List of Open Problems in Sublinear Algorithms

https://sublinear.info/

This copy was made on April 29, 2017.

The goal of this copy is to provide an offline reference. Save trees! Please don't print it unless you really have to. The up to date versions of all problems are available at https://sublinear.info/ . Please send comments at admin@sublinear.info .

Open Problems:By Number

Problems suggested at the IITK Workshop on Algorithms for Data Streams 2006:

- Problem 1: Fast L_1 Difference
- Problem 2: Quantiles
- Problem 3: L_∞ Estimation
- Problem 4: Deterministic Summary Structures
- Problem 5: Characterizing Sketchable Distances
- Problem 6: Filtering Irrelevant Data
- Problem 7: Estimating Earth-Mover Distance
- Problem 8: Mixed Norms
- Problem 9: Open-Shortest-Path-First Routing
- Problem 10: Multi-Round Communication of Gap-Hamming Distance
- Problem 11: Counting Triangles
- Problem 12: Deterministic CUR-Type Decompositions
- Problem 13: Effects of Subsampling
- Problem 14: Graph Distances
- Problem 15: Semi-Random Streams
- Problem 16: Graph Matchings
- Problem 17: The Massive, Unordered, Distributed-Data Model
- Problem 18: Finite Cursor Machines
- Problem 19: Sketching vs. Streaming
- Problem 20: Relations between Streaming Models
- Problem 21: Deterministic Heavy-Hitters & Fast Matrix Algorithms

Problems suggested at the IITK Workshop on Algorithms for Processing Massive Data Sets 2009:

- Problem 22: Random Walks
- Problem 23: Approximate 2D Width
- Problem 24: “Ultimate” Deterministic Sparse Recovery
- Problem 25: Communication Complexity and Metric Spaces
- Problem 26: Equivalence of Two MapReduce Models
- Problem 27: Modeling of Distributed Computation
- Problem 28: Randomness of Partially Random Streams
- Problem 29: Strong Lower Bounds for Graph Problems
- Problem 30: Universal Sketching
- Problem 31: Gap-Hamming Information Cost
- Problem 32: The Value of a Reverse Pass
- Problem 33: Group Testing
- Problem 34: Linear Algebra Computation
- Problem 35: Maximal Complex Equiangular Tight Frames

Problems suggested at the Bertinoro Workshop on Sublinear Algorithms 2011:

- Problem 36: Learning an f-Transformed Product Distribution
- Problem 37: Testing Submodularity
- Problem 38: Query Complexity of Local Partitioning Oracles
- Problem 39: Approximating Maximum Matching Size
- Problem 40: Testing Monotonicity and the Lipschitz Property
- Problem 41: Testing Acyclicity
- Problem 42: Graph Frequency Vectors
- Problem 43: Rank Lower Bound
- Problem 44: Approximating LIS Length in the Streaming Model
- Problem 45: Streaming Max-Cut/Max-CSP
- Problem 46: Fast JL Transform for Sparse Vectors
- Problem 47: Annotated Streaming
- Problem 48: Sketching Shift Metrics
- Problem 49: Sketching Earth Mover Distance
- Problem 50: Sparse Recovery for Tree Models

Problems suggested at the Dortmund Workshop on Algorithms for Data Streams 2012:

- Problem 51: “For All” Guarantee for Computationally Bounded Adversaries
- Problem 52: TSP in the Streaming Model

- Problem 53: Homomorphic Hash Functions
- Problem 54: Faster JL Dimensionality Reduction
- Problem 55: Applications of Clifford Algebras in Graph Streams
- Problem 56: Efficient Measures of “Surprisingness” of Sequences
- Problem 57: Coding Theory in the Streaming Model
- Problem 58: Signatures for Set Equality
- Problem 59: Low Expansion Encoding of Edit Distance
- Problem 60: Single-Pass Unweighted Matchings

Problems suggested at the Bertinoro Workshop on Sublinear Algorithms 2014:

- Problem 61: RNA Folding
- Problem 62: Principal Component Analysis with Nonnegativity Constraints
- Problem 63: Submodular Matching Maximization
- Problem 64: Matchings in the Turnstile Model
- Problem 65: Communication Complexity of Connectivity
- Problem 66: Distinguishing Distributions with Conditional Samples
- Problem 67: Difficult Instance for Max-Cut in the Streaming Model
- Problem 68: Approximating Rank in the Bounded-Degree Model

Problems suggested at the Sublinear Algorithms Workshop 2016 at Johns Hopkins University:

- Problem 69: Correcting Independence of Distributions
- Problem 70: Open Problems in L_p-Testing
- Problem 71: Metric TSP Cost Approximation
- Problem 72: Communication Complexity of Approximating Set-Intersection Join
- Problem 73: Streaming Online Algorithms
- Problem 74: Succinct Representation for Functions on Graphs

Problems suggested at the Workshop on Communication Complexity and Applications 2017 at the Banff International Research Station:

- Problem 75: Data Structure Lower Bound in the Cell Probe Model
- Problem 76: External Information and Amortized Expected Communication
- Problem 77: Frontiers in Structural Communication Complexity
- Problem 78: Linear Sketching Over F_2
- Problem 79: Cryptogenography
- Problem 80: Merlin–Arthur Communication Complexity of Connectivity

Problem 1: Fast L_1 Difference

Suggested by	Graham Cormode
Source	Kanpur 2006
Short link	https://sublinear.info/1

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:

- The algorithms of Indyk and Woodruff [IndykW-05], and simplifications by Bhuvanagiri et al. [BhuvanagiriGKS-06] 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 [BhuvanagiriG-06]. Can it also apply to L_1?
- Recent work [LiHC-06] has studied sparse random projections for L_2. Follow up work [Li-06] 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.

Problem 2: Quantiles

Suggested by	Graham Cormode
Source	Kanpur 2006
Short link	https://sublinear.info/2

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 [MankuRL-98,MankuRL-99,GreenwaldK-01,GilbertKMS-02,CormodeM-05,ShrivastavaBAS-04,GuhaM-06]. For deterministic algorithms on insert only streams, two algorithms obtain the best (and incomparable) space bounds: $O(\epsilon^{-1} \log \epsilon N)$ words [GreenwaldK-01] and $O(\epsilon^{-1} \log U)$ words [ShrivastavaBAS-04], where U is the size of the domain from which the input is drawn.

The Greenwald-Khanna algorithm [GreenwaldK-01] 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 [ShrivastavaBAS-04] is much simpler to analyze and more amenable to variations, meaning that several generalizations and alternatives have been proposed [HershbergerSST-04,CormodeKMS-06]. 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 Greenwald-Khanna algorithm, or a variation thereof, submit to a simpler analysis which will allow generalizations of the algorithm to be more easily proposed and studied?

Problem 3: L_∞ Estimation

Suggested by	Graham Cormode
Source	Kanpur 2006
Short link	https://sublinear.info/3

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 $O(\epsilon^{-1}(\log m + \log L_1))$ [MisraG-82,MetwallyAA-05], for computing on the difference between two streams the bounds are $O(\epsilon^{-1} \log m(\log m + \log L_1))$[CormodeM-05a]. 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 [CormodeM-05].[1]

Notes

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.

Problem 4: Deterministic Summary Structures

Suggested by	Sumit Ganguly
Source	Kanpur 2006
Short link	https://sublinear.info/4

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 $1 - \delta$ [CormodeM-05]. This is nearly optimal as the space lower bound is $O(\epsilon^{-1} \log(m) \log \epsilon n)$ [Ganguly-06].

However, in practice it is desirable to have deterministic algorithms rather than randomized algorithms. Using a deterministic collection of primes [Muthukrishnan-06], Ganguly [Ganguly-06] 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.

Problem 5: Characterizing Sketchable Distances

Suggested by	Sudipto Guha and Piotr Indyk
Source	Kanpur 2006
Short link	https://sublinear.info/5

Some of the early successes in developing algorithms for the data stream model related to estimating L_p norms [FeigenbaumKSV-02,Indyk-00,AlonMS-99] and the "Hamming norm" L_0 [CormodeDIM-03]. What other distances, or more generally "measures of dissimilarity," can be approximated in the data stream 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 [BroderCFM-00].

Recent work provides some preliminary results [GuhaIM-07]. 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.

Update

It was shown that for **norms** small sketches are equivalent to good embeddings into $\ell_{1-\varepsilon}$ [AndoniKR-14].

Problem 6: Filtering Irrelevant Data

Suggested by	Sariel Har-Peled
Source	Kanpur 2006
Short link	https://sublinear.info/6

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 [DemetrescuFR-06]. 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.

Problem 7: Estimating Earth-Mover Distance

Suggested by	Piotr Indyk
Source	Kanpur 2006
Short link	https://sublinear.info/7

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 [Indyk-04]. That algorithm proceeds essentially by embedding EMD into L_1 [Charikar-02,IndykT-03]. However, any such embedding must incur at least $\Omega(\sqrt{\log \Delta})$ distortion [NaorS-06]. So one would need to do something else to get $O(1)$-approximation.

Update

It was shown that the decision version of the problem (distinguish the cases, when the EMD distance is at most 1 vs. at least D for $D = O(1)$) does not allow sketches of the constant size [AndoniKR-14].

Problem 8: Mixed Norms

Suggested by	Piotr Indyk
Source	Kanpur 2006
Short link	https://sublinear.info/8

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 $O(\mathrm{poly}(\epsilon^{-1}) \cdot \sqrt{n} \cdot \mathrm{polylog}(n))$ is known as long as $A \geq 0$ [CormodeM-05b]. There are no non-trivial lower bounds known.

Problem 9: Open-Shortest-Path-First Routing

Suggested by	Sampath Kannan
Source	Kanpur 2006
Short link	https://sublinear.info/9

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. [KuroseR-04]). 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.

Problem 10: Multi-Round Communication of Gap-Hamming Distance

Suggested by	Ravi Kumar
Source	Kanpur 2006
Short link	https://sublinear.info/10

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 $\Omega(n)$ [IndykW-03,Woodruff-04]. Recently, a simpler proof was discovered using a reduction from **Index** [JayramKS-07]. 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 [IndykW-03,Woodruff-04,BarYossefJKST-02] and entropy [BhuvanagiriG-06,ChakrabartiCM-07]. 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.

Update

This conjecture was proved by Chakrabarti and Regev [ChakrabartiR-11], who showed that the communication complexity of **Gap-Hamdist** is $\Omega(n)$.

Problem 11: Counting Triangles

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 focused on the case of insertions [BarYossefKS-02,JowhariG-05,BuriolFLMS-06] although it appears that one of the algorithms in [JowhariG-05] may work when edges can be deleted. Is it possible to match the insert-only bounds when edges are inserted and deleted?

Suggested by	Stefano Leonardi
Source	Kanpur 2006
Short link	https://sublinear.info/11

Update

Ahn, Guha, and McGregor [AhnGM-12b] proposed an algorithm for streams with both insertions and deletions. It matches the best known bounds for insertion-only streaming algorithms [BuriolFLMS-06].

Problem 12: Deterministic CUR-Type Decompositions

Suggested by	Michael Mahoney
Source	Kanpur 2006
Short link	https://sublinear.info/12

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 [DrineasMM-06,DrineasMM-06a,DrineasMM-06b] 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:

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

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" [Gould-96], 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 $[(1/2)\,\mathrm{age} - (1/\sqrt{2})\,\mathrm{height} + (1/2)\,\mathrm{income}]$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 [MahoneyMD-06]. 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. [DrineasMM-06b]). *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 $1-\delta$,*
>
> $$\|A - CUR\|_{\mathrm{F}} \leq (1+\epsilon)\|A - A_k\|_{\mathrm{F}}.$$
>
> *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 $O(\mathrm{SVD}(A_k))$, which is the time required to compute the best rank-k approximation to the SVD [GolubV-89].*

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 [GuE-96] and Stewart [Stewart-99,Stewart-04] may help towards this goal. Also relevant is work by Goreinov, Tyrtyshnikov, and Zamarashkin [GoreinovTZ-97,GoreinovT-01] 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 [GoreinovTZ-97], 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 [GoreinovT-01] 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.

Problem 13: Effects of Subsampling

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?

Suggested by	Yossi Matias
Source	Kanpur 2006
Short link	https://sublinear.info/13

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 [BhattacharyyaMMY-07].

Problem 14: Graph Distances

Suggested by	Andrew McGregor
Source	Kanpur 2006
Short link	https://sublinear.info/14

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* [FeigenbaumKMSZ-05,FeigenbaumKMSZ-05a,ElkinZ-06,Baswana-06,Elkin-06] 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 [FeigenbaumKMSZ-05]. 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?

Update

Guruswami and Onak [GuruswamiO-13] showed that checking if $d_G(u,v) \leq 2(p+1)$ in p passes, where $p = O\left(\frac{\log n}{\log\log n}\right)$, requires $\Omega\left(\frac{n^{1+1/(2p+2)}}{p^{20}\log^{3/2} n}\right)$ bits of space.

Problem 15: Semi-Random Streams

Suggested by	Andrew McGregor
Source	Kanpur 2006
Short link	https://sublinear.info/15

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 [MunroP-80,DemaineLM-02,GuhaMV-06,GuhaM-06,GuhaM-07,GuhaM-07a]. 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 [GuhaM-07a]. 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 [GuhaM-06]:

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?

Problem 16: Graph Matchings

Suggested by	Andrew McGregor
Source	Kanpur 2006
Short link	https://sublinear.info/16

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 [Edmonds-65,Gabow-90,HopcroftK-73,MicaliV-80]. 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 [KalantariS-95]. The best general result is a linear time $(2/3 - \epsilon)$-approximation [DrakeH-03,PettieS-04].

