the effect of insertion at the later time of deletion.

Suppose prior to time t there is no deletion, and at time t, x_t is deleted immediately after its insertion. Assume that there is no local line approximation (Eq. 3) in the basic SA algorithm (Algorithm I), then the effect of insertion of x_t can be simply reversed by the following

$$\hat{\mathcal{F}}_t(x) \leftarrow (1-w_t)^{-1}\left(\hat{\mathcal{F}}_t(x) - w_t I(x \ge x_t)\right), \quad (5)$$

where $\hat{\mathcal{F}}_t(x)$ is the updated CDF approximation in (1). After this, we obtain $\hat{\mathcal{F}}_t(x) = \hat{\mathcal{F}}_{t-1}(x)$, which gives the desired result. Of cause, in reality, the above only holds approximately true due to the local line approximation.

Suppose now instead at time t, we need to delete a data $x_{t_0}, t_0 < t$. It is easy to see from (1) that the original weight w_{t_0} of x_{t_0} at time t_0 diminishes after each insertion of subsequent data, $x_{t_0+1}, \ldots, x_t$. In fact, at time t its weight is reduced to $w_t^{new} = w_{t_0}\Pi_{s=t_0+1}^{t}(1-w_s)$. To delete x_{t_0} at time t, we can use a similar method as in (5)

$$\hat{\mathcal{F}}_t(x) \leftarrow (1-w_t^{new})^{-1}\left(\hat{\mathcal{F}}_t(x) - w_t^{new} I(x \ge x_t)\right). \quad (6)$$

B. Algorithm

Let $\{w_t\}$ be the weight associated with x_t at the time of insertion. The weight associated with x_t after its insertion at a later time will decay due to later insertions. For an old data x_{t_0}, its weight at time $t, t > t_0$, denoted by $d_{t_0}(t)$ is in fact,

$$d_{t_0}(t) = w_{t_0}\Pi_{s=t_0+1}^{t}(1-w_s). \quad (7)$$

Suppose at time $t-1$, our CDF approximation is $\hat{\mathcal{F}}_{t-1}$. At time t, suppose there is a deletion of an earlier data x_{t_0}, and a correction of an earlier data x_{t_1} with value x'_{t_1}. To remove the contribution of deleted data to the CDF approximation, we need to track a value $D_t, 0 \le D_t < 1$ which represents the total weights of deleted data at time t. Due to deletion, the total weights of data that contributed to $\hat{\mathcal{F}}_t$ at time t is not 1, but $1 - D_t$.

Define $D_0 = 0$. At time t, with the new data arrival x_t, we update CDF approximation $\hat{\mathcal{F}}_{t-1}$ by

$$\text{Insert: } \begin{cases} \hat{\mathcal{F}}_t(x) & \leftarrow & \frac{(1-w_t)(1-D_{t-1})\hat{\mathcal{F}}_{t-1}(x)+w_t I(x \ge x_t)}{1-D_{t-1}(1-w_t)}, \\ D_t & \leftarrow & (1-w_t)D_{t-1}. \end{cases} \tag{8}$$

If there is a deletion of x_{t_0} at time t, then we further update the distribution approximation at t by

$$\text{Delete: } \begin{cases} \hat{\mathcal{F}}_t(x) & \leftarrow & \frac{(1-D_t)\hat{\mathcal{F}}_t(x)-d_{t_0}(t)I(x \ge x_{t_0})}{1-D_t-d_{t_0}(t)}, \\ D_t & \leftarrow & D_t + d_{t_0}(t), \end{cases} \tag{9}$$

where $d_{t_0}(t)$ is defined in (7). Or if there is an correction of x_{t_1} at time t with a new value x'_{t_1}, then we further update $\hat{\mathcal{F}}_t$ by

$$\text{Correction: } \begin{cases} \hat{\mathcal{F}}_t(x) & \leftarrow & \frac{(1-D_t)\hat{\mathcal{F}}_t(x)+d_{t_1}(t)(I(x \ge x_{t_1})-I(x \ge x'_{t_1}))}{1-D_t} \\ D_t & & \text{remains unchanged}, \end{cases} \tag{10}$$

The Insertion equation (8) essentially states that with the arrival of new data x_t, $\hat{\mathcal{F}}_t$ is the weighted sum of $I(x \ge x_t)$ from x_t with weight w_t, and $\hat{\mathcal{F}}_{t-1}$ with weight $(1-w_t)(1-D_{t-1})$, normalized to have a total weight of 1. Therefore the actual weight for x_t is $w_t/(1-D_{t-1}(1-w_t))$, which is larger than w_t due to deletion. After this, the weights of the deleted data in $\hat{\mathcal{F}}_t$, D_t, is now updated by a factor of $(1-w_t)$. Similarly, what the delete equation (9) does is to simply remove the influence of x_{t_0} at time t as the weight of x_{t_0} now reduces to $d_{t_0}(t)$. Finally, the correction equation (10) is a result of deletion and re-insertion for the record value at time t_1.

