OPEN PROBLEMS IN DATA STREAMS AND RELATED TOPICS
IITK WORKSHOP ON ALGORITHMS FOR DATA STREAMS '06

Abstract. This document contains a list of open problems and research directions that have been suggested by participants at the IITK Workshop on Algorithms for Data Streams. Many of the questions were discussed at the workshop or were posed during presentations. Further details, including videos of discussion sections, can be found at

`http://www.cse.iitk.ac.in/users/sganguly/workshop.html` .

Please send any comments/corrections regarding this document to `andrewm@ucsd.edu`.

Workshop Speakers.
Pankaj Agarwal
Surender Baswana
Amit Chakarabarti
Graham Cormode
Sudipto Guha
Piotr Indyk
T. S. Jayram
Ravi Kannan
Sampath Kannan
Ravi Kumar
Stefano Leonardi
Yossi Matias
Michael Mahoney
Andrew McGregor
S. Muthukrishnan
Rajeev Raman
Nicole Schweikardt
D. Sivakumar
Christian Sohler
Divesh Srivastava
Martin Strauss
Subhash Suri
Srikanta Tirthapura

Question 1: Fast L_1 Difference (Graham Cormode)

In data streaming, the focus is often on the space complexity of solving particular problems. It turns out that, in practice, when processing massive streams online, time efficiency is just as important, if not more so, than space usage. For many aggregates, such as L_2, F_0, quantiles, heavy hitters and so on, not only are the best known solutions optimal or nearly optimal in space, they also turn out to be very time efficient. Indeed, for many problems it seems that some solutions are known which require very little time to process each update in the stream. One notable exception is the problem of computing the L_1 difference between two vectors specified by streams. The well-known way to do this involves using 1-stable distributions (the Cauchy distribution), and tracking the inner product of each vector with a pseudo-random vector whose entries are each drawn from a Cauchy distribution. However, to get sufficient accuracy requires tracking a large number of independent inner-products, which means each update can be quite costly.

The main open question therefore is to study the time complexity of L_1 difference computations. Two possible directions suggest themselves:

(1) The algorithms of Indyk and Woodruff [IW05], and simplifications by Bhuvanagiri et al. [BGKS06] give improved bounds for F_k computations, $k > 2$, based on estimating large frequencies individually and removing; this approach has been extended to quantities such as entropy [BG06]. Can it also apply to L_1?

(2) Recent work [LHC06] has studied sparse random projections for L_2. Follow up work [Li06] has extended this to sparse projections using stable distributions. What time bounds does this imply for (ϵ, δ)-approximation of L_1 distance?

A more general open question arises. So far, there has been considerable success in proving space lower bounds for data stream computations using tools from communication complexity and cell probe model. Is it possible to give non-trivial time lower bounds for update cost (either worst case or amortized) on data streams? Note that the difference between an O(1) and $O(\epsilon^{-2} \log^3 n)$ algorithm for processing each update in a stream translates into the difference between an $O(n)$ and $O(n\epsilon^{-2} \log^3 n)$ algorithm, which might be considered only a small difference in traditional algorithms.

Question 2: Quantiles (Graham Cormode)

The problem of tracking the quantiles (median and generalizations thereof) of a distribution produced by a stream has attracted significant study over the last decade [MRL98, MRL99, GK01, GKMS02, CM05a, SBAS04, GM06]. For deterministic algorithms on insert only streams, two algorithms obtain the best (and incomparable) space bounds: $O(\epsilon^{-1} \log \epsilon N)$ words [GK01] and $O(\epsilon^{-1} \log U)$ words [SBAS04], where U is the size of the domain from which the input is drawn.

The Greenwald-Khanna algorithm (GK) is simple to implement, and works on streams of items drawn from arbitrary domains. However, the analysis is rather involved; moreover, attempts to modify the analysis for different situations (say, weighted input items, merging summaries together, giving different guarantees to different ranges etc.) lead to heuristics at best, which may no longer have strict guarantees and known bad cases. The q-digest algorithm [SBAS04] is much simpler to analyze and more amenable to variations, meaning that several generalizations and alternatives have been proposed [HSST04, CKMS06]. However, it carries with it a factor of $\log U$, meaning that the universe has to be known, making it impractical for tracking quantiles over streams of floating point values, or strings.

This leads to some interlinked open questions:

(1) What is the optimal space bound for an algorithm to compute quantiles of a data stream? Is $O(\epsilon^{-1})$ words achievable?

(2) Can the GK algorithm, or a variation thereof, submit to a simpler analysis which will allow generalizations of the algorithm to be more easily proposed and studied?

Question 3: L_∞ estimation (Graham Cormode)

One of the earliest results shown in data streaming is that approximating L_∞ of a stream of values requires space proportional to the dimensionality of the stream. The hard case used to prove this is when most items in the stream have frequency of occurrence 1, and approximating L_∞ is equivalent to testing whether any item has frequency two or higher. However, a variation of this problem is routinely studied under the name "heavy hitters." Here, the lower bound is avoided by asking to find all items whose frequencies are greater than some fixed fraction ϕ of the total stream length, and tolerating approximation error ϵ. Bounds are then provided which are polynomial in $(1/\phi)$ or $(1/\epsilon)$. A side effect of these algorithms is to estimate L_∞ of the stream with error proportional to ϵ times the L_1 or L_2 norm of the stream. Let the stream consist of items specified in $\log m$ bits. For insert only streams, the best space bound is $O(\epsilon^{-1}(\log m + \log L_1))$ [MG82, MAA05], for computing on the difference between two streams the bounds are $O(\epsilon^{-1} \log m(\log m + \log L_1))$ [CM05c]. These algorithms approximate the L_∞ distance in the sense above, but additionally identify a set of items which contribute significantly to the distance.