Algorithms in the data stream model were presented in [McGregor-05]. 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[1] 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[2]. 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?

Notes

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

Problem 17: The Massive, Unordered, Distributed-Data Model

Suggested by	S. Muthkrishnan
Source	Kanpur 2006
Short link	https://sublinear.info/17

The Massive, Unordered, Distributed-data (MUD) model was recently introduced by Feldman et al. [FeldmanMSSS-06] as an abstraction of part of the infrastructure used at Google. It is related to the MapReduce framework presented in [DeanG-04]. 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.

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 [FeldmanMSSS-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?

Problem 18: Finite Cursor Machines

Suggested by	Nicole Schweikardt
Source	Kanpur 2006
Short link	https://sublinear.info/18

The Finite Cursor Machine (FCM) model is an abstract model for database query processing based on abstract state machines [GroheGLSTV-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., $\mathcal{U} = (\mathbb{N}, \leq, +, \times)$)
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.

Problem 19: Sketching vs. Streaming

Suggested by	D. Sivakumar
Source	Kanpur 2006
Short link	https://sublinear.info/19

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 [FeldmanMSSS-06] for further details.

Problem 20: Relations between Streaming Models

Suggested by	Christian Sohler
Source	Kanpur 2006
Short link	https://sublinear.info/20

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 [HoffmannMR-04] 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.

Problem 21: Deterministic Heavy-Hitters & Fast Matrix Algorithms

Suggested by	Martin Strauss
Source	Kanpur 2006
Short link	https://sublinear.info/21

An important ingredient in many recent algorithms for heavy hitters is the Restricted Isometry Property, defined in [CandesRT-06] and, equivalently, in [Donoho-06]. 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 [RudelsonV-06] 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 [CormodeM-06,Muthukrishnan-06a] 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 [GilbertSTV-07] 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.

Problem 22: Random Walks

The paper of Das Sarma, Gollapudi, and Panigrahy [DasSarmaGP-08] shows a method for performing random walks in the streaming model. In particular, a random walk of length l can be simulated using $O(n)$ space and $O(\sqrt{l})$ passes over the input stream. Is it possible to simulate such a random walk using $\tilde{O}(n)$ space and a much smaller number of passes, say, at most polylogarithmic in n and l? The goal is either to show an algorithm or prove a lower bound.

Suggested by	Rina Panigrahy
Source	Kanpur 2009
Short link	https://sublinear.info/22

Das Sarma et al. [DasSarmaGP-08] simulate random walks to approximate the probability distribution on the vertices of the graph after a random walk of length l. What is the streaming complexity of approximating this distribution? What is the streaming complexity of finding the k (approximately) most likely vertices after a walk of length l?

Problem 23: Approximate 2D Width

Suggested by	Pankaj Agarwal and Piotr Indyk
Source	Kanpur 2009
Short link	https://sublinear.info/23

The width of a set P of points in the plane is defined as

$$\mathrm{width}(P) = \min_{\|a\|_2=1} \left(\max_{p \in P} a \cdot p - \min_{p \in P} a \cdot p \right).$$

For a stream of insertions and deletions of points from a $[\Delta] \times [\Delta]$ grid, we would like to maintain an approximate width of the point set. We conjecture that there is an algorithm for this problem that achieves a constant approximation factor and uses $\mathrm{polylog}(\Delta + n)$ space.

Update

The conjecture has been resolved (in the positive direction) by Andoni and Nguyen [AndoniN-12].

Problem 24: “Ultimate” Deterministic Sparse Recovery

Suggested by	Piotr Indyk
Source	Kanpur 2009
Short link	https://sublinear.info/24

We say that a vector $v \in \mathbb{R}^n$ is *k-sparse* for some $k \in \{0, \ldots, n\}$ if there are no more than k non-zero coordinates in v. The goal in the problem being considered is to design an $m \times n$ matrix A such that for any $x \in R^n$, one can recover from Ax a vector $x^* \in \mathbb{R}^n$ that satisfies the following “L_2/L_1” approximation guarantee:

$$\|x^* - x\|_2 \leq \min_{k\text{-sparse } x' \in \mathbb{R}^n} \frac{C}{\sqrt{k}} \|x' - x\|_1,$$

where $C > 0$ is a constant.

We conjecture that there is a solution that (a) uses $m = O(k \log(n/k))$ *and* (b) supports recovery algorithms running in time $O(n \operatorname{polylog} n)$.

Background

It is known that one can achieve *either* (a) *or* (b) (see, e.g., [NeedellT-10]). It is also possible to achieve both (a) and (b), but with a different “L_1/L_1” approximation guarantee, where $\|x^* - x\|_1 \leq \min_{k\text{-sparse } x'} C\|x' - x\|_1$ [IndykR-08,BerindeIR-08].

Problem 25: Communication Complexity and Metric Spaces

Poincaré Inequalities

Suggested by	T. S. Jayram
Source	Kanpur 2009
Short link	https://sublinear.info/25

Alice and Bob are given two points x and y, respectively, from a specific metric space $\mathcal{M}$. We are interested in deciding whether $d_{\mathcal{M}}(x,y) \leq R$ or $d_{\mathcal{M}}(x,y) \geq \alpha R$, where $d_{\mathcal{M}}$ is the distance function of $\mathcal{M}$, $R > 0$, and $\alpha > 1$. What amount of information must be exchanged in order to solve this problem? Answering this question is interesting in any standard communication model: unrestricted communication between the players, one-way communication, sketching, etc.

The above question can partially be answered if the metric satisfies a specific "gap" Poincaré inequality [AndoniJP-10]. It is known that another kind of Poincaré inequality is equivalent to non-embeddability into ℓ_2^2 [Matousek-02], but it is not known if non-embeddability into ℓ_2^2 implies lower bounds for communication complexity. Can one show a formal connection between the communication complexity for approximating the distance between two points and non-embeddability into ℓ_2^2?

Product Metrics

We are also interested in the following general class of metrics. Let each $\mathcal{M}_i = \langle S_i, d_i \rangle$, $1 \leq i \leq k$, be a metric space on a set S_i with a distance function d_i. A *product metric space* $\bigoplus_{i=1}^{k} \mathcal{M}_i$ is defined on the product $S_1 \times \ldots \times S_k$ with the distance function

$$d\left((x_1, \ldots, x_k), (y_1, \ldots, y_k)\right) = \mathbf{op}(d_1(x_1, y_1), \ldots, d_k(x_k, y_k)),$$

where $\mathbf{op}$ is a symmetric operator. For instance, $\bigoplus_{i=1}^{k} \mathcal{M}_i$ is a proper metric space if $\mathbf{op}$ is the maximum operator or the p-th norm for any $p \in [1, \infty)$. The case when $\bigoplus_{i=1}^{k} \mathcal{M}_i$ is not necessarily a metric space also finds applications.

Applications of product metric spaces include a nearest neighbor data structure for Ulam distance [AndoniIK-09], and a near-linear time subpolynomial-approximation algorithm for edit distance [AndoniO-09].

The following questions arise in the context of product spaces:

1. Can one design efficient communication protocols for computing the distance between a pair of points? Suppose that there is an efficient communication protocol for each $\mathcal{M}_i$. What is the communication complexity for computing the distance between two points in $\bigoplus_{i=1}^{k} \mathcal{M}_i$? Andoni, Jayram, and Pătraşcu [AndoniJP-10] prove lower bounds for some product metrics. Jayram and Woodruff [JayramW-09] show streaming algorithms which yield communication protocols.
2. Can one design efficient streaming algorithms and data structures for product metric spaces? In particular, can one efficiently compute the distance between a pair of points? Jayram and Woodruff [JayramW-09] consider the related question of computing *cascaded norms*.

Update

It has been shown [AndoniKR-14] that for **normed spaces** the above implication is true: if a normed space does not embed into ℓ_2^2 (in fact, more generally, does not *uniformly embed* into a Hilbert space), then, there is a non-trivial communication lower bound for distinguishing small and large distances.

Problem 26: Equivalence of Two MapReduce Models

Suggested by	Paul Beame
Source	Kanpur 2009
Short link	https://sublinear.info/26

The original MapReduce paper [DeanG-04] gives two distributed models. First it only says that intermediate key/value pairs with the same key are combined and sent as batch jobs to workers. Then in Section 4.2, it additionally guarantees that the batch jobs received by a single worker are sorted according to the corresponding key values. There are algorithms that rely on this additional feature of MapReduce. Are these two models equivalent? For decision problems in the complexity world, we know strong time-space trade-offs for sorting, but no similar lower bounds are known for distinctness.

Problem 27: Modeling of Distributed Computation

Suggested by	Paul Beame
Source	Kanpur 2009
Short link	https://sublinear.info/27

MapReduce has recently inspired two distributed models of computation in the theory community. One is the MUD model of Feldman et al. [FeldmanMSSS-10]. In this model they assume that every worker has at most a polylogarithmic amount of space available, which in total gives at most $\tilde{O}(n)$ space, where n is the input size (in the order of at least terabytes). The other model of computation, due to Karloff et al. [KarloffSV-10], assumes that each of $n^{1-\epsilon}$ workers has at most $n^{1-\epsilon}$ space, where ϵ is a fixed positive constant. This totals to $n^{2-2\epsilon}$ space in the entire system. Can one design an interesting and practical model that only uses $n^{1+o(1)}$ space/resources?

Problem 28: Randomness of Partially Random Streams

Suggested by	Sudipto Guha
Source	Kanpur 2009
Short link	https://sublinear.info/28

Many streaming algorithms are designed for worst-case inputs and the first step of analysis is conducted using truly random hash functions, which in the second step are replaced by hash functions that can be described using a small number of truly random bits. In practice, the input stream is often a result of some random process. Mitzenmacher and Vadhan [MitzenmacherV-08] show that as long as it has sufficiently large entropy, hash functions from a restricted family are essentially as good as truly hash functions. On a related note, Gabizon and Hassidim [GabizonH-10] show that algorithms for random-order streams need essentially no additional entropy apart from what can be extracted from the input.

In these two cases, the input can be seen as a source of randomness for the algorithm. How can one quantify the randomness of the stream in a natural way? For instance, Mitzenmacher and Vadhan set a lower bound for the Renyi entropy of each element of the stream, conditioned on the previous elements of the stream. Are there measures that are likely to be useful in practice and that are possible to verify?

Once we fix a measure of randomness, how much randomness according to this measure is necessary to derandomize or simplify specific streaming algorithms?

Problem 29: Strong Lower Bounds for Graph Problems

Suggested by	Krzysztof Onak
Source	Kanpur 2009
Short link	https://sublinear.info/29

A large number of streaming papers consider graph problems. Typically, the input stream is an arbitrarily-ordered sequence of edges. For many problems, one can show that solving the problem, even approximately, requires $\Omega(n)$ bits of space. For instance, one can relatively easily prove that finding a constant-factor approximation to the maximum matching problem requires $\Omega(n)$ bits of space. Therefore, in many cases, the desired space complexity is of the form $\tilde{O}(n)$. Despite this relaxation, it is plausible that for some popular problems, there are barriers that cannot be overcome by (approximate) algorithms that use $n^{1+o(1)}$ space and a small number of passes.

For example, let $M(G)$ be the maximum matching size in the input graph G. McGregor [McGregor-05] shows that there is an algorithm that finds a matching of size $(1-\epsilon) \cdot M(G)$ in a number of passes that is a function of only ϵ. It is plausible that for any constant k, there is no k-pass $\tilde{O}(n)$-space algorithm that finds a matching of size greater than $(1-\epsilon_k) \cdot M(G)$ times the optimum, where ϵ_k is a positive constant. In particular, to the best of my knowledge, no one-pass $\tilde{O}(n)$-space algorithm that finds a $(1-\epsilon)$-approximation for any constant $\epsilon \in (0, 1/2)$ is known. Can one prove lower bounds as suggested above? The question generalizes to other problems. For instance, the best known $\tilde{O}(n)$-space algorithms for simulating random walks require a large number of passes (see [DasSarmaGP-08] and Rina Panigrahy's question). Can one prove for these problems that a small number of passes requires $n^{1+\Omega(1)}$ space?

To the best of my knowledge, computing the BFS tree and computing the diameter are the only problems for which an $n^{1+\Omega(1)}$ lower bound for more than one pass is known [FeigenbaumKMSZ-08].

Update

Guruswami and Onak [GuruswamiO-13] showed that the following problems require roughly $n^{1+\Omega(1/p)}$ bits of space in p passes: testing if there is a perfect matching, checking if v and w are at distance at most $2(p+1)$, and checking if there is a directed path from v to w, where v and w are known to the algorithm in advance.

Problem 30: Universal Sketching

Suggested by	Jelani Nelson
Source	Kanpur 2009
Short link	https://sublinear.info/30

Rather than designing different sketching algorithms for every problem, it would be desirable to have algorithms that where *universal*, in some sense, for a variety of problems. Specifically, let $\mathcal{F}$ be a family of functions mapping frequency vector $[-M, M]^n$ to $\mathbb{R}$. We say could say a sketching algorithm A is (ϵ, δ) universal for $\mathcal{F}$ if for all $x \in [-M, M]^n$, A can recover a $(1 + \epsilon)$ approximation each $f(x)$ for any $f \in \mathcal{F}$ with probability $1 - \delta$.