To extend the basic SA algorithm (Algorithm 1) to allow dynamic data operations, we simply need to replace the probability update in line 3 by evaluating the updated $\hat{\mathcal{F}}_t(x)$ in (8), (9) and (10) (depending on the situation), at previous quantile estimate S_{t-1} (similar to (**??**)). Denote this value by p_t. With $D_0 = 0$, this implies the following probability update equations. At time t, in the case of an data insertion of x_t, we update p_t and D_t by

$$\text{Insert: } \begin{cases} p_t \leftarrow (1-D_{t-1}(1-w_t))^{-1} \\ \quad ((1-w_t)(1-D_{t-1})p + w_t I(S_{t-1} \ge x_t)), \\ D_t \leftarrow (1-w_t)D_{t-1}. \end{cases} \tag{11}$$

If there is a deletion of x_{t_0} at time t, then we further update p_t and D_t by

$$\text{Delete: } \begin{cases} p_t \leftarrow (1-D_t-d_{t_0}(t))^{-1} \\ \quad ((1-D_t)p - d_{t_0}(t)I(S_{t-1} \ge x_{t_0})), \\ D_t \leftarrow D_t + d_{t_0}(t), \end{cases} \tag{12}$$

where $d_{t_0}(t)$ is defined in (7). Or if there is an update of x_{t_1} at time t with a new value x'_{t_1}, then we further update p_t and D_t by

$$\begin{aligned} \text{Correction: } \quad p_t \leftarrow (d_{t_1}(t)(I(S_{t-1}(i) \ge x_{t_1}) - I(S_{t-1} \ge x'_{t_1})) \\ +(1-D_t)p)(1-D_t)^{-1}. \end{aligned} \tag{13}$$

After the probability update step, we can use the same local linear approximation method in Algorithm 1 to derive the updated quantile estimate. We summarize our algorithm in Algorithm 2. As for the density estimate step (line 6), it is similar to that in Algorithm 1 by ignoring all deleted data. Although this is not a crucial step, one do need to be careful especially with a small density estimate as it affects the stability of the algorithm.

Algorithm 2 SA Algorithm for Estimating the Quantile of an Input Stream with Dynamic Data Operations for a Probability p.

1: At time 0, let the initial quantile estimate be S_0, the initial density estimate be f_0, and set $D_0 = 0$.
2: **for** each time t **do**
3: Update the quantile probability p_t of S_{t-1} and D_t according to (11), (12) and (13) if appropriate;
4: Construct a local linear approximation $\hat{\mathcal{F}}_t$ of the distribution function F_t at point (S_{t-1}, p_t) using a line with slope f_t;
5: Update the quantile estimate S_t by the solution to $\hat{\mathcal{F}}_t(S_t) = p$.
6: Update the density estimate f_t: $f_t = (1-w_t)f_{t-1} + w_t(2c)^{-1}I(|x_t - S_t| \le c)$.
7:

C. Weight Examples

Let us now consider how our algorithm applies to the case of diminishing $1/t$ weights and constant weights.

1) $1/t$ Weights: This choice of weight is used for stationary data as all data values have equal weights in the CDF approximation. For this choice of weight, we will show that D_t is the proportion of deletes in the data by induction. Suppose that this is true for $t-1$, and there are k deletes up to time $t-1$. With the arrival of x_t, by (8), we have

$$D_t = D_{t-1}(1-1/t) = k/(t-1)(t-1)/t = k/t,$$

which is actually the ratio of deletes in the data up to t. If there is deletion at t of an earlier record, then it is easy to see from (7) that $d_{t_0}(t) = 1/t$. Therefore, $D_t = (k+1)/t$ which is again the proportion of deletes. For correction, there is no change in D_t, so D_t remains to be the proportion of deletes. In fact, in this case, for the insertion of x_t, we can easily see that $\hat{\mathcal{F}}_t(x)$ is the weighted sum of $\hat{\mathcal{F}}_{t-1}(x)$ and $I(x \ge x_t)$ with weights $(1-(t-k)^{-1})$ and $(t-k)^{-1}$, respectively. Therefore, the actual weight given to x_t is in fact $1/(t-k)$ not the initial weight $1/t$. This change is due the deletion of k points.