The open question is whether it is possible to approximate L_∞ with additive error in terms of ϵ times L_1 or L_2 with less space. In particular, is it possible to reduce the dependency on m, since this is not needed in the output? One possible direction is to analyze data structures such as the Count-Min sketch, from which items frequencies can be estimated and in which m does not occur in the (word) space complexity [CM05a].[1]

Question 4: Deterministic Summary Structures (Sumit Ganguly)

Given a stream of elements of the form (i, δ) where $i \in [n]$ and $\delta \in \{-1, 1\}$ define the frequency of an element to be $f_i = \sum_{(i,\delta)} \delta$. We wish to find estimates $\hat{f}_i$ for each f_i such that

$$|\hat{f}_i - f_i| \leq \epsilon L_1$$

where $L_1 = \sum_i |f_i|$. The Count-Min algorithm is a randomized $O(\epsilon^{-1} \log(mn) \log \delta^{-1})$-space algorithm that returns such estimates with probability $1 - \delta$ [CM05a]. This is nearly optimal as the space lower bound is $O(\epsilon^{-1} \log(m) \log \epsilon n)$ [GM07a].

However, in practice it is desirable to have deterministic algorithms rather than randomized algorithms. Using a deterministic collection of primes [Mut06a], [GM07a] devised a deterministic $O(\phi^{-2}\epsilon^{-1} \log^2(mn))$-space algorithm that returned all items i with $|f_i| \geq \phi L_1$ and no j satisfying $|f_j| \leq (1 - \epsilon)\phi L_1$. While this algorithm has the advantage of being deterministic, it uses more space than the Count-Min algorithm. Does there exist a deterministic algorithm that uses the same amount of space as Count-Min? Such an algorithm would lead to space-efficient algorithms for a range of problems including hierarchical heavy hitters, estimating inner product sizes, approximately optimal B-bucket histograms etc. Unfortunately, we conjecture that no such algorithm exists. Either an algorithm or lower bound would be very interesting.

Question 5: Characterizing Sketchable Distances (Sudipto Guha & Piotr Indyk)

Some of the early successes in developing algorithms for the data stream model related to estimating L_p norms [FKSV02, Ind00, AMS99] and the "Hamming norm" L_0 [CDIM03]. What other distances, or more generally "measures of dissimilarity," can be approximated in the data stream

[1] Formally, $\log m$ does affect the bit space complexity in two places: the data structure consists of $O(\log 1/\delta)$ hash functions whose specification requires $O(\log m)$ bits; and $O(\epsilon^{-1} \log 1/\delta)$ counters which in the worst case may count to the L_1 norm of the whole stream – this may perhaps be addressed by using approximate counters.

model? Do all sketchable distances essentially arise as norms, specially, if deletions are allowed? Note that the set similarity distance (symmetric difference over union) can be estimated in the streaming model in the absence of deletions [BCFM00].

Recent work provides some preliminary results [GIM07]. Let $f = (f_1, \ldots, f_n)$ and $g = (g_1, \ldots, g_n)$ be two frequency vectors defined by a stream in the usual way. Consider a distance $d(f, g) = \sum_i \phi(f_i, g_i)$ where $\phi : \mathbb{N} \times \mathbb{N} \to \mathbb{R}^+$ and $\phi(x, x) = 0$. If there exist $a, b, c \in \mathbb{N}$ such that

$$\max\left(\frac{\phi(a+c,a)}{\phi(b+c,b)}, \frac{\phi(a,a+c)}{\phi(b,b+c)}\right) > \alpha^2$$

then it can be shown that any one-pass α-approximation of $d(f, g)$ requires $\Omega(n)$ space where the stream defining f and g has length $O(n(a + b + c))$. Similar results hold for multiple-pass algorithms and for probabilistic divergences of the form $d(f, g) = \sum_i \phi(p_i, q_i)$ where $p_i = f_i/L_1(f)$ and $q_i = g_i/L_1(g)$. These results suggest that for a distance d to be sketchable, $d(x, y)$ needs to be some function of $x - y$. In particular, they show that multiplicative approximation of all f-divergences and Bregman divergences, such as Kullback-Leibler and Hellinger, requires $\Omega(n)$ space with L_1 and L_2^2 being notable exceptions.

Question 6: Filtering Irrelevant Data (Sariel Har-Peled)

For many problems most of the stream is irrelevant and a good use of a streaming algorithm could be to filter out the irrelevant parts of the stream such that the data left is small enough to be processed by an I/O efficient algorithm. How effective can a small-space algorithm be at such filtering for a given problem? An alternative idea that addresses similar issues is to allow a data stream algorithm to delete and annotate the stream and take multiple passes as in [DFR06]. If the deletion of irrelevant elements was a large component of the algorithm then it would not make sense to measure the total number of passes taken by the algorithm but, rather, the total number of elements processed.

Question 7: Estimating Earth-Mover Distance (Piotr Indyk)

Consider a stream of red points R and blue points B from a 2-dimensional grid $[\Delta]^2$, in an arbitrary order. We assume $|R| = |B| = n$. The Earth-Mover Distance (EMD) between R and B is the value of the min-cost matching between R and B, i.e.,

$$\mathrm{EMD}(R, B) = \min_{\pi: R \to B} \sum_{p \in R} \|p - \pi(p)\|$$

where π is a one-to-one mapping, and $\|\cdot\|$ is (say) the L_1 norm.

What are the space vs. approximation tradeoffs achievable by streaming algorithms for this problem? In particular, is there an $O(1)$-approximation algorithm using $(\log n + \log \Delta)^{O(1)}$ space?

It is known that that there is an $O(\log \Delta)$-approximation algorithm using that much space [Ind04]. That algorithm proceeds essentially by embedding EMD into L_1 [Cha02, IT03]. However, any such embedding must incur at least $\Omega(\sqrt{\log \Delta})$ distortion [NS06]. So one would need to do something else to get $O(1)$-approximation.