An example would be when $\mathcal{F}$ is $\{F_p : 0 \leq p \leq 2\}$. A simple approach would be to discretize p and to utilize the fact that $\ell_p(x) \approx \ell_{p'}(x)$ if p and p' are sufficiently close. Better yet would be to interpolate through a small set of values, using ideas from Harvey, Nelson, and Onak [HarveyNO-08]. Consequently it should be possible to be universal for $\mathcal{F} = \{F_p : 0 \leq p \leq 2\}$while using only slightly more space than that required to estimate a specific F_p. For what other families are there efficient universal algorithms? It seems that the Indyk-Woodruff [IndykW-05] technique would be useful here, and that the work of Braverman and Ostrovsky [BravermanO-10] is also highly relevant.

Problem 31: Gap-Hamming Information Cost

Suggested by	Amit Chakrabarti
Source	Kanpur 2009
Short link	https://sublinear.info/31

In the Gap-Hamming problem, two players Alice and Bob have vectors $x, y \in \{0,1\}^n$ respectively and wish to compute the function f

$$f(x,y) = \begin{cases} 0 & \text{if } \Delta(x,y) \leq n/2 - \sqrt{n} \\ 1 & \text{if } \Delta(x,y) \geq n/2 + \sqrt{n} \end{cases}$$

where $\Delta(x,y) = |\{i : x_i \neq y_i\}|$ is the Hamming distance between the vectors and we are promised that $|\Delta(x,y) - n/2| \geq \sqrt{n}$. The problem became interesting in the streaming community because a lower bound on the communication complexity of evaluating f yields a lower bound on the space required by a streaming algorithm to estimate the number of distinct elements or the entropy of a stream. After a series of papers, it is know that evaluating f requires $\Omega(n)$ communication [IndykW-03,Woodruff-04,BrodyC-09,BrodyCRVW-10,ChakrabartiR-11] even if an unlimited number of rounds of communication are used.

An increasingly popular technique in communication complexity is to prove bounds by bounding the information cost [ChakrabartiSWY-01,BarYossefJKS-04]. Here we consider random input (X,Y) and consider the mutual information between the input and the random transcript of the protocol $\Pi(X,Y)$:

$$I(XY; \Pi(X,Y)) = H(XY) - H(XY|\Pi(X,Y)) \; .$$

It would be interesting to prove a lower bound on the information cost for the Gap-Hamming problem for some natural input distribution.

Update

A linear lower bound on the information cost for a certain distribution was shown by [KerenidisLLRX-12]. Later, [BravermanGPW-13] showed that a linear lower bound also holds for the uniform distribution.

Problem 32: The Value of a Reverse Pass

Suggested by	Andrew McGregor
Source	Kanpur 2009
Short link	https://sublinear.info/32

Multi-pass stream algorithms have been designed for a range of problems including longest increasing subsequences [LibenNowellVZ-06,GuhaM-08], graph matchings [McGregor-05], and various geometric problems [ChanC-07]. However, the existing literature almost exclusively considers the case when the multiple passes are in the same direction. One exception is recent work by Magniez et al. [MagniezMN-10] on the **DYCK**$_2$ problem: given a length n string in the alphabet "(,),[,]", determine whether it is well-parenthesized, i.e., it can be generated by the grammar $S \rightarrow (S) \mid [S] \mid SS \mid \epsilon$? For this problem it can be shown that with one forward and one reverse pass over the input, the problem can be solved with $O(\log^2 n)$ space. On the other hand, any algorithm using $O(1)$ forward passes and no reverse passes, requires $\Omega(\sqrt{n})$ space [ChakrabartiCKM-10,JainN-10]. For what other natural problems is there such a large separation?

Problem 33: Group Testing

Suggested by	Ely Porat
Source	Kanpur 2009
Short link	https://sublinear.info/33

Given a set $S \subset [n]$ of size at most k, we want to identify S by the following 2-stage process.

1. We choose a set of subsets $T_1, \ldots, T_m \subset [n]$. For each T_i we learn whether or not $T_i \cap S = \emptyset$.
2. Based on the outcomes of the first m tests, we may choose $j_1, \ldots, j_{O(k)} \in [n]$. For each j_i we learn whether or not $j_i \in S$.

The goal is to minimize m, the number of tests performed in the first stage. Without any further restrictions it has been shown that $m = O(k \log n/k)$ suffices [BonisGV-05]. However, for various pattern matching applications, we have the constraint that each T_i needs to be an arithmetic progression, e.g., $T_i = \{2, 8, 14, 20, \ldots\}$. In this case, $m = O(k \log^2 n)$ suffices. Is it possible to decrease this to $m = O(k \log n)$?

Problem 34: Linear Algebra Computation

It is often not the case that the entire data sits on a single machine and that we are allowed to make one or more passes over it. Instead the data is often distributed across multiple systems. This is one of the reasons why the streaming model does not have more impact in practice for linear algebra computation. It would be great to design new models that address this shortcoming.

Suggested by	Michael Mahoney
Source	Kanpur 2009
Short link	https://sublinear.info/34

Consider also the following problem. Let A be an $m \times n$ matrix and let k be a rank parameter. Let $P_{A,k}$ be the projection matrix on the best rank-k left (or right) singular subspace. The goal is to compute the diagonal of $P_{A,k}$ exactly or approximately in a small number of passes in the streaming model, or even better, in a new model that addresses the aforementioned shortcoming.

Problem 35: Maximal Complex Equiangular Tight Frames

Suggested by	Joel Tropp
Source	Kanpur 2009
Short link	https://sublinear.info/35

Consider a system of unit vectors $\{x_k : k = 1, 2, \dots, N\}$ in $\mathbb{C}^d$. It can be shown that the maximum inner product among these vectors satisfies the Welch bound

$$\max_{i \neq j} |\langle x_i, x_j \rangle| \geq \sqrt{\frac{N - d}{d(N - 1)}}.$$

Miraculously, when this bound is attained, the modulus of the inner product between every pair of vectors is identical. Such a configuration is referred to as an equiangular tight frame (ETF).

It can be shown that the cardinality N of an ETF is $\mathbb{C}^d$ must satisfy the bound $N \leq d^2$. When this bound is attained, the ETF is referred to as a maximal ETF. In other words, a maximal ETF is a system of d^2 unit vectors in $\mathbb{C}^d$ whose pairwise inner products share the modulus $(d + 1)^{-1/2}$.

A striking geometric fact about maximal ETFs is that each one corresponds with a regular simplex consisting of d^2 points embedded in the set of rank-one, trace-one, complex, Hermitian matrices with dimension d. This correspondence is achieved by mapping each vector x in the ETF to the matrix xx^*. Researchers believe that there is a maximal ETF for every dimension d. This question, so far, has resisted all efforts at solution.

Problem 36: Learning an f-Transformed Product Distribution

Suggested by	Rocco A. Servedio
Source	Bertinoro 2011
Short link	https://sublinear.info/36

In this learning setting there are n independent Bernoulli random variables $X_1, \ldots, X_n$ with *unknown* $E[X_i] = p_i$. There is a *known* transformation function $f : \{0,1\}^n \mapsto R$, where R is some range. The learner has access to independent draws from $f(X_1, \ldots, X_n)$; i.e., each example for the learner is obtained by independently drawing $X_1, \ldots, X_n$, applying f, and giving the result to the learner. Call this distribution D_f. The learner's job is to construct a hypothesis distribution D' over the range set such that the variation distance between D_f and D' is at most ϵ, with high probability.

Question: Give some necessary or sufficient conditions on f that make the "learn an f-transformed product distribution" problem solvable using $O_\epsilon(1)$ queries, independent of n.

Background: The following is known [DaskalakisDS-11]:

1. For $f(X) = X_1 + \ldots + X_n$, there's a learning algorithm using poly$(1/\epsilon)$ queries independent of n.
2. For $f(X) = \sum_{i=1}^{n} i \cdot X_i$, any algorithm for learning to constant accuracy must make $\Omega(n)$ queries.

Problem 37: Testing Submodularity

Suggested by	C. Seshadhri
Source	Bertinoro 2011
Short link	https://sublinear.info/37

A function $f : \{0,1\}^n \mapsto R$ is *submodular* if for every $i \in [n]$ and every $S \subset T$, such that $i \notin T$,

$$f(T \cup \{i\}) - f(T) \leq f(S \cup \{i\}) - f(S) \ .$$

Question: How efficient can we test that f is submodular (in terms of number of queries to f). In particular, can one do it in $\text{poly}(n/\epsilon)$? Special cases of interest that are open:

- f is monotone and for every S and $i \in [n]$, $f(S \cup \{i\}) - f(S)$ is either 0 or 1. In this case f is the rank function of a matroid.
- A more special case (suggested by Noam Nisan): f is said to be a *coverage valuation* if every $i \in [n]$ is associated with a set V_i from some measurable space with a measure μ (we might want to think of V_i as discrete, in which case the measure is just the cardinality). Then f is defined by $f(S) = \mu(\bigcup_{i \in S} V_i)$. Observe that such f is a submodular function.

Background: The problem is interesting in algorithmic game theory. The best known upper bound on the number of queries is $O(\epsilon^{-\sqrt{n} \log n})$ [SeshadhriV-11]. We do not know the answer even for constant size R, although for $R = \{0,1\}$ it is easy.

Problem 38: Query Complexity of Local Partitioning Oracles

A local partitioning oracle is defined in the paper of Hassidim, Kelner, Nguyen, and Onak [HassidimKNO-09], and an implicit construction of a partitioning oracle is shown in the earlier paper of Benjamini, Schramm, and Shapira [BenjaminiSS-08]. Partitioning oracles are a useful abstraction for approximation and testing algorithms in the bounded degree model.

Suggested by	Krzysztof Onak
Source	Bertinoro 2011
Short link	https://sublinear.info/38

The best known oracle for bounded-degree planar graphs makes at most $d^{\mathrm{poly}(1/\epsilon)}$ queries to the underlying graph to answer each query about the resulting partition, where d is the bound on the maximum vertex degree in the graph. See [Onak-10] for a description of the method.

Question: Can one design an oracle that makes only $\mathrm{poly}(d/\epsilon)$ queries? If so, then among other things, this would lead to a tester for planarity in the bounded-degree model that makes only $\mathrm{poly}(1/\epsilon)$ queries, resolving an open question of Benjamini et al. [BenjaminiSS-08].

Update

Levi and Ron [LeviR-13] showed a partitioning oracle for bounded-degree minor-free graphs that makes only $(d/\epsilon)^{O(\log(1/\epsilon))}$ queries to the input graph for each query about the partition.

Problem 39: Approximating Maximum Matching Size

Consider graphs with maximum degree bounded by d. It is possible to approximate the size of the maximum matching up to an additive ϵn in time that is a function of only ϵ and d [NguyenO-08,YoshidaYI-09]. The fastest currently known algorithm runs in $d^{O(1/\epsilon^2)}$ time [YoshidaYI-09].

Question: Is there an algorithm that runs in $\mathrm{poly}(d/\epsilon)$ time?

Suggested by	Krzysztof Onak
Source	Bertinoro 2011
Short link	https://sublinear.info/39

Problem 40: Testing Monotonicity and the Lipschitz Property

Suggested by	Sofya Raskhodnikova
Source	Bertinoro 2011
Short link	https://sublinear.info/40

Positive answers to the conjectures below would imply better testers for monotonicity and the Lipschitz property. Consider a function $f : \{0,1\}^d \to \mathbb{R}$. It corresponds to a d-dimensional hypercube with the vertex set $\{0,1\}^d$ and (directed or undirected, depending on the problem) edges (x, y) for all x and y, where y can be obtained from x by increasing one bit. Each node x is labeled by a real number $f(x)$.

1. A directed edge (x, y) of the hypercube is *violated* if $f(x) > f(y)$. Function f is *monotone* if no edges are violated.
   Question: Suppose v edges are violated. Give an upper bound on the number of node labels that have to be changed to make f monotone.
   Background: The best known bound is vd [DodisGLRRS-99] but the conjecture is v.
2. An undirected edge (x, y) of the hypercube is *violated* if $|f(x) - f(y)| > 1$. Function f is *Lipschitz* if no edges are violated.
   Question: Suppose v edges are violated. Give an upper bound on the number of node labels that have to be changed to make function f Lipschitz in terms of v and d.
   Background: Nothing nontrivial is known for real labels. The conjecture is $O(v)$.

For integer labels, the best known bound is $2v \cdot \mathrm{ImageDiameter}(f)$ where $\mathrm{ImageDiameter}(f) = \max_x f(x) - \min_x f(x)$ [JhaR-11].

Update

The conjecture has been resolved (in the positive direction) by Chakrabarty and Seshadhri [ChakrabartyS-13].

Problem 41: Testing Acyclicity

Suggested by	Dana Ron
Source	Bertinoro 2011
Short link	https://sublinear.info/41

Consider the problem of testing acyclicity in *directed* bounded-degree graphs (in the incidence list model, where one can query both outgoing and incoming edges).

Question: What is the best algorithm for this problem?

Background: There is a lower bound of $\Omega(n^{1/3})$ for adaptive, two-sided error algorithms, where n is the number of vertices [BenderR-02]. No sublinear upper bound is known. (For dense graphs, in the adjacency matrix model, one can test the property using $\mathrm{poly}(1/\epsilon)$ queries.) The best known lower bound for 1-sided error testing is only $\Omega(\sqrt{n})$.

Problem 42: Graph Frequency Vectors

Suggested by	Noga Alon
Source	Bertinoro 2011
Short link	https://sublinear.info/42

For a graph G, a k-disc around a vertex v is the subgraph induced by the vertices that are at distance at most k from v. The frequency vector of k-discs of G is a vector indexed by all isomorphism types of k-discs of vertices in G which counts, for each such isomorphism type K, the fraction of k-discs of vertices of G that are isomorphic to K. The following is a known fact observed in a discussion with Lovász. It is proved by a simple compactness argument.