2) Constant Weights: This choice of weight is good for both stationary data or non-stationary data. For non-stationary data, it can track changes in data distributions and works in a similar fashion as the Exponentially Weighted Moving Averages of empirical CDFs. Let $w_t = w$ for a positive w. Let $s_1 < s_2 < \ldots < s_k$ be the set of index of deleted data until time t, where k is the total number of deletes before time t. With the arrival of x_t, it is easy to show that

$$D_t = (1-w)^{t-s_1-1}w + (1-w)^{t-s_2-1}w + \ldots + (1-w)^{t-s_k-1}w.$$

This is because from (7), $d_{t_0}(t) = w(1-w)^{t-t_0+1}$, and also from (8), every insertion will reduce D_t by a factor of $(1-w)$. Notice that D_t is the sum of weights of deleted data, hence the result.

Our probability update equations (11), (12) and (13) are carefully designed to remove the effect of deleted data in the CDF approximations of of $\mathcal{F}_t$. In the case of stationary $\mathcal{F}_t$ where both the inserted data and its deletion/correction mechanism result an equilibrium, (for example, when the deletes occur with a stationary lag distribution), the quantile estimates given by Algorithm 2 with $1/t$ and constant weights should converge around the right quantile. We don't have a formal proof as it is technically very challenging. (The proof for the convergence of the basic SA algorithm (Algorithm I) involves very technical arguments). However, in Section IV, we shall present simulation results that strongly suggest that similar convergence results for Algorithm 1 holds for Algorithm 2.

IV. Simulation Studies and Applications

We evaluate our algorithm using simulation studies with stationary data and a real application with non-stationary data.

A. Simulation Studies

We investigate various issues of the proposed algorithm for quantile estimation: convergence, choice of weights and accuracy. We demonstrate that our estimates converge to the true quantile, similar to the basic algorithm for data with insertions only ([7], [8], [6]). Due to space limitation, we only present results for data with deletions, as both corrections and updates can be viewed as a sort of combination of insertions and deletions.

1) Data Distributions and Weights: For comprehensiveness, we evaluate our algorithms using data generated from four distributions: a uniform distribution on $[0,1]$ $(Unif)$, a normal (Gaussian) distribution's with mean 0 and variance 1 $(norm)$, standard exponential with variance 1(exp), and heavy tailed Pareto distribution with index 2 $(pareto2)$. (A random variable $X, X \geq 1$ with a Pareto distribution with an index α is defined by $P(X > x) = x^{-k}$.) The Pareto distribution with a index 2 has a finite mean, but infinite variance. We consider two choices of weights: diminishing $1/t$ weights and fixed weights.

2) Evaluation Methods: For each simulation setup (specified by data generation and weight choice for quantile estimation), we run the experiment 100 times. For a probability p, let q be its true quantile and $S_t^{(j)}$ be the quantile estimates based on observations up to t in the jth run. The accuracy of the quantiles estimates S_t is measured primary by the empirical median absolute deviation defined by

$$MAD(S_t) = \text{median}|S_t^{(j)} - q|, \tag{14}$$

which is more robust than the mean square error.

3) Results for Stationary Data with Deletions: The mechanism for generating data with deletion is as follows. First, we generate a stream of 2000 data points according to one of the four distribution type specified in Section IV-A1). For each distribution, we also compute a threshold $thre$ which is the nominal 80% quantile of the distribution. If x_t exceeds the threshold, we shall delete the data after a certain lag time l_t, generated from a uniform distribution between 10 and 50. It can be easily seen the stationary data distribution after deletion is the conditional distribution of X_t given that $X_t \leq thre$, irrespective of the specified lag distribution. Let q be the quantile of the stationary distribution $\mathcal{G}$ of inserted data, and q_{new} be the quantile after deletion. This implies $q_{new} = q_{0.8p}$. It is also worthwhile to point out that although 2000 is not a large number, it is enough to evaluate our algorithm as it does not need a lot of data to reach convergence. In fact, the evaluation results hardly change whether there are 2000 or 2 million data points.