Question 8: Mixed Norms (Piotr Indyk)

For any vector x, let $\|x\|_0$ be a norm-like function computing the number of non-zero elements in x. Consider the following norm-like function $\|\cdot\|_{2,0}$ over $n \times n$ matrices $A = [a_1 \ldots a_n]$:

$$\|A\|_{2,0} = \left(\sum_{i=1}^{n} (\|a_i\|_0)^2\right)^{1/2}$$

Assume we are given a stream of m updates (i, j, δ) to A, interpreted as $A[i, j] := A[i, j] + \delta$, starting from $A = 0$. What is the smallest space needed by a streaming algorithm estimating $\|A\|_{2,0}$ up to a factor of $1 \pm \epsilon$? An upper bound of $O(\text{poly}(\epsilon^{-1})\sqrt{n}\,\text{polylog}(n))$ is known as long as $A \geq 0$ [CM05b]. There are no non-trivial lower bounds known.

Question 9: OSPF Routing (Sampath Kannan)

Open-Shortest-Path-First (OSPF) routing is an intra-domain routing protocol where each link of a network is assigned a weight and each packet is forwarded along the shortest path given these weights (e.g. [KR04].) Initially, the weight of a link is the reciprocal of the bandwidth of the link. However, as a link becomes congested, it would make sense to discourage the use of the link by increasing the weight of the link. We are interested in setting these weights such that the flows in the network are routed "optimally" for some appropriate notion of optimality. At present this is done locally at each link. Unfortunately, this often causes oscillatory behavior. Is there a distributed-stream approach to this problem? In particular, traffic is monitored at each router subject to the usual streaming constraints and a limited amount of communication is permitted between the routers. Given these limitations, is it possible to implement a better, more "global" solution.

It should be mentioned that for many notions of optimality, achieving optimal routing is NP-hard even when the traffic matrix is known and weights are set by some central authority. Consequently, it would be necessary to focus on notions of optimality that are at least achievable in such an idealized setting. Alternatively, one could ask which heuristics can be implemented in the distributed-stream setting. Comparisons between different solutions could be in terms of the rate of convergence to stable solutions.

Question 10: Multi-Round Communication of Gap-Hamdist (Ravi Kumar)

Consider the communication problem Gap-Hamdist: Alice and Bob are given length n binary strings x and y such that either the Hamming distance $\Delta(x, y) \leq n/2$ or $\Delta(x, y) \geq n/2 + \sqrt{n}$. The one-way communication complexity of Gap-Hamdist is known to be $\Omega(n)$ [IW03, Woo04]. Recently, a simpler proof was discovered using a reduction from Index [JKS07]. Is the multi-round communication complexity also $\Omega(n)$? There is a $\Omega(\sqrt{n})$ lower-bound from a reduction from Set-Disjointness but we conjecture that the lower-bound is actually $\Omega(n)$.

If the conjecture is true then it would imply stronger multiple-pass lower bounds for estimating F_0 [IW03, Woo04, BYJK$^+$02] and entropy [BG06, CCM07]. Alternatively, if the conjecture is not true then it would be interesting to see if better multi-pass algorithms exist for F_0 and entropy.

Question 11: Counting Triangles (Stefano Leonardi)

Given a stream in which edges are inserted and deleted to/from an unweighted, undirected graph, how well can we count triangles and other sub-graphs? Most of the previous work has focussed on the case of insertions [BYKS02, JG05, BFL$^+$06] although it appears that one of the algorithms in [JG05] may work when edges can be deleted. Is it possible to match the insert-only bounds when edges are inserted and deleted?

Question 12: Deterministic CUR-Type Decompositions. (Michael Mahoney)

A CUR-decomposition of A expresses A as a product of three matrices, C, U, and R, where C consists of a small number of actual columns of A, R consists of a small number of actual rows of A, and U is a small, carefully constructed matrix that guarantees that the product CUR is "close" to A. Recent work [DMM06a, DMM06b, DMM06c] proved the existence and provided

efficient randomized algorithms for CUR decompositions that are nearly as good as the best rank-k approximation to A that is obtained by truncating the SVD. Hence, the columns of A that are included in C, as well as the rows of A that are included in R, can be used in place of the eigencolumns and eigenrows, with the added benefit of improved interpretability in terms of the original data. Note the structural simplicity of a CUR matrix decomposition:

$$\underbrace{\begin{pmatrix} \\ A \\ \\ \end{pmatrix}}_{m\times n} \approx \underbrace{\begin{pmatrix} \\ C \\ \\ \end{pmatrix}}_{m\times c} \underbrace{\begin{pmatrix} \\ U \\ \\ \end{pmatrix}}_{c\times r} \underbrace{\begin{pmatrix} \\ R \\ \\ \end{pmatrix}}_{r\times n}.$$

We briefly expand on the latter point. In many cases, an important step in data analysis is to construct a compressed representation of A that may be easier to analyze and interpret. The most common such representation is obtained by truncating the SVD at some number $k \ll \min\{m, n\}$ terms, in large part because this provides the "best" rank-k approximation to A when measured with respect to any unitarily invariant matrix norm. Unfortunately, the basis vectors (the so-called eigencolumns and eigenrows) provided by this approximation (and with respect to which every column and row of the original data matrix is expressed) are notoriously difficult to interpret in terms of the underlying data and processes generating that data. Gould, in the "Mismeasure of Man" [Gou96], provides examples where such reification of the singular vectors (or principal components or "factors") resulted in social policy with potentially devastating consequences for large groups. For example, the vector [(1/2) age - $(1/\sqrt{2})$ height + (1/2) income], being one of the significant uncorrelated "factors" from a dataset of people's features, is not particularly informative. From an analyst's point of view, it would be highly preferable to have a low-rank approximation that is nearly as good as that provided by the SVD but that is expressed in terms of a small number of *actual columns* and/or *actual rows* of a matrix, rather than linear combinations of those columns and rows. Our CUR matrix decomposition is a direct formulation of this problem. For example, the CUR matrix decomposition was recently applied to hyperspectrally-resolved medical imaging data [MMD06]. In this application, a column corresponds to an image at a single physical frequency and a row corresponds to a single spectrally-resolved pixel, and it was shown that data reconstruction and classification tasks can be performed with little loss in quality even after substantial data compression.

The main existing result for CUR matrix decompositions is the following.