Fact: For every $\epsilon > 0$, there is an $M = M(\epsilon)$ such that for every 3-regular graph G, there exists a 3-regular graph H on at most $M(\epsilon)$ vertices (independent on $|V(G)|$), such that variation distance between the frequency vector of the 100-discs in G and the frequency vector of the 100-discs in H is at most ϵ.

Question: Find *any* explicit estimate on $M(\epsilon)$. Nothing is currently known.

Updates

Partial progress has been made by Fichtenberger et al. [FichtenbergerPS-15], who proved that if all k-discs in G are trees (i.e., G has girth greater than $2k+1$), then $|V(H)| \leq \frac{10^{10^{50}}}{\epsilon}$. The result generalizes to arbitrary degree bound d and $k \geq 0$. H can be constructed in constant time at the cost of roughly an additional factor ϵ^{-1}. It was sketched by the same authors at the Workshop on Algorithms in Communication Complexity, Property Testing and Combinatorics in Moscow in 2016 (http://math.ucsd.edu/~sbuss/SPB_Workshops/AlgCPTC_1.html) that one can also construct H if G is planar by using the planar separator theorem. In this case, $|V(H)| \in O(\epsilon^{-4})$.

Problem 43: Rank Lower Bound

We want to prove that the following tall matrix has full column rank. The columns are indexed by $a_1, \ldots, a_k$ from the field F_{2^n} where n is prime; the rows are indexed by degrees $d_1 \ldots d_r$. The entry in the ith column and jth row is equal to $a_i^{d_j}$.

Suggested by	Madhu Sudan
Source	Bertinoro 2011
Short link	https://sublinear.info/43

Question: Is it true that for all k there exists an r such that for all $d_1, \ldots, d_r$ that are powers of 2 and for all $a_1, \ldots, a_k$ that are linearly independent over F_2, the rank of the matrix is k?

Background: Note that if $d_i = i$ and $r \geq k$, then the matrix is Vandermonde and so has full rank. If $d_i = 2^i$, then also the matrix has full rank (Lemma 19 in [GrigorescuKS-08]). The general case, when d_i's are arbitrary, and not successive powers of two remains open (Conjecture 5.9 in [BenSassonGMSS-11]).

Problem 44: Approximating LIS Length in the Streaming Model

The goal of LIS is to compute a 2-approximation of the length of the longest increasing subsequence in a given stream of elements.

Question: What is the randomized streaming space complexity of LIS, for one pass or possibly a constant number of passes?

Suggested by	Amit Chakrabarti
Source	Bertinoro 2011
Short link	https://sublinear.info/44

Background: Gopalan et al. [GopalanJKK-07] gave an $O(n^{1/2}\operatorname{polylog}(n))$-space *deterministic* streaming algorithm, using one pass, that achieves c-approximation for any fixed $c > 0$. For deterministic algorithms [GalG-07,ErgunJ-08] showed an $\Omega(n^{1/2})$ space lower bound, for a constant number of passes. The latter arguments proceed by proving a lower bound for related communication complexity problems. However, it is known that the randomized communication complexity of those problem is $O(\log n)$ [Chakrabarti-10] so a different approach is needed.

Problem 45: Streaming Max-Cut/Max-CSP

The problem is defined as follows: given a stream of edges of an n-node graph G, estimate the value of the maximum cut in G.

Suggested by	Robert Krauthgamer
Source	Bertinoro 2011
Short link	https://sublinear.info/45

Question: Is there an algorithm with an approximation factor strictly better than $1/2$ that uses $o(n)$ space?

Background: Note that $1/2$ is achievable using random assignment argument. Moreover, using sparsification arguments [Trevisan-09,AhnG-09], one can obtain a better approximation ratio using $O(n \operatorname{polylog} n)$ space. Woodruff and Bhattacharyya (private communication) noted that subsampling $O(n/\epsilon^2)$ edges gives, with high probability, an ϵ-additive approximation for all cuts, and thus $1+\epsilon$ multiplicative approximation for MAX-CUT.

Question: What about general constraint satisfaction problems with fixed clause-length and alphabet-size? In this case it is even not known how to obtain $O(n \operatorname{polylog} n)$ space bound.

Updates

The progress on the MAX-CUT problem in the streaming setting:

- Estimating the maximum cut to within a factor of $(1-\varepsilon)$ requires $n^{1-O(\varepsilon)}$ space [KapralovKS-15,KoganK-15].
- There exists a constant $\varepsilon_* > 0$ such that obtaining a $(1-\varepsilon_*)$ approximation to MAX-CUT requires $\Omega(n)$ space [KapralovKSV-17].
- In random-order streams, $\Omega(\sqrt{n})$ space is needed to obtain a better than $1/2$ approximation [KapralovKS-15].

Problem 46: Fast JL Transform for Sparse Vectors

Suggested by	Jelani Nelson
Source	Bertinoro 2011
Short link	https://sublinear.info/46

Consider a distribution over linear mappings A from R^d to R^k, $k = O(\log(1/P)/\epsilon^2)$. We say that it has an (ϵ, P)-JL property if for any vector $x \in R^d$ we have

$$\|Ax\|_2 = (1 \pm \epsilon)\|x\|_2$$

with probability $1 - P$.

Question: Can we construct a distribution with this property such that the matrix-vector product Ax can be evaluated in time $(s + k) \cdot \mathrm{polylog}(d)$ time given an s-sparse x?

Background: Such an algorithm is not known even for $s = d$ (unless k is larger [AilonL-11], [KrahmerW-11]).

Question: Provide an explicit construction of a distribution with the (ϵ, P)-JL property such that the random matrix A can be generated using $O(\log(d/P))$ bits.

Problem 47: Annotated Streaming

Suggested by	Graham Cormode
Source	Bertinoro 2011
Short link	https://sublinear.info/47

In the annotated stream model [ChakrabartiCM-09], a stream is augmented with 'annotation', which takes the form of a proof of a property of the stream. In its simplest form, the annotation may just be a reordering of the stream to make it easy to compute a function of interest. The key parameters in this model are H, the size of the annotation, and V, the space needed by the streaming party to view the stream and verify the proof. The annotation proposed should be such that an honest annotation will always be accepted, while a mistaken annotation will be detected and rejected with high probability.

We consider the problem of counting the number of triangles in a graph described by a stream of edges (where each edge is promised to occur at most once). Partial results from the above reference are that $H = O(n^2)$ and $V = \tilde{O}(1)$ is possible, as is $H = O(n^{3/2}), V = O(n^{3/2})$.

Question: Can one achieve $H = V = \tilde{O}(n)$?

Update

This question was answered affirmatively by Thaler [Thaler-16].

Problem 48: Sketching Shift Metrics

Suggested by	Alexandr Andoni
Source	Bertinoro 2011
Short link	https://sublinear.info/48

For any $x, y \in \{0,1\}^n$, define the *shift metric*

$$\mathrm{sh}(x,y) = \min_{\sigma} H(x, \sigma(y)),$$

where σ ranges over all n cyclic permutations of $\{1 \ldots n\}$, and $H()$ is the hamming distance.

For any $c > 20$, the promise problem P_c is to distinguish whether $\mathrm{sh}(x,y) > n/10$ or $\mathrm{sh}(x,y) < n/c$. Consider probabilistic mappings $L_c : \{0,1\}^n \to \{0,1\}^s$. We say that L_c is a sketching scheme for P_c if there is an algorithm that, for any $x, y \in \{0,1\}^n$ satisfying the promise of P_c, given $L_c(x)$ and $L_c(y)$, solves P_c with probability at least 0.9.

Question: Is there a sketching scheme for P_c where $c = O(1)$ and $s = O(1)$?

Background: If the shift metric is replaced by Hamming metric, one can achieve $s = O(1)$ using random sampling [KushilevitzOR-00]. The actual problem can be solved for $c = O(\log^2 n)$ and $s = O(1)$ [AndoniIK-08]. The algorithm proceeds by embedding the shift metric into Hamming metrics, and it is known that this step must induce $\Omega(\log n)$ distortion [KhotN-06]. There's also a solution for $c = 1 + \epsilon$ and $s = \tilde{O}(\epsilon^{-2}\sqrt{n})$ [CrouchM-11].

Problem 49: Sketching Earth Mover Distance

Suggested by	Piotr Indyk
Source	Bertinoro 2011
Short link	https://sublinear.info/49

For any two subsets A, B of the planar grid $[n]^2$, $|A| = |B|$, define

$$\mathrm{EMD}(A, B) = \min_{\pi: A \to B} \sum_{a \in A} \|a - \pi(a)\|_1,$$

where π ranges over one-to-one mapping from A to B.

Question: What is the sketching complexity of c-approximating EMD? That is, consider a distribution over mappings L_c that maps subset of $[n]^2$ to $\{0,1\}^s$, such that for any sets A, B with $|A| = |B|$, given $L_c(A), L_c(B)$, one can estimate $\mathrm{EMD}(A, B)$ up to a factor of c, with probability $\geq 2/3$. Is it possible to construct such a distribution for constant c and $s = \mathrm{polylog}\, n$?

Background: It is known that one can achieve $s = O(\log n)$ for $c = O(\log n)$ by embedding EMD into ℓ_1 [IndykT-03,Charikar-02], and $s = n^{O(1/c)}\, \mathrm{polylog}\, n$ for any $c \geq 1$ [AndoniDIW-09].

Problem 50: Sparse Recovery for Tree Models

Suggested by	Piotr Indyk
Source	Bertinoro 2011
Short link	https://sublinear.info/50

For any $n = 2^h - 1$, we can identify the coordinates of a vector $v \in \mathbb{R}^n$ with the nodes of a full binary tree B_h of height h and root 1. We define a *k-sparse tree model* $\mathcal{T}_k$ to be a set of all $T \subset [n]$ of size k which form a connected subtree in B_h rooted at 1.

We want to design an $m \times n$ matrix A such that for any $x \in \mathbb{R}^n$, one can recover from Ax a vector $x^* \in \mathbb{R}^n$ such that:

$$\|x^* - x\|_1 \leq \min_{x' \in \mathbb{R}^n,\ \mathrm{supp}(x') \subset T \text{ for some } T \in \mathcal{T}_k} C\|x' - x\|_1,$$

where $\mathrm{supp}(x')$ is the set of non-zero coefficients of x', and $C > 0$ is a constant.

Question: Is it possible to achieve $m = O(k)$ for some constant $C > 0$?

Background: It is possible to achieve a weaker bound of $m = O(k \log(n/k))$ even if $\mathcal{T}_k$ is replaced by the set of all k-subsets of $[n]$ [CandesRT-06a]. However, since $|\mathcal{T}_k| = \exp(O(k))$, one can expect a better bound for $\mathcal{T}_k$. The best C we know how to achieve for $m = O(k)$ is $O(\sqrt{\log n})$ [IndykP-11] (building on [BaraniukCDH-10]).

Update

Indyk and Razenshteyn [IndykR-13] improved the bound on m to $O(k \log(n/k)/\log\log(n/k))$for constant C. In a follow-up paper, Bah et al. [BahBC-14] presented an *efficient* recovery algorithm for this number of measurements.

Problem 51: “For All” Guarantee for Computationally Bounded Adversaries

Suggested by	Martin Strauss
Source	Dortmund 2012
Short link	https://sublinear.info/51

There are two types of compressed sensing guarantees, illustrated using two players:

- *For all*: Charlie constructs the sensing matrix ϕ, and then Mallory constructs the signal $x = x(\phi)$ as a function of ϕ. The Compressed Sensing question is to recover the approximate signal $\tilde{x}$ from the measurement ϕx. The best guarantee possible is the following ℓ_2/ℓ_1 guarantee:

$$||\tilde{x} - x||_2 \leq \epsilon/\sqrt{k}||x_{\text{opt}} - x||_1.$$

- *For each*: Charlie construct a distribution D over sensing matrices ϕ. Then Mallory constructs a vector $x = x(D)$ dependent on the distribution only. Finally, a sensing matrix ϕ is sampled from the distribution D. The goal is again to recover $\tilde{x}$, with good probability over the choice of ϕ. It turns out a stronger guarantee, termed ℓ_2/ℓ_2, is possible:

$$||\tilde{x} - x||_2 \leq (1+\epsilon)||x_{\text{opt}} - x||_2.$$

In some sense the two “worlds” are incomparable: the first one works for all x but obtains weaker error guarantee, and the second one works for each x with some probability but gets better error guarantee.

Question: How can we get the best of both worlds (“for all” with ℓ_2/ℓ_2 error) ?

Once we require “for all”, it is provably impossible to obtain ℓ_2/ℓ_2 guarantee. But what if Mallory has bounded computational resources to construct a “bad” x?

A preliminary result considers the following setting. Mallory sees ϕ and writes down a sketch of ϕ (in bounded space). Then Mallory produces x from this sketch only. Then ℓ_2/ℓ_2 is possible for such x's.

Generally, we would like to allow Mallory to be probabilistic polynomial time, and have a ϕ so that Mallory still cannot find an input $x = x(\phi)$ that breaks the recovery algorithm.

Problem 52: TSP in the Streaming Model

We have n points living in $\{1, \ldots, \Delta\}^2$.

Suggested by	Christian Sohler
Source	Dortmund 2012
Short link	https://sublinear.info/52

Question: Can we approximate the value of the TSP tour (Traveling Salesman Problem) of the n points when streaming over the points in one pass, using small space ($\log^{O(1)} \Delta$)?

