# Some Statistical Analysis Algorithms on Data Streams

S. Muthukrishnan
muthu@cs.rutgers.edu

## 1 Introduction

This note presents the streaming framework for message streams and describes results on finding various outliers – most frequent items, deviants — on data streams; the discussion here naturally leads to questions of what are suitable stream analyses for monitoring message streams?

## 2 Data Streams

Data streams are implicit descriptions of various data distributions on appropriate universes. For example, in the context of text/document streams, the universe may be the set of words that appear in the documents pertaining to a topic or a cluster. A function on this universe may be the number of occurrences of each word in the stream so far, or the number of documents that contain each word, etc. In the case of email streams, the universe may be the tuple of to and from email addresses. A function on this universe may be the number of emails sent, or it may be a function of the words in the emails, over the entire stream or only the past day, etc. Yet another example is the time series data where the universe is quanta of time, say, 1 minute, and the function is the number of emails sent from a domain/source during that quanta. Many mix and match of time/words/emails/userids/statistics are possible in defining universe and function. A stream may (typically does) describe multiple data distributions of interest.

Recently, a formal model for managing and mining data streams has emerged: working space, per-item processing time and query processing times are all required to be significantly smaller than (i.e., sublinear in) the data size or the universe size on which the data distribution is described. This is a break from traditional algorithm design where "linear"– space and time–has long being the goal. The motivation for this model is clear: per-item processing has to be fast to match the incoming stream rate, typically workspace that matches the stream rate speed for access is expensive and often limited, and query processing time has to be fast because many of the applications for data streams are near-real time ones involving actionable response.

In this model, one typically monitors the stream and maintains various statistics about the data distributions of interest, often only approximately. The goal is to develop "warning systems" that show significant changes in the stream characteristics so we can flag for analysts' attention. Several results of this genre have appeared recently. Here are two new results which maintain fairly sophisticated statistics. Others analyses of interest include estimating rare items [6], doing trend analysis [7], etc.

### 2.1 Deviants

Deviants are defined as follows. We have a data distribution over $n$ items. Suppose we wish to approximate it using some model, say histograms, i.e., by piecewise constant disjoint functions. Given a bound of at most $k$ deviants, the problem is to

1. determine at most $k$ individual points to be removed, and
2. histogramming the rest using $B$ pieces,

such that the overall approximation is the best in, say, $L_2$ sense. The premise is that deviants are useful outliers for data mining.

In [1] we show an algorithm that uses sublinear resources on a data stream and determines approximate deviants. More formally, in $polylog((B+k), \log n, 1/\epsilon)$ space and in time $polylog((B+k), \log n, 1/\epsilon)$ per data item, we can solve the deviants problem to approximation factor $1+\epsilon$, for a streaming time series. Experiments with real data set of IP flow on lins in the backbone of a large ISP show very good performance for our approximate streaming algorithms.

For the background on deviants, see [2].

### 2.2 Most Frequent Items

The problem here is to monitor a stream of items from universe $1, \ldots, u$ and maintain the set of items that appear the most. There are various results previously known. Chiefly:

1. The problem is to find all items that appear at least $K$ times, in a data stream of size $n$. In [4], authors show present an algorithm to find a superset of items of size at most $1/K$, using space $O(1/K)$ and $O(1)$ time per symbol.

2. The problem is to find the items with $k$ highest frequency. The authors in [3] present a sketching algorithm that gets all items of frequency at least $(1+\epsilon)n_k$ and no items of frequency less than $(1-\epsilon)n_k$, where $n_k$ is the frequency of the $k$-th most frequent item in the data stream. Space used depends on data dsitribution, going from $O(k)$ to $O(n)$ in the worst case, being reasonable for Zipfian distributions.

3. Sampling based algorithms by Gibbons et al. [8]. Guarantees are complicated.

We have the following result. The problem is to find all items that appear at least $n/K$ times for parameter $K$, but items may be inserted and/or deleted. In [5], we show an algorithm that uses time and space $poly(K, \log n, 1/\epsilon)$ and outputs all items that appear at least $(1-\epsilon)n/K$ times with high probability. No other previously known result works in presence of both inserts and deletes. Experiments with real data set from telephone calls that are active at any point show good performance for our algorithms.

## 3 Concluding Remarks

The streaming scenario leads to a natural question. Consider the large matrix of who mailed whom and with what keywords or categories and at what time. We can summarize this matrix over time and perform various analyses: find outliers, find time of day/week/month analyses, wavelet-based or window-based and other trend analyses, etc. In the best case, such analyses may be thought of as broad-stroke indicators of emerging incidents in the stream, eg., is there a significant difference in the distribution between today and yesterday? etc. Is there an application for these methods in homeland security context?

## References

[1] Mining deviants on data streams. S. Muthukrishnan, R. Shah and J. Vitter. Nov. 2002.

[2] Mining deviants in time series databases. H. Jagadish, N. Koudas and S. Muthukrishnan. Proc. VLDB, 1999.

[3] Finding frequent items on data streams. M. Charikar, K. Chen and M. Farach-Colton. ICALP, 2002.

[4] Finding frequent elements in streams and bags. R. Karp, C. Papadimitriou and S. Shenker. Manuscript, 2002.

[5] G. Cormode and S. Muthukrishnan. What is hot and what is not: Dynamically maintaining frequent items. Manuscript, 2002.

[6] M. Datar and S. Muthukrishnan Estimating rarity and similarity in data streams. ESA, 2002.

[7] F. Ergun, S. Muthukrishnan and C. Sahinalp. Sublinear methods for identifying representative trends. Nov, 2002.

[8] P. Gibbons, Y. Matias and V. Poosala. Fast incremental maintenance of approximate histograms VLDB'97.