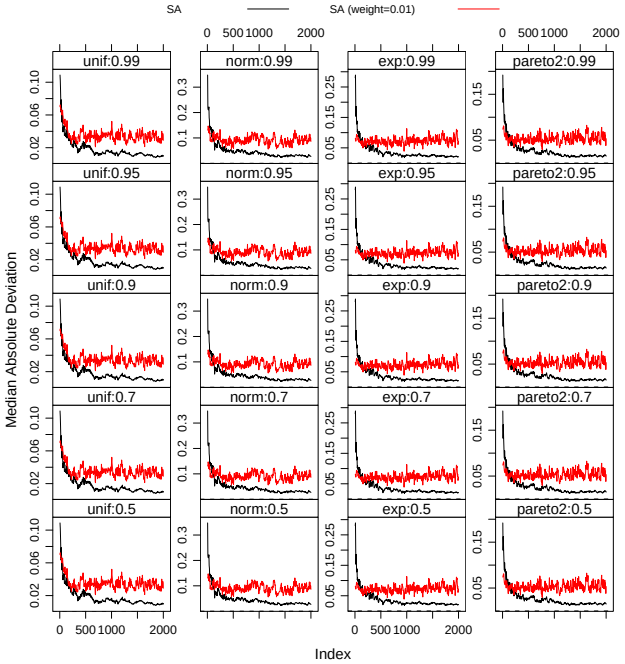

Fig. 1. Median absolute deviation of SA quantile estimates with $1/t$ weights (black) and a constant weight $w = 0.01$ (gray), where the streaming data are generated randomly from a $unif$, $norm$, exp, or $pareto2$ distribution with deletions.

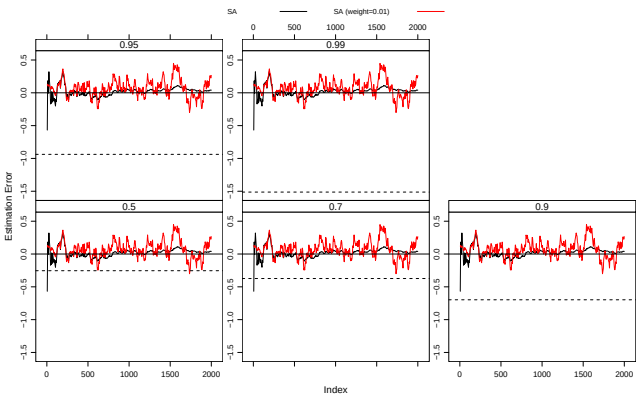

Fig. 2. Quantile estimation error for a sample run in Figure 1 with data generated using $norm$ distribution with deletion ($1/t$ weights: black; constant weights: gray). The dotted line indicates the adjustment of the quantiles from the original data to those of the data after deletion.

The probabilities whose quantiles we are interested in are: $p = 0.5, 0.75, 0.9, 0.99$. Figure 1 shows MAD of SA quantile estimates with data deletions, for two choices of weights, $1/t$ weights (black) and constant weights $w = 0.01$ (gray) for 30

combinations of data distributions and quantile probabilities. Irrespective of the data distribution and the quantile probabilities, the MAD for estimates with $1/t$ weights converges to 0, and the MAD for our SA with constant weight 0.01 stabilizes very early on in iterations. For a closer examination, in Figure 2, we also took one sample run for the normal distribution and plotted the estimation errors for both estimates as a function of iterations. We also added a dotted line that indicates the adjustment of the true quantiles from the original data to that of the data after deletion, i.e., $q_{new} - q$. Since our deletion occurs at the tail portion (20%), the adjustments to the tail quantiles are quite significant.

Results from both Figures 1 and 2 support our theoretical conjecture that our algorithm converges fast for the stationary case in a fashion similar to the well known convergence result for data with insertions only, for both kinds of weights, irrespective of the distributions. We observe similar behavior for stationary data with corrections and updates.

B. An Application

We demonstrate our proposed algorithm using a data example from one real operational 3G wireless network of a US national-wide provider. We collect a one-day packet trace at the home agent of one provider on August 11, 2008. All traffic to and from the Internet of subscribers within a region go through the home agent. We would like to track the quantiles of active flow sizes over time.

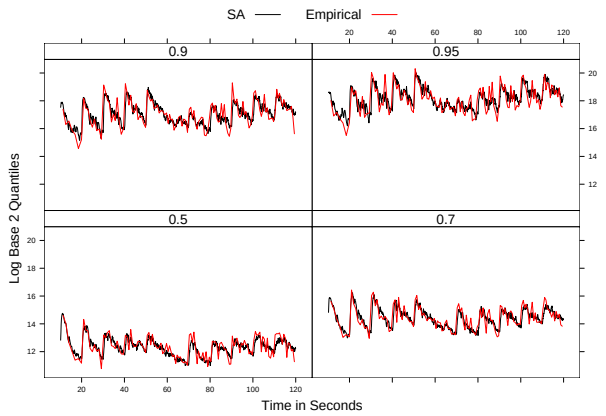

Fig. 3. Quantile estimates for wireless network flow sizes, with flow deletions and updates. Different panel represent estimates for different probabilities 0.5, 0.7, 0.9, 0.95. The black curve represents the estimate based on SA with weight 0.001, and the red curve represents the empirical estimate by tracking the set of most recent 500 active flows.