One can achieve a 2-approximation by computing a minimum spanning tree in small space, and use the MST to approximate TSP. The question is whether one can obtain an approximation factor $c < 2$ in polylog space. There are other natural related question, such as computing the Earth-Mover Distance over the points in the stream (see Problem 49).

Problem 53: Homomorphic Hash Functions

Question: Construct a hash function $h : \mathbb{F}_p^n \to \mathbb{F}_p^m$, where $m < n$, satisfying the following properties:

- h is linear: $h(u + v) = h(u) + h(v)$ for all $u, v \in \mathbb{F}_p^n$;
- for any u, v, we have $\Pr_h[h(u) = h(v)] = \frac{c}{p^m}$ for some constant c independent of n, m.

Suggested by	Ely Porat
Source	Dortmund 2012
Short link	https://sublinear.info/53

One solution is by considering a random linear function, given by the matrix M. Then we have that $\Pr_M[Mu = Mv] = \Pr_M[M(u - v) = 0] = 1/p^m$. This function would require $O(nm \log p)$ random bits, and computing h takes $O(nm)$ time. We would like more efficient solutions. Ely and coauthors claim a solution with $O((n + m) \log p)$ bits, and $O((n + m) \log(n + m))$ time. If one considers Reed-Salomon codes, it seems that they would give worse bound on second property (probability of collision).

Problem 54: Faster JL Dimensionality Reduction

Suggested by	Jelani Nelson
Source	Dortmund 2012
Short link	https://sublinear.info/54

The standard Johnson-Lindenstrauss lemma states the following: for any $0 < \epsilon < 1/2$, any $x_1 \ldots x_n \in \mathbb{R}^d$, there exists $A \in \mathbb{R}^{k \times d}$ with $k = O(1/\epsilon^2 \cdot \log n)$, such that for any i, j we have $\|Ax_i - Ax_j\|_2 = (1 \pm \epsilon)\|x_i - x_j\|_2$.

The main question is to construct A's that admit faster computation time of Ax. There are several directions to try to obtain more efficient A:

- Fast JL (FFT-based): Here, the runtime is of the form $O(d \log d + \mathrm{poly}(k))$ to compute Ax ($d \log d$ is usually the most significant term).
- Sparse JL: Here, the runtime is of the form $O(\epsilon k \|x\|_0 + k)$, where $\|x\|_0$ is the number of non-zero coordinates of x (i.e., it works well for sparse vectors).

Question: Can one obtain a JL matrix A such that one can compute Ax in time $\tilde{O}(\|x\|_0 + k)$?

One possible avenue would be by considering a “random” k by k submatrix of the FFT matrix. This may or may not lead to the desired result.

Problem 55: Applications of Clifford Algebras in Graph Streams

Suggested by	He Sun
Source	Dortmund 2012
Short link	https://sublinear.info/55

Some of the recent results in graph streaming algorithms [KaneMSS-12,ManjunathMPS-11] use *complex-valued* sketches to capture the graph structure. While it had been known earlier that integer-valued sketches can be used to count triangles, Kane et al. [KaneMSS-12] developed a complex-valued sketch to count the number of occurrences of an arbitrary subgraph of constant size. These techniques also extend to variations of the subgraph counting problem, for instance counting a directed or (labelled) subgraph. However, the bounds on the space complexity which depends on the variance of the sketches are quite loose for most graph families.

It is interesting to compare these results to the framework of designing randomized algorithms for computing the permanent. Let A be a 0-1 matrix, and B be the matrix obtained from A by replacing each 1 uniformly and randomly with an element from a finite set D. With suitable choices of the set D, the determinant of B can be used to approximate the permanent of A. As shown by Chien et al. [ChienRS-03] and discussed by Muthukrishnan [Muthukrishnan-06], by choosing elements of D from $\mathbb{Z}$, $\mathbb{C}$, or a Clifford algebra, the variance of the estimator drops significantly each time when we move to a more "complex" algebra. It seems plausible that similar techniques can be used to improve the space complexity of graph streaming algorithms which are based on complex-valued random variables.

Question: Find suitable applications of Clifford algebra in designing algorithms in graph streams.

Problem 56: Efficient Measures of "Surprisingness" of Sequences

Suggested by	Rina Panigrahy
Source	Dortmund 2012
Short link	https://sublinear.info/56

Consider a sequence of i.i.d. random bits $S \in \{0,1\}^n$.

Question: Find efficient measures of how surprising/unbelievable S appears to be. (Good heuristic for measuring how probable/improbably a string is.)

For example, if we see $0, 0, 0, \ldots$, we won't believe it is random (i.e., it is surprising.)

One existing measure is the (k^{th}-order) Shannon entropy H_k (H_0 would correspond to taking the entropy of the empirical frequencies of 0s and 1s). However, it fails to say that a string like $(0, 0, \ldots, 0, 1, 1, \ldots, 1)$ is surprising (from the point of view of densities it looks pretty random).

Ideal solution is to consider the Kolmogorov complexity, but it is hard (impossible) to compute.

A particular setting of the strings to consider may be: suppose each bit is generated from a biased independent coin, but the bias of the coin changes (slowly?) over time. Is there a good compression here?

Problem 57: Coding Theory in the Streaming Model

Suggested by	Atri Rudra
Source	Dortmund 2012
Short link	https://sublinear.info/57

Consider the problem of "codeword testing" in the data stream model. In particular, consider a code $C : \Sigma^k \to \Sigma^n$ with distance[1] d. The specific problem is the following:

> The input to the problem is a vector $\mathrm{y} \in \Sigma^n$ and integer parameters $0 \leq \tau_1 < \tau_2 \leq n$. The algorithm has to decide whether
>
> $$\Delta(\mathrm{y}, C) \leq \tau_1 \text{ or } \Delta(\mathrm{y}, C) \geq \tau_2,$$
>
> where $\Delta(\mathrm{y}, C)$ is the Hamming distance of y from the closest codeword in C.

Ideally, we want a one-pass, $\log^{O(1)} n$ space algorithm to solve the problem above for some *good* code C (that is, we have $k \geq \Omega(n)$ and $d \geq \Omega(n)$). Or if we prove a hardness result, one would like a hardness result for *every* good code C. (For the sake of simplicity, assume that the algorithm has access to some succinct description of the code C.)

The main technical motivation comes from the case when $\tau_1 = 0$ and $\tau_2 \geq \epsilon n$ for any fixed $\epsilon > 0$ but with *constant* number of queries to y (i.e. in the property testing world). This question is perhaps the open question in the codeword testing literature. The case of $\tau_1 > 0$ also makes sense in the property testing world and has been studied [GuruswamiR-05]. (See the paper for some potential practical motivations.)

One of the original motivation (in [RudraU-10]) for the study of the data-streaming version of the question was possibly to use communication complexity results to prove the impossibility of good locally testable codes.

It was shown in [RudraU-10] that for the well-known Reed-Solomon codes, the data stream version of the problem can be solved for $\tau_1 = 0$ and $\tau_2 = 1$ with one pass and logarithmic space. It can also be shown that the classical Berlekamp-Massey algorithm for decoding Reed-Solomon codes implies a solution for the case $\tau_2 = \tau_1 + 1$ with one pass and space $\tilde{O}(\tau_1)$[2]. Finally, [McGregorRU-11] showed how to solve this problem in one pass and $O(k \log n)$ space. This question is wide open:

> Solve the problem above with one pass and $\tilde{O}(\min(k, \tau_1))$ space.

In fact the very special case of the problem above for $k = \tau_1 = \sqrt{n}$ with one pass and space $o(\sqrt{n})$ is also open. This is open even for the special case of Reed-Solomon codes.

Notes

1. The distance of a code C is the minimum Hamming distance between any two codewords, i.e., $\min_{\mathrm{x} \neq \mathrm{y} \in \Sigma^k} |\{i \in [n] | C(\mathrm{x})_i \neq C(\mathrm{y})_i\}|$
2. There is a small catch: the algorithm actually computes the location of errors *if* the number of errors is at most τ_1. However, results in [RudraU-10] can be used to verify if the returned error locations are indeed correct.

Problem 58: Signatures for Set Equality

Suggested by	Rasmus Pagh
Source	Dortmund 2012
Short link	https://sublinear.info/58

Given $S \subseteq \{1,..n\}$, we would like to construct a fingerprint so that later, given fingerprints of two sets, we can check the equality of the two sets. There are (at least) two possible solutions to the problem:

1. $h(S) = \left(\sum_{i \in S} x^i\right) \bmod p$ for random $x \in \mathbb{Z}_p$. Update time would be roughly $\log p = \Omega(\log n)$. One would like to obtain a better update time.
2. $h(S) = \left(\prod_{i \in S} (x-i)\right) \bmod p$, and random x. Insertion can be done in constant time. But the fingerprint is not linear.

Question: Can we construct a fingerprint that achieves constant update time and is linear, while using $O(\log n)$ random bits? Ideally updates would include insertions and deletions. Linearity would imply, for example, that if $S_1 \subseteq S_2$ we can compute $h(S_2 \backslash S_1)$ in constant time, as the difference of $h(S_2)$ and $h(S_1)$.

Problem 59: Low Expansion Encoding of Edit Distance

Suggested by	Hossein Jowhari
Source	Dortmund 2012
Short link	https://sublinear.info/59

Let $T = \bigcup_{i=1}^{n} \{0,1\}^i$. For a pair of strings $(x, y) \in T \times T$, let $\mathrm{ed}(x, y)$ denote the edit distance between x and y, which is defined as the minimum number of character insertion, deletion, and substitution needed for converting x into y.

Question: Is there a mapping $f : T \to \{0,1\}^m$ satisfying the following conditions

- f is injective, i.e. it does not map different inputs to the same point.
- $m = O(n^c)$ for some constant $c \geq 1$.
- For strings with $\mathrm{ed}(x, y) = 1$ we have $\mathcal{H}(f(x), f(y)) \leq C$ for $C = o(\log n)$.

The same question holds for randomized mappings as long as they map different x and y to different points with high probability. Currently the best upper bound on C is $O(\log n \log^* n)$ achieved through a randomized mapping that deploys the Locally Consistent Parsing method [CormodePSV-00]. For non-repetitive strings (the Ulam distance) there is a deterministic mapping with $C \leq 6$ and $c = 2$. Preferably we would like to have mappings that are efficiently computable and are equipped with polynomial time decoding algorithms (x can be obtained from $f(x)$ efficiently). See [Jowhari-12] for motivations on the problem.

Problem 60: Single-Pass Unweighted Matchings

Suggested by	Andrew McGregor
Source	Dortmund 2012
Short link	https://sublinear.info/60

Suppose you have $O(n \operatorname{polylog} n)$ memory and a single pass over a stream of m edges (arbitrarily ordered) on n nodes. How well can you approximate the size of the maximum cardinality matching? A trivial greedy algorithm finds a $1/2$-approximation but that's still the best known algorithm in the general setting. Kapralov [Kapralov-12] showed that achieving better than a $1 - 1/e$ approximation is impossible. If the stream is randomly ordered, Konrad et al. [KonradMM-12] presented a $1/2 + 0.005$-approximation. Other variants of the question are also open, e.g., achieving a $(1 - \epsilon)$ approximation in the minimum number of passes (see, e.g., Ahn and Guha [AhnG-11]) or the best approximation possible for maximum weighted matching in a single pass (see, e.g., Epstein et al. [EpsteinLMS-11]).

Problem 61: RNA Folding

Suggested by	Qin Zhang
Source	Bertinoro 2014
Short link	https://sublinear.info/61

An RNA sequence is a string of letters from the alphabet $\{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{U}\}$, where A---U and C---G form pairings. A set of pairings in such a string is said to be non-crossing if there are no pairs of the form (i, j) and (k, l), where $i < k < j < l$.

A *maximum non-crossing matching* is a pairing of A---U and C---G of maximum cardinality that is non-crossing. Given a string of length n, such a matching can be computed in $O(n^2)$ space and $O(n^3)$ time via dynamic programming [Eddy-04].

Note that there is a trivial 2-approximation to the optimal matching. Find the optimal matchings on the $\{\mathrm{A}, \mathrm{U}\}$ and $\{\mathrm{C}, \mathrm{G}\}$ subsequences, and take the larger one. In particular, this implies that a 2-approximation to the *size* of the optimal non-crossing matching can be computed in $O(1)$ space.

How well can the size of the optimal non-crossing matching be approximated in $\mathrm{polylog}(n)$ space and a small number of passes?

(Additional question from Alex Andoni: Can the problem be solved in the RAM model with running time better than the trivial dynamic programming?)

Problem 62: Principal Component Analysis with Nonnegativity Constraints

Suggested by	Andrea Montanari
Source	Bertinoro 2014
Short link	https://sublinear.info/62

Given a symmetric matrix A, we can think of Principal Component Analysis (PCA) as maximizing $x^\top Ax$ subject to $\|x\| = 1$. If we also add the condition $x \geq 0$, this problem becomes NP-hard. We can define a convex relaxation:

$$\max \operatorname{Tr}(WA) \quad \text{s.t.} \quad \operatorname{Tr}(W) = 1,\ W \geq 0,\ W \succeq 0.$$

Suppose that A is a random matrix. In particular, set A_{ij} to be i.i.d $N(0,1)$. Then empirical results show that the resulting W is a rank-1 matrix, which means that we recover the optimal x exactly.