Theorem (Drineas et al. [DMM06c]). *Given a matrix $A \in \mathbb{R}^{m\times n}$ and an integer $k \ll \min\{m, n\}$, there exist randomized algorithms such that if $c = O(\epsilon^{-2}k \log k \log(1/\delta))$ columns of A (in expectation) are chosen to construct C, and then $r = O(\epsilon^{-2}c \log c \log(1/\delta))$ rows of A (in expectation) are chosen to construct R, then with probability at least $1 - \delta$,*

$$\|A - CUR\|_F \leq (1 + \epsilon)\|A - A_k\|_F.$$

Here, the matrix U is a weighted Moore-Penrose inverse of the intersection between C and R, and A_k is the best rank-k approximation to A. The randomized algorithm runs in time $O(SVD(A_k))$, which is the time required to compute the best rank-k approximation to the SVD [GL89].

Many important questions remain open within the context of CUR-type decompositions. The most important one is to devise *deterministic* algorithms. Whereas, from a theoretical viewpoint, the randomized algorithms are satisfactory, deterministic algorithms would be much preferable. Results of Gu and Eisenstat [GE96] and Stewart [Ste99, Ste04] may help towards this goal. Also relevant is work by Goreinov, Tyrtyshnikov, and Zamarashkin [GTZ97, GT01] that was motivated

by applications such as scattering, in which large coefficient matrices have blocks that can be easily approximated by low-rank matrices. They showed that if the matrix A is approximated by a rank-k matrix to within an accuracy ϵ then *there exists* a choice of k columns and k rows, i.e., C and R, and a low-dimensional $k \times k$ matrix U constructed from the elements of C and R, such that $A \approx CUR$ in the sense that $\|A - CUR\|_2 \leq \epsilon f(m,n,k)$, where $f(m,n,k) = 1 + 2\sqrt{km} + 2\sqrt{kn}$. In [GTZ97], the choice for these matrices is related to the problem of determining the minimum singular value σ_k of $k \times k$ submatrices of $n \times k$ orthogonal matrices. In addition, in [GT01] the choice for C and R is interpreted in terms of the maximum volume concept from interpolation theory, in the sense that columns and rows should be chosen such that their intersection W defines a parallelepiped of maximum volume among all $k \times k$ submatrices of A.

A second research topic is to improve the error bounds of previous results, and improve the dependency of the number of sampled columns and rows on k and ϵ. Again, the aforementioned results from the numerical linear-algebra community will serve as starting points.

Question 13: Effects of Subsampling (Yossi Matias)

When processing very fast streams, it is not feasible to run a streaming algorithm on the entire stream, even one that can process each element in $O(1)$ time. Rather it is necessary to sample from the stream and to process the sub-stream using a streaming algorithm. For standard problems such as estimating F_0, how does the sub-sampling affect that the accuracy of the streaming algorithms? How should the sampling rate and the per-element time-complexity of a streaming algorithm be traded-off to achieve optimal results?

Another way to formalize this question, suggested by Muthukrishnan, is in terms of what part of the stream to skip and which to stream. A formal definition of the model and algorithms for estimating F_2 and others can be found in [BMMY07].

Question 14: Graph Distances (Andrew McGregor)

Given a stream of edges defining a graph G, how well can we estimate $d_G(u,v)$, the length of the shortest path between two nodes u and v? Progress that has been made on this problem is based on constructing *spanners* [FKM$^+$05a, FKM$^+$05b, EZ06, Bas06, Elk06] where subgraph H of G is an (α,β)-spanner for G if,

$$\forall x, y \in V,\ d_G(x,y) \leq d_H(x,y) \leq \alpha \cdot d_G(x,y) + \beta\ .$$

Clearly, an (α,β)-spanner gives an $\alpha + \beta/d_G(u,v)$ approximation to $d_G(u,v)$. Since a spanner is constructed independently of u and v it is perhaps surprising that this approach gives nearly optimal results for approximating $d_G(u,v)$ in a single pass [FKM$^+$05a]. It is unclear whether there is a better approach for multiple pass algorithms. Clearly, $d_G(u,v)$ can be computed exactly in $d_G(u,v)$ passes but for $d_G(u,v)$ large this is infeasible. Can we do better? For example, how well can $d_G(u,v)$ be approximated in $O(\log n)$ passes? What if the edges arrived in random order?

Question 15: Semi-Random Streams (Andrew McGregor)

What is the right notion of "semi-random" order streams? While streams are normally assumed to be ordered by some omnipotent adversary, there is a growing body of work in which the order of the stream is assumed to be chosen uniformly from the set of all possible orderings [MP80, DLOM02, GMV06, GM06, GM07b, GM07c]. This "full-random" ordering is interesting as a form of average-case analysis or in a stochastic setting in which each element of the stream is an independent sample drawn from some fixed unknown distribution [GM07c]. More generally, it would be interesting to develop algorithms whose performance degraded smoothly as the stream ordering became "less-random." This begs the question of what it means to be "semi-random."

The following notions were recently proposed [GM06]:

(1) *t-Bounded-Adversary-Random:* A *t-bounded adversary* is a space-bounded adversary that can delay at most t elements at a time, i.e., can transform a stream $\langle x_1, \ldots, x_m \rangle$ into a stream of the form $\langle x_{\sigma(1)}, \ldots x_{\sigma(m)} \rangle$ if the permutation σ satisfies,

$$\forall i \in [m],\ |\{j \in [m] : j < i \text{ and } \sigma(i) < \sigma(j)\}| \leq t\ .$$

The order of a stream is t-bounded-adversary-random if it is generated by a t-bounded adversary acting on a stream whose order is random.

(2) ϵ-*Generated-Random:* Consider a set of elements $\{x_1, \ldots, x_m\}$. Then a permutation σ defines a stream $\langle x_{\sigma(1)}, \ldots, x_{\sigma(m)} \rangle$. We say the ordering of this stream is ϵ-*Generated Random* if σ is chosen according to some distribution ν such that $\|\mu - \nu\|_1 \leq \epsilon$ where μ is the uniform distribution over all possible orderings.