1) Quantiles of Network Flow Sizes: We insert, delete, and update our flow record in following manner. New flows continuously arrive. Active flow records are updated every 10 seconds. When the TCP FIN packet of a flow is received, the flow is deleted. A flow without receiving TCP FIN is timed out in one minute. This introduces a big variation in terms of flow deletions as some are deleted right away (small flows with FIN), and some are deleted until very late (flows without a FIN or large flows). These and the heavy tailed nature of flow sizes create significant challenges for finding quantiles.

We demonstrate our quantile estimation algorithm using a two minute interval of this flow record data. In this two minute interval, there are about 270K flow records, with 113K new flow insertions, 86K flow deletions, and 71K flow updates. The probabilities whose quantiles are of interest here are: 0.5, 0.7, 0.9, 0.95. To validate our algorithm, at every second, we extract the set of most recent active flows, and compute its empirical quantiles. The flow sizes here have a heavy tailed distribution. For example, the empirical size quantiles at 60sec are: 1.1K, 9.7K, 100K, 258K and 1337K bytes.

Quantile estimates (black) using our algorithm with deletions and updates are shown in log based 2 scale in Figure 3, with a constant weight 0.001. For comparison, we also draw the empirical quantiles computed from the most recent 500 active flows (gray). The window size 500 is chosen to match with the weight 0.001. Since we update active flows in 10 second interval, we see some periodic behaviour every 10 seconds. Despite that, our quantile estimates match very well with the empirical estimates for four probabilities, 0.5, 0.7, 0.9, 0.95.

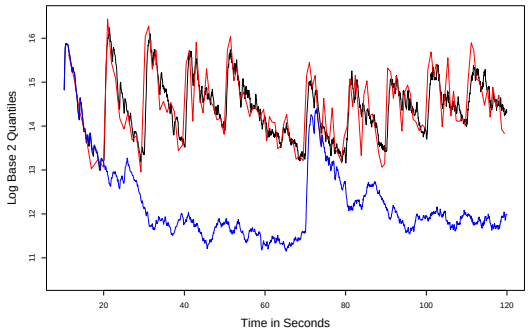

Fig. 4. Quantile estimates for wireless network flow sizes, for probability $p = 0.7$. Both the SA estimates (black) and the closely matched empirical estimates (gray) are shown in Figure 4. However, the SA estimates (bottom gray) by treating flow size updates as erroneous records are far from the truth.

2) Updates vs. Corrections: We illustrate the distinction of updates vs corrections here. In this example, we only have flow updates but not corrections. However, if we treat flow updates as corrections hypothetically, then the quantile estimates are far from the empirical estimates as seen in Figure 4.

REFERENCES

[1] A. Arasu and G. S. Manku. Approximate counts and quantiles over sliding windows. In *PODS '04: Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 286–296, New York, NY, USA, 2004. ACM.

[2] J. Cao, L. E. Li, A. Chen, and T. Bu. Incremental tracking of multiple quantiles for network monitoring in cellular networks. In *The ACM International Workshop on Mobile Internet Through Cellular Networks (MICNET)*, 2009.

[3] J. M. Chambers, D. A. James, D. Lambert, and S. V. Wiel. Monitoring networked applications with incremental quantile estimation. *Statistical Science*, 21:463–475, 2006.

[4] F. Chen, D. Lambert, and J. C. Pinheiro. Incremental quantile estimation for massive tracking. In *KDD '00: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2000.

[5] A. C. Gilbert, S. Guha, P. Indyk, Y. Kotidis, S. Muthukrishnan, and M. J. Strauss. Fast, small-space algorithms for approximate histogram maintenance. In *STOC '02: Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 389–398, 2002.

[6] H. J. Kushner and G. G. Yin. *Stochastic Approximation Algorithms and Applications*. Springer-Verlag, 1997.

[7] H. Robbins and S. Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951.

[8] L. Tierney. A space-efficient recursive procedure for estimating a quantile of an unknown distribution. *SIAM Jounal of Scientific and Statistical Computing*, 4(4), 1983.