Is this true in general? Note that we can prove that the solution is rank 1 if $A = vv^T + (\text{small amount of noise})$

Problem 63: Submodular Matching Maximization

Suggested by	Amit Chakrabarti
Source	Bertinoro 2014
Short link	https://sublinear.info/63

Let $G = (V, E)$ be a graph. Fix a monotone submodular function $f : 2^E \to \mathbb{R}$. A matching $M \subseteq E$ is called a *maximum submodular matching* (MSM) with respect to f if it maximizes $f(E)$. This generalizes maximum weight matching (MWM). Suppose the graph edges are streaming and we are allowed only one pass. It is known that using $O(n \log n)$ space we can approximate MWM within a factor of $4 + \epsilon$ [CrouchS-14] and MSM (for any f) within 7.75 [ChakrabartiK-14]. It is also known that we cannot approximate MWM to a factor better than $\frac{e}{e-1}$ using $n \operatorname{polylog}(n)$ space [Kapralov-12].

Can we show a stronger lower bound for maximum *submodular* matchings? A conjecture is that it will be hard to get a better than 2-approximation in one pass with the same space constraints.

A related question (due to Deeparnab Chakrabarty): Is there an instance-wise gap between MWMs and MSMs in the stream setting, for some choice of submodular f and with the MWM instance being derived by evaluating f at singleton sets?

Problem 64: Matchings in the Turnstile Model

Consider an unweighted graph on n nodes defined by a stream of edge insertions and deletions. Is it possible to approximate the size of the maximum cardinality matching up to constant factor given a single pass and $o(n^2)$ space? Recall that a factor 2 approximation is easy in $O(n \log n)$ space if there are no edge deletions.

Suggested by	Andrew McGregor
Source	Bertinoro 2014
Short link	https://sublinear.info/64

Updates

The question is fully settled when the goal is to output the edges of an approximate maximum matching: to obtain an α-approximation to maximum matching in dynamic streams, $\Omega(n^2/\alpha^3)$ space is necessary [AssadiKLY-16] and $\widetilde{O}(n^2/\alpha^3)$ space is sufficient [AssadiKLY-16,ChitnisCEHMMV-16]. When the goal is only to estimate the value of maximum matching (as opposed to finding the edges), $\Omega(n/\alpha^2)$ space is necessary and $\widetilde{O}(n^2/\alpha^4)$ space is sufficient [AssadiKL-17].

Problem 65: Communication Complexity of Connectivity

Suggested by	Andrew McGregor
Source	Bertinoro 2014
Short link	https://sublinear.info/65

A recent result in graph sketching [AhnGM-12] can be rephrased in terms of a simultaneous message communication protocol with public coins. Specifically, suppose that n players are each given a row of the adjacency matrix of some graph. The players simultaneously send a message to a central player who must then determine whether the graph is connected. Existing work shows that this is possible with $O(\log^3 n)$ bit messages from each player. Are $O(\log^2 n)$ or $O(\log n)$ bits sufficient? Also, is there a non-trivial lower bound if the players must use private coins?

Problem 66: Distinguishing Distributions with Conditional Samples

Suggested by	Eldar Fischer
Source	Bertinoro 2014
Short link	https://sublinear.info/66

Suppose we are given access to two distributions P and Q over $\{1, 2, \ldots, n\}$ and wish to test if they are the same or are at least ϵ apart under the ℓ_1 distance. Assume that we have access to *conditional samples*: a query consists of a set $S \subseteq \{1, 2, \ldots, n\}$ and the output is a sample drawn from the conditional distribution on S [ChakrabortyFGM-13,CanonneRS-14]. In other words, if p_j is the probability of drawing an element j from P, a conditional sample from P restricted to S is drawn from the distribution where

$$\Pr(j) = \begin{cases} \frac{p_j}{\sum_{i \in S} p_i} & \text{if } j \in S, \\ 0 & \text{otherwise.} \end{cases}$$

It is known that if one of the distributions is fixed, then the testing problem requires at most $\tilde{O}(1/\epsilon^4)$ queries, which is independent of n [CanonneRS-14].

What can we say if both distributions are unknown? The best known upper bound is $\tilde{O}\left(\frac{\log^5 n}{\epsilon^4}\right)$ [CanonneRS-14].

Updates

Acharya, Canonne, and Kamath [AcharyaCK-14] showed that $\Omega(\sqrt{\log \log n})$ conditional queries are needed in this case for some constant $\epsilon > 0$. Contrary to the case of only one distribution unknown, if both distributions are unknown, the required number of queries is a function of n. Falahatgar, Jafarpour, Orlitsky, Pichapathi, and Suresh [FalahatgarJOPS-15] showed that $O\left(\frac{\log \log n}{\epsilon^5}\right)$ queries are sufficient. This determines the query complexity of the problem up to a factor of $\sqrt{\log \log n}$.

Problem 67: Difficult Instance for Max-Cut in the Streaming Model

Suggested by	Robert Krauthgamer
Source	Bertinoro 2014
Short link	https://sublinear.info/67

We are interested in **Max-Cut** in the streaming model, and specifically in the tradeoff between approximation and space (storage) complexity. Formally, in the **Max-Cut** problem, the input is a graph G, and the goal is to compute the maximum number of edges that cross any single cut (V_1, V_2). This is clearly equivalent to computing the least number of edges that need to be removed to make the graph bipartite. In the streaming model, we assume that the input graph is seen as a sequence of edges, in an arbitrary order, and the goal is to compute the **Max-Cut** value, i.e., the number of edges (or approximate it). There is no need to report the cut itself. For instance, it is easy to approximate **Max-Cut** within factor $1/2$ using $O(\log n)$ space, by simply counting the total number edges in the input and reporting $|E(G)|/2$.

Here is a concrete suggestion for a hard input distribution, which is known to be a hard instance for bipartiteness testing in sparse graphs [GoldreichR-02]. Let G' be a graph consisting of a cycle of length n (where n is even) and a random matching. It is known that with high probability, G' is an expander and at least $0.01n$ edges have to be removed to turn it into a bipartite graph. Let G'' be a graph consisting of a cycle of length n and a random matching, with the constraint that the matching must consist only of *odd chords*: these are chords that are an odd number of vertices apart on the cycle. It is easy to see that G'' is always bipartite.

The total number of edges in both G' and G'' is exactly $3n/2$. It is easy to see that

- G'' has a cut of size $3n/2$,
- with high probability, G' has no cut of size greater than $(3/2 - 0.01)n$

How much space is required to distinguish between these two graphs in the streaming model? Is it $\Omega(n)$? And what about the (multi-round) communication complexity of the problem, namely, the edges of the input graph are split between two parties, Alice and Bob, who need to estimate the **Max-Cut**?

Updates

The progress on the complexity of **Max-Cut** is described in updates on Problem 45.

Problem 68: Approximating Rank in the Bounded-Degree Model

Suggested by	Yuichi Yoshida
Source	Bertinoro 2014
Short link	https://sublinear.info/68

Let $A : \mathbb{F}_p^{m \times n}$ be a matrix such that each row and column has a constant number of non-zero entries (hence, $m = O(n)$). The matrix A can be accessed via the following types of queries. If we specify the i-th row, then we obtain the indices j for which $A_{i,j} \neq 0$. Similarly, if we specify the j-th column, then we obtain the indices i for which $A_{i,j} \neq 0$. For a parameter $\epsilon > 0$, we want to approximate the rank of A to within $\pm \epsilon n$. How many queries are needed to accomplish this task?

When $p = 2$ and each row has exactly two ones, A can be seen as the incidence matrix of a graph, and its rank is equal to $n - c$, where c is the number of connected components. In this case, the rank can be approximated efficiently, with $\tilde{O}(1/\epsilon^2)$ queries, because we know how to efficiently approximate c [ChazelleRT-05].

In general, we conjecture that $\Omega(n)$ queries are necessary. The difficulty in showing this lower bound arises from the fact that few techniques for proving $\Omega(n)$ lower bounds for the bounded-degree model are known. Bogdanov, Obata, and Trevisan [BogdanovOT-02] show a lower bound of $\Omega(n)$ for the problem of testing the satisfiability of **E3LIN-2** instances in the bounded-degree model. However, the lower bound is obtained by considering a distribution of instances of the form $Ax = b$, where A is fixed and b is random. Hence, we cannot directly apply the construction to the rank problem as we only have A.

Problem 69: Correcting Independence of Distributions

Suggested by	Clément Canonne
Source	Baltimore 2016
Short link	https://sublinear.info/69

Let p be an unknown (discrete) probability distribution over product space $[n] \times [n]$, and $\varepsilon \in (0,1]$. Suppose p is ε-*close to independent* (in total variation distance), i.e., there exists a product distribution $q = q_1 \times q_2$ on $[n] \times [n]$ such that

$$d_{\mathrm{TV}}(p,q) = \max_{S \subseteq [n]\times[n]} (p(S) - q(S)) \leq \varepsilon.$$

Given access to independent samples drawn from p, the goal is to *correct* p, that is, to provide access to independent samples from a distribution $\tilde{p}$ satisfying (with high probability):

- $d_{\mathrm{TV}}(p,\tilde{p}) = O(\varepsilon)$ (i.e., the corrected distribution is faithful to the original one),
- $\tilde{p} = \tilde{p}_1 \times \tilde{p}_2$ (i.e., the corrected distribution is a product distribution),

with a *rate* as good as possible, where the rate is the number of samples from p required to provide a single sample from $\tilde{p}$.

Achieving a rate of 2 is simple: drawing (x_1, y_1) and (x_2, y_2) from p and outputting (x_1, y_2) provides sample access to $\tilde{p} = p_1 \times p_2$, which can be shown to be 3ε-close to p [BatuFFKRW-01].

Question 1: Is a rate $r < 2$ achievable? What about an amortized rate (to provide $q = o(n^2)$ samples from the same distribution $\tilde{p}$[1])?

Question 2: What about the same question, when relaxing the second item to only ask that p be *improved*: that is, to provide sample access to a distribution $\tilde{p}$ guaranteed to be $\frac{\varepsilon}{2}$-close to a product distribution?

Note: This question fits within the framework of "sampling improvers," introduced in [CanonneGR-16]. In this framework, given access to a probability distribution only close to having a desired property, one aims at providing access to corrected samples from a nearby distribution that exhibits this property.

1. The restriction $o(n^2)$ comes from the fact that, after n^2 samples, one can actually learn the distribution p, and then compute a good corrected definition $\tilde{p}$ offline. Hence, the range of interest is when having to provide a number of samples negligible in front of what learning p would require.

Problem 70: Open Problems in L_p-Testing

Suggested by	Grigory Yaroslavtsev
Source	Baltimore 2016
Short link	https://sublinear.info/70

Extending the usual setting of property testing to functions $f: \{1, \ldots, n\}^d \to [0, 1]$, Berman et al. [BermanRY-14] study property testing with regard to L_p distances between functions. Namely, these distances are defined as $\mathrm{dist}_p(f, g) = \frac{\|f-g\|_p}{\|\mathbf{1}\|_p}$ (for $p > 0$), so that for instance $\mathrm{dist}_1(f, g) = \frac{\|f-g\|_1}{n^d}$ (and if the functions are Boolean, we get back the Hamming distance).

Let P be a class of functions (e.g. monotone, convex, Lipschitz, etc.) A non-tolerant L_1 property tester has to distinguish functions f that have a property P from those that are ϵ-far, i.e. $\inf_{g \in P} \mathrm{dist}_1(f, g) \geq \epsilon$. A tolerant L_1 property tester has to distinguish functions f that are ϵ_1-close to a property P ($\inf_{g \in P} \mathrm{dist}_1(f, g) \leq \epsilon_1$) from those that are ϵ_2-far ($\inf_{g \in P} \mathrm{dist}_1(f, g) \geq \epsilon_2$).

- **Problem 1:** [BermanRY-14] describe a non-tolerant L_1-tester for convexity whose query complexity, $O(\frac{1}{\varepsilon^{d/2}} + \frac{1}{\varepsilon})$, grows exponentially with the dimension d. Is this exponential dependence necessary, or is there a tester with query complexity $O(\frac{1}{\varepsilon^{o(d)}})$?
- **Problem 2:** Obtain a tolerant L_1 tester for monotonicity for $d \geq 3$. (There exist testers, albeit maybe non-optimal, in the case $d = 1$ or $d = 2$, from [BermanRY-14]; nothing non-trivial is known for higher dimensions.)

Note: Slides describing the setting and open problems can be found on Grigory's webpage (http://grigory.us/#lp-testing). Slides of a longer talk are available here (http://grigory.us/files/talks/BRY-STOC14.pdf).

Problem 71: Metric TSP Cost Approximation

Suggested by	Krzysztof Onak
Source	Baltimore 2016
Short link	https://sublinear.info/71

This problem appeared in the paper of Czumaj and Sohler [CzumajS-09], who gave an efficient algorithm for approximating the weight of the minimum spanning tree of a metric space in the following setting. The input is a metric space $\mathcal{M}$ on n points. For any pair of points in $\mathcal{M}$, the algorithm is allowed to query their distance. The algorithm of Czumaj and Sohler computes a multiplicative $(1+\epsilon)$-approximation to the *weight* of the minimum spanning tree in time $\tilde{O}(n) \cdot \mathrm{poly}(1/\epsilon)$. This immediately yields a $(2+\epsilon)$-approximation to the length of the minimum travelling salesman tour.

Question: Can one compute a $(2-\delta)$-approximation to the length of the minimum travelling salesman tour with $o(n^2)$ queries or—even better—in $o(n^2)$ time, for a positive absolute constant δ?