How do these notions relate to each other? Can we develop algorithms whose performance degrades smoothly as the stream ordering becomes "less-random" using either definition? For a given application, which notion is more appropriate? Are there other useful definitions for semi-random order?

Question 16: Graph Matchings (Andrew McGregor)

Given a weighted graph with n nodes and m edges, the maximum weighted matching (MWM) problem is to find the set of edges of maximum weight such that no two edges share an end-point. MWM is a classic graph problem and exact polynomial solutions are known [Edm65, Gab90, HK73, MV80]. The fastest of these algorithms solves the maximum weighted matching problem with running time $O(nm + n^2 \log n)$. For massive graphs this is still too much and there has been recent work on finding faster approximate algorithms. For the unweighted problem, a linear-time approximation-scheme is known [KS95]. The best general result is a linear time $(2/3 - \epsilon)$-approximation [DH03, PS04].

Algorithms in the data stream model were presented in [McG05]. These include $O(n \log n)$-space, $O_\epsilon(1)$-pass algorithms that return a $(1 - \epsilon)$-approximation in the unweighted case and a $(1/2 - \epsilon)$-approximation in the weighted case. Both are also linear time algorithm in the RAM model. The algorithms for unweighted matching are based on finding augmenting paths[2] for an existing matching. Many of the ideas used for finding augmenting paths in the unweighted case carry over to the weighted case. However, it seems that the intrinsic difficulty in achieving a $(1-\epsilon)$-approximation in the weighted case is that there may be augmenting cycles[3]. It seems hard to find augmenting cycles in the streaming model. Is there a lower-bound or does there exist an $O_\epsilon(1)$-pass $O(n \log n)$-space algorithm that returns an $(1 - \epsilon)$-approximation for MWM. In the RAM model, does there exist a linear time $(1 - \epsilon)$-approximation for MWM?

Question 17: The Massive, Unordered, Distributed-data Model (S. Muthkrishnan)

The Massive, Unordered, Distributed-data (MUD) model was recently introduced by Feldman et al. [FMS+06] as an abstraction of part of the infrastructure used at Google. It is related to the MapReduce framework presented in [DG04]. In the multi-round, multi-key MUD model, n data records are distributed arbitrarily between M machines. Each machine maps each record to (key, value) pairs. All pairs corresponding to the same key are then "reduced" to a single record. This reduction is performed by an $O(\text{polylog}\, n)$-space streaming computation. The process repeats for a total of l rounds.

[2] An augmenting path is a simple paths of odd length such that every second edge in the current matching.

[3] An augmenting cycle is an even length cycles such that every second edge is in the matching and swapping the matched edges for the unmatched edges will increase the weight of the matching.

The model is very powerful and it was proven that any EREW-PRAM algorithm can be simulated in the multi-round, multi-key MUD model if the number of keys and rounds is sufficiently large [FMS+06]. In practice we are primarily interested in computing with a small number of keys and rounds. What can be computed given k keys and l rounds?

Question 18: Finite Cursor Machines (Nicole Schweikardt)

The Finite Cursor Machine (FCM) model is an abstract model for database query processing based on abstract state machines [GGL+07]. This model has the following fixed structure:

(1) A *background structure* $\mathcal{U}$ that consists of an infinite set U of potential database entries, and some functions and predicates on U (e.g., $\mathcal{U} = (\mathbb{N}, <, +, \times)$)
(2) A *database schema* σ that consists of a finite number of relation symbols $R_1, \ldots, R_t$ of arities $r_1, \ldots, r_t$.

The input of a problem in this model is a database D of schema σ where D is a collection of t tables $R_1^D, \ldots, R_t^D$ and each table R_i^D is a list of elements from U^{r_i}. On every input table, the FCM has a fixed number of cursors which can only move from top to bottom. Apart from this, the FCM also has an internal memory consisting of a constant number of "modes" (comparable to the states of a Turing machine) and a register for storing up to $o(n)$ many bits where n is the total number of tuples in D.

Is there a Boolean query from Relational Algebra (or, equivalently, a sentence of first-order logic), that cannot be computed by any composition of FCMs and sorting operations? We conjecture that there is no such Boolean query.

Question 19: Sketching vs. Streaming (D. Sivakumar)

Show that any symmetric function that admits a good streaming algorithm also admits a sketching algorithm. In terms of communication complexity, consider trying to evaluate a symmetric function $f(x, y)$ in each of the following models:

(1) *One-Way Communication:* The player knowing x sends a single message to the player knowing y who then has to compute $f(x, y)$.
(2) *Simultaneous Communication:* Both players send a message simultaneously to a third party who then has to compute $f(x, y)$.

Obviously any function that can be evaluated with B bits of communication in the simultaneous model can be evaluated in the one-way model with B bits of communication. Are there natural functions that require significantly more communication in the simultaneous model than in the one-way model? It is known that any total, permutation-invariant function that can be computed in the one-way model can be computed in the simultaneous model. See [FMS+06] for further details.

Question 20: Relations between Streaming Models (Christian Sohler)

There are many different models for data streaming. For example, in geometry we have the insertion-only model, insertion/deletion model, and the sliding window model. In the insertion-only model we are given a stream of points $p_1, \ldots, p_n$. In the insertion/deletion model the stream consists of Insert(p) and Delete(p) operations and is assumed to be valid in the sense that no point is deleted that has not been previously inserted and no point is inserted twice. In the sliding window model we get a stream $p_1, \ldots, p_m$ but we are only interested in the n most recent points.

How do these models relate to each other? Obviously, any algorithm for the insertion/deletion model is also an algorithm in the insertion-only model. Under which assumptions is the opposite true as well? Is there any relation between the insertion/deletion model and the sliding window model? The models are not equivalent since one can obtain an exact algorithm for the sum of points (centroid) under the insertion/deletion model but only a $(1+\epsilon)$-approximation in the sliding

window model. Is it possible to prove that (under reasonable assumptions) these two models are equivalent within a certain approximation factor, i.e., if there is an α-approximation algorithm in one model then there is a $(c\alpha)$-approximation algorithm in the other model?

