An Experimental Investigation of Online Aggregation Problems in Data Streaming

September 26, 2009

1 Online Data Aggregates

Data Streams [9] have been an important research area for the last ten years, following the explosion of the Internet (and distributed network computing in general) where vast amounts of data arrive rapidly and have to be processed *efficiently*, i.e. in almost-real time and respecting the natural space limitations of the network hardware. We are primarily interested in answering statistical queries regarding the distribution of the elements of the data stream and traditionally interest has been shifted towards space-complexity efficiency of streaming algorithms (e.g. [1]).

However, data streaming is an intrinsically dynamic-over-time computational environment and thus we believe it is essential to design *online* streaming algorithms and deploy *competitive analysis* [4] as an efficiency-measuring framework, comparing their performance to that of an optimal offline adversary who knows the entire input stream in advance. Furthermore, in such a framework it seems theoretically, as well as practically, preferable to average-out our queries throughout the length of the stream, i.e. compute statistical *aggregates* [3] of the underlying stream.

2 Aggregate Max in windowed streaming

In windowed streaming, at every time point we want to answer statistics queries regarding only the last $w \in \mathbb{N}$ observations (where w is the window's length), not being interested in the rest (older part) of the stream. Computing the exact maximum over a sliding window is feasible only in linear space [6] which is generally unacceptable for a data stream algorithm, so we must resolve to approximation. In the aggregate max problem we are allowed to maintain only $k \ll w$ items in memory, wanting the best (maximum) among them to be as close to the maximum value of the entire current window, averaging out our performance throughout all sliding windows up to now.

2.1 Experimental evaluation of known results

In [3] a natural deterministic algorithm having a competitive ratio of $k/(k-1)$ is given. However, due to the *worst-case* approach of competitive analysis, computer experimentation could give useful insight to the average or "real-life" performance of that algorithm in various input distributions, as well as point out "cruel" instances. Furthermore, in the same paper the authors prove a deterministic competitive ratio of $1 + \Theta(1/k)$ for the aggregate max problem, with the constants residing inside this Θ-notation varying substantially, leaving space for computer-aided experiments to help us close this gap.

A special, interesting case of the aggregate max problem, not addressed in [3] is that where we can only store a single item at every given time. Implementing an algorithm in this case becomes much more straightforward and we hope that experiments in various families of input streams will point us to the right online algorithm for this special case, as well as shed further light to the quest of bridging the gap at the aforementioned competitive ratio. Finally, a variation of the aggregate max problem, which would surely benefit from extensive computer-aided calculations, would be that of considering the order statistics of the items rather than their exact values.

3 Other Aggregate Statistics and Model Extensions

To our knowledge, [3] is the only work in competitive analysis of data stream statistics, leaving space for many other statistical questions that have been already extensively studied in the "traditional" framework of space-complexity efficiency. For example, a well studied problem is that of maintaining the frequent items of a given stream or its *heavy-hitters* [5]. This is a very important question in actual network traffic applications and some of the well-known space efficient algorithms [7, 8] can be easily adopted to our online framework. Experiments can be very useful here since, unfortunately, we know very little regarding the performance of online algorithms with regard to other statistics than the aggregate max and thus, practical indications may turn out to be vital in devising new algorithms and testing the performance of existing ones in our new, online setting. Other statistics might be finding the number of distinct elements that have appeared in the stream [2] or computing its frequency moments [1]. Of course, all the above statistics are not necessarily to be applied to a sliding window model but they can also be studied in a very general *aging* framework, where we assign to each past element a weight corresponding to its importance, depending on the time that has elapsed since its appearence.

References

[1] N. Alon, Y. Matias, and M. Szegedy. The Space Complexity of Approximating the Frequency Moments. *Journal of Computer and System Sciences*, 58(1):137–147, 1999.

[2] Ziv Bar-Yossef, T. S. Jayram, Ravi Kumar, D. Sivakumar, and Luca Trevisan. Counting distinct elements in a data stream. In *RANDOM '02: Proceedings of the 6th International Workshop on Randomization and Approximation Techniques*, pages 1–10, 2002.

[3] L. Becchetti and E. Koutsoupias. Competitive analysis of aggregate max in windowed streaming. In *Proceedings of the 36th International Colloquium on Automata, Languages and Programming (ICALP): Part I*, pages 156–170. Springer, 2009.

[4] Allan Borodin and Ran El-Yaniv. *Online Computation and Competitive Analysis*. Cambridge University Press, 1998.

[5] Graham Cormode and Marios Hadjieleftheriou. Finding frequent items in data streams. *Proc. VLDB Endow.*, 1(2):1530–1541, 2008.

[6] Mayur Datar, Aristides Gionis, Piotr Indyk, and Rajeev Motwani. Maintaining stream statistics over sliding windows. In *SIAM Journal on Computing*, pages 635–644, 2002.

[7] Erik D. Demaine, Ro López-ortiz, and J. Ian Munro. Frequency estimation of internet packet streams with limited space. In *Proceedings of the 10th Annual European Symposium on Algorithms (ESA)*, pages 348–360. Springer-Verlag, 2002.

[8] A. Metwally, D. Agrawal, and AE Abbadi. Efficient computation of frequent and top-k elements in data streams. In *Proceedings of the 10th ICDT International Conference on Database Theory*, pages 398–412. Springer, 2005.

[9] S. Muthukrishnan. *Data streams: Algorithms and applications*. Now Publishers Inc, 2005.