Notes:

- It is difficult to apply the approach known from the Christofides algorithm, which computes a $3/2$-approximation in the more standard setting. It requires computing a minimum spanning tree, while Czumaj and Sohler show that obtaining a constant-factor approximation to the minimum spanning tree (*not just its weight*) requires $\Omega(n^2)$ queries.
- For the specific case of $(1,2)$-metrics (i.e., when all distances between points are either 1 or 2), there is an $\tilde{O}(n) \cdot \mathrm{poly}(1/\epsilon)$-time $(1.75+\epsilon)$-approximation algorithm. It uses the fact that if there is a short travelling salesman tour, then there must be a large matching in the graph obtained by connecting pairs of points at distance 1. Therefore, the cases of long and short shortest tours can be distinguished efficiently using a known graph matching algorithm [OnakRRR-12]. (This is a joint observation with Anupam Gupta.) However, this approach does not seem to generalize to arbitrary metrics.

Problem 72: Communication Complexity of Approximating Set-Intersection Join

For $\varepsilon \in (0,1]$ and $n \geq 1$, consider the following communication complexity problem $\mathrm{SIJ}_{n,\varepsilon}$: Alice and Bob are given matrices $A, B \in \{0,1\}^{n \times n}$, respectively, and wish to output a $(1+\varepsilon)$-approximation to the number of non-zero entries in the product $C = AB$. What is the two-way randomized communication complexity $R_\delta(\mathrm{SIJ}_{n,\varepsilon})$ (where δ is the probability of error)?

Suggested by	Qin Zhang
Source	Baltimore 2016
Short link	https://sublinear.info/72

Known facts [GuchtWWZ-15]:

- $R^{\rightarrow}_{1/n}(\mathrm{SIJ}_{n,\varepsilon}) = \tilde{O}(\frac{n}{\varepsilon^2})$ (one-way communication),
- $R_\delta(\mathrm{SIJ}_{n,\varepsilon}) = \Omega(\frac{n}{\varepsilon^{2/3}})$ for some absolute constant $\delta > 0$.

What is the right dependence on ε?

Note: SIJ stands for “Set-Intersection Join,” which is the motivation for this question.

Problem 73: Streaming Online Algorithms

A tad more open-ended than the usual open problems from this website, this one arises from the following observation. While streaming algorithms are often motivated by the constraints due to “big data,” in practice this data is not *static*: however, most streaming algorithms use the structure and regularities encountered in the stream to compute solutions or results only available at the end of the streaming, in hindsight.

Suggested by	Edo Liberty
Source	Baltimore 2016
Short link	https://sublinear.info/73

In practice, however, one often needs to be *online* as well: that is, to maintain a “good as of now” solution as the stream goes, and to combine the usual memory constraints of the streaming setting with the objectives of the standard online analysis setting. (Note that, in contrast to the majority of the machine learning literature, one cannot in general make the assumption of stochasticity; the performance thus has to be compared in the adversarial setting, or an intermediate, better suited model has to be developed.)

Question: Which model can capture this real-world need for a combination of streaming and online competitivity constraints, and what algorithms can then one obtain in this model?

Note: For some work in this direction, one can consult [LibertySS-16].

Problem 74: Succinct Representation for Functions on Graphs

Suggested by	Robert Krauthgamer
Source	Baltimore 2016
Short link	https://sublinear.info/74

Suppose we want to design a data structure that stores, for a given edge-weighted (undirected) graph $G = (V, E_G, w_G)$, the values of the minimum st-cuts for all $s, t \in V$. A naive method is to construct a table containing the value for each pair, requiring $O(|V|^2)$ space (machine words). Alternatively, one may construct a Gomory–Hu tree [GomoryH-61]. This is a tree $T = (V, E_T, w_T)$, in which the minimum st-cut values are equal to those in G. Since T is a tree, it requires only $O(|V|)$ space.

Thus for this problem, a very space-efficient data structure (perhaps even the best one) is itself a graph G', and it encodes the desired values in a natural manner, just compute the same function (min st-cut) on G'. But is this the case for all such functions on graphs, or is there a (natural) case where a potentially complicated data structure outperforms a graphical encoding? The question applies both to exact and approximate computations of the function.

Some relevant examples:

- A randomized data structure for $(1+\varepsilon)$-approximating any cut value (or Rayleigh quotient) in G, which is more space-efficient than the known graphical representation, can be found in [AndoniCKQWZ-16].
- For a given graph G with k terminals, the exact values of all the minimum cuts between subsets of terminals can be stored simply as a list of 2^k numbers. The best graphical representation known is of size roughly 2^{2^k} [HagerupKNR-98,KhanR-14].
- A deterministic data structure for $(1+\varepsilon)$-approximating any multicommodity flow on a set of k given terminals in G. There is no known graphical representation for this, whose size depends on k and ε, but not on G [AndoniGK-14].

Problem 75: Data Structure Lower Bound in the Cell Probe Model

Suggested by	Kasper Green Larsen
Source	Banff 2017
Short link	https://sublinear.info/75

Input is an $n \times n$ boolean matrix M. We can preprocess M and store a data structure. Then on query v, an n bit vector, we need to output Mv, which is matrix multiplication with $\cdot$ replaced by $\wedge$ and $+$ replaced by $\vee$. The preprocessing time is denoted by t_p and query time is denoted by t_q.

It is conjectured that in the word-RAM model, $t_p + nt_q \geq n^{3-o(1)}$. But in the cell-probe model, Larsen and Williams [LarsenW-17] give a data structure that uses space $n^2 + n^{7/4}\sqrt{w}$, i.e., just $n^{7/4}\sqrt{w}$ extra bits, where w is the word size (which is typically $\Theta(\log n)$). The data structure computes Mv using $t_q = n^{7/4}/\sqrt{w}$ probes in the worst case. Such a data structure that stores only a small amount of extra bits is called a *succinct* data structure.

Question: Can we show a lower bound of $\omega(n)$ on t_q in the cell-probe model for succinct data structures?

Problem 76: External Information and Amortized Expected Communication

Suggested by	Mark Braverman
Source	Banff 2017
Short link	https://sublinear.info/76

For a function $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, distribution μ on inputs $\{0,1\}^n \times \{0,1\}^n$, where Alice's and Bob's inputs are random variables X and Y, respectively, the external information complexity for two-player zero-error protocols is defined as

$$\mathrm{IC}^{\mathrm{ext}}(F, 0, \mu) := \inf_{\Pi \text{ that solve } F \text{ correctly always}} I_\mu(\Pi; XY) \,.$$

We denote by $\overline{\mathrm{CC}}(F^n, 0, \mu^n)$ the expected communication complexity of F^n with respect to the distribution μ^n for zero-error protocols.

Either prove or disprove that

$$\mathrm{IC}^{\mathrm{ext}}(F, 0, \mu) = \lim_{n \to \infty} \frac{\overline{\mathrm{CC}}(F^n, 0, \mu^n)}{n} \,.$$

Problem 77: Frontiers in Structural Communication Complexity

Suggested by	Justin Thaler
Source	Banff 2017
Short link	https://sublinear.info/77

In the $\mathbf{UPP}$ communication model, two parties execute a (private coin) randomized communication protocol, and must output the correct answer with probability strictly greater than 1/2. Forster [Forster-01] proved a linear lower bound on the $\mathbf{UPP}$ communication complexity of Inner Product Mod 2. $\mathbf{UPP}$ is essentially the most powerful two-party communication model against which we know how to prove lower bounds.[1]

The (informal) open question is to prove a superlogarithmic lower bound for any natural communication complexity class that can compute problems outside of $\mathbf{UPP}$.

Here are two interesting candidate communication classes (both of these classes are subsets of $\mathbf{AM}$, a well-known frontier class in communication complexity).

1. The communication analog of non-interactive statistical zero knowledge proofs ($\mathbf{NISZK}$). This model can be defined as follows. For a given function $f(x,y) \rightarrow \{0,1\}$, Alice and Bob engage in a (private coin) randomized communication protocol in which they exchange at most k bits, at the end of which Bob outputs a string in $\{0,1\}^k$. If $f(x,y) = 1$ (respectively, $f(x,y) = 0$), then the distribution of Bob's output string must have statistical distance at most $1/3$ (respectively, at least $2/3$) from uniform. The cost of the protocol is k.
2. The communication complexity class $\mathbf{OIP}_+^{[2]}$, which is the two-party communication analog of 2-message streaming interactive proofs [ChakrabartiCMTV-15]. This model involves three parties: Alice, Bob, and Merlin. Alice knows x, Bob knows y, and Merlin knows both x and y. At the start of the protocol, Alice sends a randomized message to Bob (not seen by Merlin). Then Bob sends a message to Merlin, who sends a single message back to Bob. Bob then outputs 0 or 1. If $f(x,y) = 1$, then there must be a Merlin strategy that convinces Bob to output 1 with probability at least 2/3. If $f(x,y) = 0$, then for every Merlin strategy, Bob must output 0 with probability at least $2/3$. The cost of the protocol is the sum of the length of all three messages sent (Alice to Bob, Bob to Merlin, Merlin to Bob).

[BoulandCHTV-16] showed that both $\mathbf{NISZK}$ and $\mathbf{OIP}_+^{[2]}$ can, with logarithmic cost, compute (promise) problems outside of $\mathbf{UPP}$. So the open question is to exhibit an explicit function, such as **Disjointness** or **Inner-Product-Mod-2**, that cannot be computed by logarithmic cost $\mathbf{NISZK}$ or $\mathbf{OIP}_+^{[2]}$ protocols. As far as we know, this is open even for 1-way $\mathbf{NISZK}$, in which Alice sends a single message to Bob of length k, after which Bob outputs a string in $\{0,1\}^k$ that must be either close or far from uniform depending on whether $f(x,y) = 1$. (This model is also a special case of $\mathbf{OIP}_+^{[2]}$.) Even this model can, with logarithmic cost, compute functions outside $\mathbf{UPP}$, yet as far as we know it is possible that **Index** does not have a logarithmic cost protocol in this model.

Notes

1. Let us ignore the example of **Parity-P**, which can compute **Inner-Product-Mod-2** with constant communication, yet a linear lower bound on the **Parity-P** communication complexity of **Equality** follows from a matrix rank argument.

Problem 78: Linear Sketching Over F_2

Suggested by	Grigory Yaroslavtsev
Source	Banff 2017
Short link	https://sublinear.info/78

For a function $f : \{0,1\}^n \to \{0,1\}$, we define its deterministic linear sketch complexity $D^{\text{lin}}(f)$ as a the smallest number k such that there exist k sets $S_1, \ldots, S_k \subseteq [n]$ such that for any $x \in \{0,1\}^n$, we can compute $f(x)$ using $\sum_{i \in S_1} x_i, \ldots, \sum_{i \in S_k} x_i$, where the sum is mod 2. For randomized linear sketch complexity, which is denoted by $R^{\text{lin}}(f)$, the k sets are chosen in advance from a joint distribution and are available for recovering $f(x)$. Please see the paper by Kannan, Mossel, and Yaroslavtsev [KannanMY-16] for more details.

Given f, we also define $f^+ : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$as $f^+(x,y) = f(x \oplus y)$ for all $x, y \in \{0,1\}^n$, where $\oplus$ denotes bitwise XOR. It is known that $D^{\text{lin}}(f) = D^{\to}(f^+)$ [MontanaroO-09], where $D^{\to}$ denotes one-way communication complexity (Alice sends one message to Bob).

Prove (or disprove) the following conjecture: $R^{\text{lin}}(f) = \tilde{\Theta}(R^{\to}(f^+))$.

Problem 79: Cryptogenography

Suggested by	Joshua Brody
Source	Banff 2017
Short link	https://sublinear.info/79

Cryptogenography, introduced by Brody et al. [BrodyJSW-14], is concerned with the following question: “How to share a secret without revealing the secret owner?” In this problem, there are k players and an eavesdropper. Input is a random bit $X \in \{0,1\}$, also called “the secret.” The secret owner J is chosen uniformly at random from $[k]$. Players have private randomness and they can communicate publicly on a shared blackboard visible to everyone. Players are said to win if

- everyone learns the secret, and
- eavesdropper does not guess the secret owner correctly.

We are interested in maximizing the probability that players win (maximum success probability). For $k = 2$, the following trivial protocol has success probability 0.25. Just output 1. It can be shown easily that maximum success probability is at most 0.5. The bounds can be improved to $0.3384 \leq$maximum success probability ≤ 0.361. Here is a protocol that achieves success probability $1/3$.

First round:

If Alice has the secret,

- with probability $2/3$, she decides that Bob speaks in the second round
- with probability $1/3$, she speaks in the second round.

Else (if she does not have the secret)

- with probability $1/3$, she decides that Bob speaks in the second round
- with probability $2/3$, she speaks in the second round.

Second round:

If the speaker has the secret, announce it. Otherwise, announce a random bit.

For large k, the following bounds can be shown: $0.5644 \leq$maximum success probability ≤ 0.75. Can we improve these bounds? For more information, including the state of the art bounds, see the papers of Jakobsen [Jakobsen-14] and Doerr and Kunnemann [DoerrK-16].

Problem 80: Merlin–Arthur Communication Complexity of Connectivity

Suggested by	Amit Chakrabarti
Source	Banff 2017
Short link	https://sublinear.info/80

We have a function $f : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$. In the Merlin–Arthur communication model, Alice gets an $x \in \mathcal{X}$ and Bob gets a $y \in \mathcal{Y}$. Merlin is an all-knowing, all-powerful entity who sends them a proof at the beginning. Then Alice and Bob communicate to find $f(x,y)$. A protocol Π solves f if, for all x, y,

- $f(x,y) = 1 \implies \exists \text{ proof} : \Pr[\Pi(x,y,\text{ proof}) = 1] \geq 2/3$ and
- $f(x,y) = 0 \implies \forall \text{ proofs} : \Pr[\Pi(x,y,\text{ proof}) = 1] \leq 1/3$

We denote the communication complexity of f in the above model as $\mathrm{MA}^{\to}(f)$. The communication cost here does not include the proof size. It is known that $\mathrm{MA}^{\to}(\mathbf{Disj}) = \tilde{O}(\sqrt{n})$ [AaronsonW-09] and $\mathrm{MA}^{\to}(\mathbf{InnerProd}) = \tilde{O}(\sqrt{n})$.