Since the above questions are quite general and may be difficult to answer, here is one that may be easier to solve: Can you prove that the reset model [HMR04] is equivalent to uniform sampling?

In the reset model we have a stream of updates (i, p) telling that the new position of point number i is p. The conjecture is that the reset model is equivalent to uniform random sampling. Since one direction is immediate, one has to prove that any algorithm in the reset model can be turned into a streaming algorithm that initially chooses a set of points (indices) uniformly at random and tracks the positions of these points. Then the algorithm computes its output based on the position of these points.

Question 21: Deterministic Heavy-Hitters & Fast Matrix Algs (Martin Strauss)

An important ingredient in many recent algorithms for heavy hitters is the Restricted Isometry Property, defined in [CRT06] and, equivalently, in [Don06]. A matrix Φ with d columns has the m-RIP if any submatrix Φ_0 of m columns has low-distortion, i.e., for all x, we have $\|x\|_2 \leq \|\Phi_0 x\|_2 \leq 2\|x\|_2$ (after appropriate normalization). The identity matrix has this property; we want to minimize the number of rows in matrices with the m-RIP against a lower bound of $\Omega(m \log d)$ rows. It is known that an $O(m \log d) \times d$ matrix of independent Gaussian entries has this property with high probability. Because it is expensive to store fully random numbers, researchers have also looked at pseudorandom and deterministic constructions. It is also known [RV06] that a random collection of $m \log^4(d)$ rows of a $d \times d$ Fourier matrix (or any unitary matrix whose entries have bounded magnitude) has the m-RIP. A technique in [CM06, Mut06b] gives a deterministic construction of a matrix with $m^2 \operatorname{polylog}(d)$ rows with the m-RIP.

This leads to the following open questions:

(1) Give a polynomial-time deterministic construction of a $m \operatorname{polylog}(d) \times d$ matrix with the m-RIP. One possibility is constructing a set of $m \operatorname{polylog}(d)$ rows of the Fourier matrix.
(2) Give a zero-error randomized construction of such a matrix. Equivalently, give a deterministic polynomial-time test for such matrices (which can be applied to randomized constructions).
(3) Improve the number of rows for the Fourier construction from $O(m \log^4 d)$ to $O(m \log d)$ (or show a larger lower bound for Fourier matrices in particular). If necessary, substitute another unitary, bounded-magnitude matrix of your choice for Fourier.

Some related open questions are as follows. A bottleneck in the runtime of [GSTV07] is the time to multiply an $m \times m$ submatrix F_{RC} of the the $d \times d$ Fourier matrix F by a vector v of length m.

(1) Provide a $o(m^2)$-time algorithm to multiply F_{RC} by v, given as worst case input v, the subset R of rows and the subset C of columns.
(2) Provide a $o(m^2)$-time algorithm to multiply F_{RC} by v, given as worst case input v and the subset C of columns, but given *random* set R of rows. The algorithm should take time $o(m^2)$ in expectation or with high probability with respect to R.
(3) Same questions, but with Fourier replaced by another unitary, bounded-magnitude matrix of your choice.

References

[AMS99] Noga Alon, Yossi Matias, and Mario Szegedy. The space complexity of approximating the frequency moments. *Journal of Computer and System Sciences*, 58(1):137–147, 1999.

[Bas06] Surender Baswana. Faster streaming algorithms for graph spanners, 2006.

[BCFM00] Andrei Z. Broder, Moses Charikar, Alan M. Frieze, and Michael Mitzenmacher. Min-wise independent permutations. *J. Comput. Syst. Sci.*, 60(3):630–659, 2000.

[BFL+06] Luciana S. Buriol, Gereon Frahling, Stefano Leonardi, Alberto Marchetti-Spaccamela, and Christian Sohler. Counting triangles in data streams. In *ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 253–262, 2006.

[BG06] Lakshminath Bhuvanagiri and Sumit Ganguly. Estimating entropy over data streams. In *ESA*, pages 148–159, 2006.

[BGKS06] Lakshminath Bhuvanagiri, Sumit Ganguly, Deepanjan Kesh, and Chandan Saha. Simpler algorithm for estimating frequency moments of data streams. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 708–713, 2006.

[BMMY07] S. Bhattacharyya, A. Madeira, S. Muthukrishnan, and T. Ye. How to scalably skip past streams. In *WSSP (Workshop with ICDE)*, 2007.

[BYJK+02] Ziv Bar-Yossef, T.S. Jayram, Ravi Kumar, D. Sivakumar, and Luca Trevisan. Counting distinct elements in a data stream. In *Proc. 6th International Workshop on Randomization and Approximation Techniques in Computer Science*, pages 1–10, 2002.

[BYKS02] Ziv Bar-Yossef, Ravi Kumar, and D. Sivakumar. Reductions in streaming algorithms, with an application to counting triangles in graphs. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 623–632, 2002.

[CCM07] Amit Chakrabarti, Graham Cormode, and Andrew McGregor. A near-optimal algorithm for computing the entropy of a stream. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 328–335, 2007.

[CDIM03] Graham Cormode, Mayur Datar, Piotr Indyk, and S. Muthukrishnan. Comparing data streams using hamming norms (how to zero in). *IEEE Trans. Knowl. Data Eng.*, 15(3):529–540, 2003.

[Cha02] Moses Charikar. Similarity estimation techniques from rounding algorithms. In *STOC*, pages 380–388, 2002.

[CKMS06] Graham Cormode, Flip Korn, S. Muthukrishnan, and Divesh Srivastava. Space- and time-efficient deterministic algorithms for biased quantiles over data streams. In *PODS*, pages 263–272, 2006.

[CM05a] Graham Cormode and S. Muthukrishnan. An improved data stream summary: the count-min sketch and its applications. *J. Algorithms*, 55(1):58–75, 2005.

[CM05b] Graham Cormode and S. Muthukrishnan. Space efficient mining of multigraph streams. In *ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 271–282, 2005.

[CM05c] Graham Cormode and S. Muthukrishnan. What's new: finding significant differences in network data streams. *IEEE/ACM Trans. Netw.*, 13(6):1219–1232, 2005.

[CM06] Graham Cormode and S. Muthukrishnan. Combinatorial algorithms for compressed sensing. In Paola Flocchini and Leszek Gasieniec, editors, *Structural Information and Communication Complexity, 13th International Colloquium, SIROCCO 2006, Chester, UK, July 2-5, 2006, Proceedings*, volume 4056 of *Lecture Notes in Computer Science*, pages 280–294. Springer, 2006.

[CRT06] Emmanuel J. Candès, Justin K. Romberg, and Terence Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(1):489–509, 2006.

[DFR06] Camil Demetrescu, Irene Finocchi, and Andrea Ribichini. Trading off space for passes in graph streaming problems. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 714–723, 2006.

[DG04] Jeffrey Dean and Sanjay Ghemawat. MapReduce: Simplified data processing on large clusters. In *OSDI*, pages 137–150, 2004.

[DH03] Doratha E. Drake and Stefan Hougardy. Improved linear time approximation algorithms for weighted matchings. In *RANDOM-APPROX*, pages 14–23, 2003.

[DLOM02] Erik D. Demaine, Alejandro López-Ortiz, and J. Ian Munro. Frequency estimation of internet packet streams with limited space. In *ESA*, pages 348–360, 2002.

[DMM06a] Petros Drineas, Michael W. Mahoney, and S. Muthukrishnan. Sampling algorithms for ℓ_2 regression and applications. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 1127–1136, 2006.

[DMM06b] Petros Drineas, Michael W. Mahoney, and S. Muthukrishnan. Subspace sampling and relative-error matrix approximation: Column-based methods. In *APPROX-RANDOM*, pages 316–326, 2006.

[DMM06c] Petros Drineas, Michael W. Mahoney, and S. Muthukrishnan. Subspace sampling and relative-error matrix approximation: Column-row-based methods. In *ESA*, pages 304–314, 2006.

[Don06] David L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.

[Edm65] Jack Edmonds. Maximum matching and a polyhedron with 0,1-vertices. *J. Res. Nat. Bur. Standards*, 69(B):125–130, 1965.

[Elk06] Michael Elkin. A near-optimal fully dynamic distributed algorithm for maintaining sparse spanners, 2006.

[EZ06] Michael Elkin and Jian Zhang. Efficient algorithms for constructing $(1+\epsilon, \beta)$-spanners in the distributed and streaming models. *Distributed Computing*, 18(5):375–385, 2006.

[FKM+05a] Joan Feigenbaum, Sampath Kannan, Andrew McGregor, Siddharth Suri, and Jian Zhang. Graph distances in the streaming model: the value of space. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 745–754, 2005.

[FKM+05b] Joan Feigenbaum, Sampath Kannan, Andrew McGregor, Siddharth Suri, and Jian Zhang. On graph problems in a semi-streaming model. *Theoretical Computer Science*, 348(2-3):207–216, 2005.

[FKSV02] Joan Feigenbaum, Sampath Kannan, Martin Strauss, and Mahesh Viswanathan. An approximate L^1 difference algorithm for massive data streams. *SIAM Journal on Computing*, 32(1):131–151, 2002.

[FMS+06] Jon Feldman, S. Muthukrishnan, Anastasios Sidiropoulos, Cliff Stein, and Zoya Svitkina. On the complexity of processing massive, unordered, distributed data, 2006.

[Gab90] Harold N. Gabow. Data structures for weighted matching and nearest common ancestors with linking. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 434–443, 1990.

[GE96] M. Gu and S.C. Eisenstat. Efficient algorithms for computing a strong rank-revealing QR factorization. *SIAM Journal on Scientific Computing*, 17:848–869, 1996.

[GGL+07] Martin Grohe, Yuri Gurevich, Dirk Leinders, Nicole Schweikardt, Jerzy Tyszkiewicz, and Jan Van den Bussche. Database query processing using finite cursor machines. In *ICDT*, pages 284–298, 2007.

[GIM07] Sudipto Guha, Piotr Indyk, and Andrew McGregor. Sketching information divergences. In *Conference on Learning Theory*, 2007.

[GK01] Michael Greenwald and Sanjeev Khanna. Space-efficient online computation of quantile summaries. In *ACM SIGMOD International Conference on Management of Data*, pages 58–66, 2001.

[GKMS02] Anna C. Gilbert, Yannis Kotidis, S. Muthukrishnan, and Martin Strauss. How to summarize the universe: Dynamic maintenance of quantiles. In *Proc. 28th International Conference on Very Large Data Bases*, pages 454–465, 2002.

[GL89] G.H. Golub and C.F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, 1989.

[GM06] Sudipto Guha and Andrew McGregor. Approximate quantiles and the order of the stream. In *ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 273–279, 2006.

[GM07a] Sumit Ganguly and Anirban Majumder. CR-precis: A deterministic summary structure for update data streams. In *ESCAPE*, 2007.

[GM07b] Sudipto Guha and Andrew McGregor. Lower bounds for quantile estimation in random-order and multi-pass streaming. *Manuscript*, 2007.

[GM07c] Sudipto Guha and Andrew McGregor. Space-efficient sampling. In *AISTATS*, pages 169–176, 2007.

[GMV06] Sudipto Guha, Andrew McGregor, and Suresh Venkatasubramanian. Streaming and sublinear approximation of entropy and information distances. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 733–742, 2006.

[Gou96] Stephen Jay Gould. *The Mismeasure of Man*. W. W. Norton and Company, 1996.

[GSTV07] A. C. Gilbert, M. J. Strauss, J. A. Tropp, and R. Vershynin. One sketch for all: fast algorithms for compressed sensing. *ACM Symposium on Theory of Computing*, 2007. To appear.

[GT01] S. A. Goreinov and E. E. Tyrtyshnikov. The maximum-volume concept in approximation by low-rank matrices. *Contemporary Mathematics*, 280:47–51, 2001.

[GTZ97] S. A. Goreinov, E. E. Tyrtyshnikov, and N. L. Zamarashkin. A theory of pseudoskeleton approximations. *Linear Algebra and its Applications*, 261:1–21, August 1997.

[HK73] John E. Hopcroft and Richard M. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM J. Comput.*, 2(4):225–231, 1973.

[HMR04] M. Hoffmann, S. Muthukrishnan, and R. Raman. Location streams: Models and algorithms. Technical Report 2004-28, DIMACS, May 2004.

[HSST04] John Hershberger, Nisheeth Shrivastava, Subhash Suri, and Csaba D. Tóth. Adaptive spatial partitioning for multidimensional data streams. In *ISAAC*, pages 522–533, 2004.

[Ind00] Piotr Indyk. Stable distributions, pseudorandom generators, embeddings and data stream computation. *IEEE Symposium on Foundations of Computer Science*, pages 189–197, 2000.

[Ind04] Piotr Indyk. Algorithms for dynamic geometric problems over data streams. *ACM Symposium on Theory of Computing*, pages 373–380, 2004.

[IT03] Piotr Indyk and Niten Thaper. Fast color image retrieval via embeddings. *Workshop on Statistical and Computational Theories of Vision (at ICCV)*, 2003.

[IW03] Piotr Indyk and David P. Woodruff. Tight lower bounds for the distinct elements problem. *IEEE Symposium on Foundations of Computer Science*, pages 283–288, 2003.

[IW05] Piotr Indyk and David P. Woodruff. Optimal approximations of the frequency moments of data streams. In *ACM Symposium on Theory of Computing*, pages 202–208, 2005.

[JG05] Hossein Jowhari and Mohammad Ghodsi. New streaming algorithms for counting triangles in graphs. In *COCOON*, pages 710–716, 2005.

[JKS07] T. S. Jayram, Ravi Kumar, and D. Sivakumar. Simple lower bound on one-way Gap-Hamming. In `http://www.cse.iitk.ac.in/users/sganguly/slides/ravikumar.pdf`, 2007.

[KR04] James F. Kurose and Keith W. Ross. *Computer Networking: A Top-Down Approach Featuring the Internet*. Addison Wesley, 2004.

[KS95] Bahman Kalantari and Ali Shokoufandeh. Approximation schemes for maximum cardinality matching. Technical Report LCSR–TR–248, Laboratory for Computer Science Research, Department of Computer Science. Rutgers University, August 1995.

[LHC06] Ping Li, Trevor Hastie, and Kenneth Ward Church. Very sparse random projections. In *KDD*, pages 287–296, 2006.

[Li06] Ping Li. Very sparse stable random projections, estimators and tail bounds for stable random projections, 2006.

[MAA05] Ahmed Metwally, Divyakant Agrawal, and Amr El Abbadi. Efficient computation of frequent and top-k elements in data streams. In *ICDT*, pages 398–412, 2005.

[McG05] Andrew McGregor. Finding graph matchings in data streams. In *APPROX-RANDOM*, pages 170–181, 2005.

[MG82] Jayadev Misra and David Gries. Finding repeated elements. *Sci. Comput. Program.*, 2(2):143–152, 1982.

[MMD06] Michael W. Mahoney, Mauro Maggioni, and Petros Drineas. Tensor-cur decompositions for tensor-based data. In *ACM SIGKDD international conference on knowledge discovery and data mining*, pages 327–336, 2006.

[MP80] J. Ian Munro and Mike Paterson. Selection and sorting with limited storage. *Theor. Comput. Sci.*, 12:315–323, 1980.

[MRL98] Gurmeet Singh Manku, Sridhar Rajagopalan, and Bruce G. Lindsay. Approximate medians and other quantiles in one pass and with limited memory. In *ACM SIGMOD International Conference on Management of Data*, pages 426–435, 1998.

[MRL99] Gurmeet Singh Manku, Sridhar Rajagopalan, and Bruce G. Lindsay. Random sampling techniques for space efficient online computation of order statistics of large datasets. In *ACM SIGMOD International Conference on Management of Data*, pages 251–262, 1999.

[Mut06a] S. Muthukrishnan. Data streams: Algorithms and applications. *Now Publishers*, 2006.

[Mut06b] S. Muthukrishnan. Some algorithmic problems and results in compressed sensing. In *Allerton Conference*, 2006.

[MV80] Silvio Micali and Vijay V. Vazirani. An $O(\sqrt{V}E)$ algorithm for finding maximum matching in general graphs. In *FOCS*, pages 17–27, 1980.

[NS06] Assaf Naor and Gideon Schechtman. Planar earthmover is not in l_1. In *FOCS*, pages 655–666, 2006.

[PS04] Seth Pettie and Peter Sanders. A simpler linear time 2/3-ϵ approximation for maximum weight matching. *Inf. Process. Lett.*, 91(6):271–276, 2004.

[RV06] Mark Rudelson and Roman Vershynin. Sparse reconstruction by convex relaxation: Fourier and gaussian measurements. In *Proceedins of 40th Annual Conference on Information Sciences and Systems*, 2006.

[SBAS04] Nisheeth Shrivastava, Chiranjeeb Buragohain, Divyakant Agrawal, and Subhash Suri. Medians and beyond: new aggregation techniques for sensor networks. In *SenSys*, pages 239–249, 2004.

[Ste99] G.W. Stewart. Four algorithms for the efficient computation of truncated QR approximations to a sparse matrix. *Numerische Mathematik*, 83:313–323, 1999.

[Ste04] G.W. Stewart. Error analysis of the quasi-Gram-Schmidt algorithm. Technical Report UMIACS TR-2004-17 CMSC TR-4572, University of Maryland, College Park, MD, 2004.

[Woo04] David P. Woodruff. Optimal space lower bounds for all frequency moments. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 167–175, 2004.