For $x, y \in \{0,1\}^{\binom{n}{2}}$, interpreting x and y as edges of an n vertex graph, define $\mathbf{Connect}$ as follows. If $x \cup y$ is connected, $\mathbf{Connect}(x,y) = 1$, else $\mathbf{Connect}(x,y) = 0$. Using the Ahn-Guha-McGregor [AhnGM-12] linear sketch for connectivity, we can show that $D^{\to}(\mathbf{Connect}) = \tilde{O}(n)$, where $D^{\to}$ denotes one-way communication complexity (Alice sends one message to Bob, and there is no Merlin).

Is $\mathrm{MA}^{\to}(\mathbf{Connect}) = o(n)$?

Bibliography

[AaronsonW-09]
Scott Aaronson and Avi Wigderson. *Algebrization: A New Barrier in Complexity Theory.* ACM Transactions on Computation Theory, 1(1), 2009.

[AcharyaCK-14]
Jayadev Acharya, Clément Canonne, and Gautam Kamath. *A Chasm Between Identity and Equivalence Testing with Conditional Queries.* In *CoRR*, abs/1411.7346, 2014.

[AhnG-09]
Kook Jin Ahn and Sudipto Guha. *Graph sparsification in the semi-streaming model.* In *International Colloquium on Automata, Languages and Programming*, pages 328-338, 2009.

[AhnG-11]
Kook Jin Ahn and Sudipto Guha. *Laminar Families and Metric Embeddings: Non-bipartite Maximum Matching Problem in the Semi-Streaming Model.* In *CoRR*, abs/1104.4058, 2011.

[AhnGM-12]
Kook Jin Ahn, Sudipto Guha, and Andrew McGregor. *Analyzing graph structure via linear measurements.* In *SODA*, pages 459-467, 2012.

[AhnGM-12b]
Kook Jin Ahn, Sudipto Guha, and Andrew McGregor. *Graph sketches: sparsification, spanners, and subgraphs.* In *PODS*, pages 5-14, 2012.

[AilonL-11]
Nir Ailon and Edo Liberty. *An almost optimal unrestricted fast Johnson-Lindenstrauss transform.* In *ACM-SIAM Symposium on Discrete Algorithms*, pages 185-191, 2011.

[AlonMS-99]
Noga Alon, Yossi Matias, Mario Szegedy. *The Space Complexity of Approximating the Frequency Moments.* J. Comput. Syst. Sci. 58(1):137-147, 1999.

[AndoniCKQWZ-16]
Alexandr Andoni, Jiecao Chen, Robert Krauthgamer, Bo Qin, David P. Woodruff, and Qin Zhang. *On Sketching Quadratic Forms.* In *ITCS*, pages 311-319, 2016.

[AndoniDIW-09]
Alexandr Andoni, Khanh Do Ba, Piotr Indyk, and David P. Woodruff. *Efficient sketches for earth-mover distance, with applications.* In *IEEE Symposium on Foundations of Computer Science*, pages 324-330, 2009.

[AndoniGK-14]
Alexandr Andoni, Anupam Gupta, and Robert Krauthgamer. *Towards $(1+\varepsilon)$-Approximate Flow Sparsifiers.* In *SODA*, pages 279-293, 2014.

[AndoniIK-08]
Alexandr Andoni, Piotr Indyk, and Robert Krauthgamer. *Earth mover distance over high-dimensional spaces.* In *SODA*, pages 343-352, 2008.

[AndoniIK-09]
Alexandr Andoni, Piotr Indyk, and Robert Krauthgamer. *Overcoming the ℓ_1 non-embeddability barrier: algorithms for product metrics.* In *ACM-SIAM Symposium on Discrete Algorithms*, pages 865-874, 2009.

[AndoniJP-10]
Alexandr Andoni, T. S. Jayram, and Mihai Patrascu. *Lower bounds for edit distance and product metrics via Poincaré-type inequalities.* In *ACM-SIAM Symposium on Discrete Algorithms*, pages 184-192, 2010.

[AndoniKR-14]
Alexandr Andoni, Robert Krauthgamer, and Ilya Razenshteyn. *Sketching and Embedding are Equivalent for Norms.* In *CoRR*, abs/1411.2577, 2014.

[AndoniN-12]
Alexandr Andoni and Huy L. Nguyen. *Width of points in the streaming model.* In *ACM-SIAM Symposium on Discrete Algorithms*, pages 447-452, 2012.

[AndoniO-09]
Alexandr Andoni and Krzysztof Onak. *Approximating edit distance in near-linear time.* In *ACM Symposium on Theory of Computing*, pages 199-204, 2009.

[AssadiKLY-16]
Sepehr Assadi, Sanjeev Khanna, Yang Li, and Grigory Yaroslavtsev. *Maximum Matchings in Dynamic Graph Streams and the Simultaneous Communication Model.* In *SODA*, pages 1345-1364, 2016.

[AssadiKL-17]
Sepehr Assadi, Sanjeev Khanna, and Yang Li. *On Estimating Maximum Matching Size in Graph Streams.* In *SODA*, pages 1723-1742, 2017.

[BahBC-14]
Bubaccar Bah, Luca Baldassarre, and Volkan Cevher. *Model-based Sketching and Recovery with Expanders.* In *ACM-SIAM Symposium on Discrete Algorithms*, 2014.

[BaraniukCDH-10]
Richard G. Baraniuk, Volkan Cevher, Marco F. Duarte, and Chinmay Hegde. *Model-based compressive sensing.* IEEE Transactions on Information Theory, 56(4):1982-2001, 2010.

[BarYossefJKS-04]
Ziv Bar-Yossef, T. S. Jayram, Ravi Kumar, and D. Sivakumar. *An information statistics approach to data stream and communication complexity.* J. Comput. Syst. Sci., 68(4):702-732, 2004.

[BarYossefJKST-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.

[BarYossefKS-02]
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.

[Baswana-06]
Surender Baswana. *Faster streaming algorithms for graph spanners.* 2006.

[BatuFFKRW-01]
Tugkan Batu, Eldar Fischer, Lance Fortnow, Ravi Kumar, Ronitt Rubinfeld, and Patrick White. *Testing random variables for independence and identity.* In *FOCS*, pages 442-451, 2001.

[BenderR-02]
Michael A. Bender and Dana Ron. *Testing properties of directed graphs: acyclicity and connectivity.* Random Struct. Algorithms, 20(2):184-205, 2002.

[BenjaminiSS-08]
Itai Benjamini, Oded Schramm, and Asaf Shapira. *Every minor-closed property of sparse graphs is testable.* In *ACM Symposium on Theory of Computing*, pages 393-402, 2008.

[BenSassonGMSS-11]
Eli Ben-Sasson, Elena Grigorescu, Ghid Maatouk, Amir Shpilka, and Madhu Sudan. *On sums of locally testable affine invariant properties.* Electronic Colloquium on Computational Complexity (ECCC), 18:79, 2011.

[BerindeIR-08]
Radu Berinde, Piotr Indyk, and Milan Ruzic. *Practical near-optimal sparse recovery in the l_1 norm.* Allerton, 2008.

[BermanRY-14]
Piotr Berman, Sofya Raskhodnikova, Grigory Yaroslavtsev. *L_p-Testing.* In *ACM Symposium on Theory of Computing*, pages 164-173, 2014.

[BhattacharyyaMMY-07]
S. Bhattacharyya, A. Madeira, S. Muthukrishnan, and T. Ye. *How to scalably skip past streams.* In *WSSP (Workshop with ICDE)*, 2007.

[BhuvanagiriG-06]
Lakshminath Bhuvanagiri and Sumit Ganguly. *Estimating entropy over data streams.* In *ESA*, pages 148-159, 2006.

[BhuvanagiriGKS-06]
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.

[BlumLR-93]
Manuel Blum, Michael Luby, and Ronitt Rubinfeld. *Self-Testing/Correcting with Applications to Numerical Problems.* J. Comput. Syst. Sci. 47(3):549-595, 1993.

[BogdanovOT-02]
Andrej Bogdanov, Kenji Obata, and Luca Trevisan. *A lower bound for testing 3-colorability in bounded-degree graphs.* In *FOCS*, pages 93-102, 2002.

[BonisGV-05]
Annalisa De Bonis, Leszek Gasieniec, and Ugo Vaccaro. *Optimal two-stage algorithms for group testing problems.* SIAM J. Comput., 34(5):1253-1270, 2005.

[BoulandCHTV-16]
Adam Bouland, Lijie Chen, Dhiraj Holden, Justin Thaler, and Prashant Nalini Vasudevan. *On SZK and PP.* CoRR abs/1609.02888, 2016.

[BravermanGPW-13]
Mark Braverman, Ankit Garg, Denis Pankratov, and Omri Weinstein. *Information lower bounds via self-reducibility.* In *CSR*, 2013.

[BravermanO-10]
Vladimir Braverman and Rafail Ostrovsky. *Zero-one frequency laws.* In *ACM Symposium on Theory of Computing*, pages 281-290, 2010.

[BroderCFM-00]
Andrei Z. Broder, Moses Charikar, Alan M. Frieze, and Michael Mitzenmacher. *Min-wise independent permutations. J. Comput. Syst. Sci.*, 60(3):630-659, 2000.

[BrodyC-09]
Joshua Brody and Amit Chakrabarti. *A multi-round communication lower bound for Gap Hamming and some consequences.* In *IEEE Conference on Computational Complexity*, pages 358-368, 2009.

[BrodyCRVW-10]
Joshua Brody, Amit Chakrabarti, Oded Regev, Thomas Vidick, and Ronald de Wolf. *Better Gap-Hamming lower bounds via better round elimination.* In *APPROX-RANDOM*, pages 476-489, 2010.

[BrodyJSW-14]

Joshua Brody, Sune K. Jakobsen, Dominik Scheder, and Peter Winkler. *Cryptogenography.* In *ITCS*, pages 13-22, 2014.

[BuriolFLMS-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.

[CandesRT-06]
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.

[CandesRT-06a]
Emmanuel J. Candès, Justin Romberg, and Terence Tao. *Stable signal recovery from incomplete and inaccurate measurements.* Comm. Pure Appl. Math., 59(8):1208-1223, 2006.

[CanonneGR-16]
Clément L. Canonne, Themis Gouleakis, and Ronitt Rubinfeld. *Sampling Correctors.* In *ITCS* (to appear), 2016.

[CanonneRS-14]
Clément L. Canonne, Dana Ron, and Rocco A. Servedio. *Testing equivalence between distributions using conditional samples.* In *SODA*, pages 1174-1192, 2014.

[Chakrabarti-10]
Amit Chakrabarti. *A note on randomized streaming space bounds for the longest increasing subsequence problem.* Electronic Colloquium on Computational Complexity (ECCC), 10(10), 2010.

[ChakrabartiCKM-10]
Amit Chakrabarti, Graham Cormode, Ranganath Kondapally, and Andrew McGregor. *Information cost tradeoffs for augmented index and streaming language recognition.* In *IEEE Symposium on Foundations of Computer Science*, pages 387-396, 2010.

[ChakrabartiCM-07]
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.

[ChakrabartiCM-09]
Amit Chakrabarti, Graham Cormode, and Andrew McGregor. *Annotations in data streams.* In *International Colloquium on Automata, Languages and Programming*, pages 222-234, 2009.

[ChakrabartiCMTV-15]
Amit Chakrabarti, Graham Cormode, Andrew McGregor, Justin Thaler, and Suresh Venkatasubramanian. *Verifiable Stream Computation and Arthur-Merlin Communication.* In *IEEE Conference on Computational Complexity*, pages 217-243, 2015.

[ChakrabartiK-14]
Amit Chakrabarti and Sagar Kale. *Submodular Maximization Meets Streaming: Matchings, Matroids, and More.* In *IPCO*, to appear, 2014.

[ChakrabartiR-11]
Amit Chakrabarti and Oded Regev. *An optimal lower bound on the communication complexity of Gap-Hamming-Distance.* In *ACM Symposium on Theory of Computing*, pages 51-60, 2011.

[ChakrabartiSWY-01]
Amit Chakrabarti, Yaoyun Shi, Anthony Wirth, and Andrew C. Yao. *Informational complexity and the direct sum problem for simultaneous message complexity.* In *IEEE Symposium on Foundations of Computer Science*, pages 270-278, 2001.

[ChakrabartyS-13]
Deeparnab Chakrabarty and C. Seshadhri. *Optimal bounds for monotonicity and Lipschitz testing over hypercubes and hypergrids.* In *ACM Symposium on Theory of Computing*, 2013.

[ChakrabortyFGM-13]
Sourav Chakraborty, Eldar Fischer, Yonatan Goldhirsh, Arie Matsliah. *On the power of conditional samples in distribution testing.* In *ITCS*, pages 561-580, 2013.

[ChanC-07]
Timothy M. Chan and Eric Y. Chen. *Multi-pass geometric algorithms.* Discrete & Computational Geometry, 37(1):79-102, 2007.

[Charikar-02]
Moses Charikar. *Similarity estimation techniques from rounding algorithms.* In *STOC*, pages 380-388, 2002.

[ChazelleRT-05]
Bernard Chazelle, Ronitt Rubinfeld, and Luca Trevisan. *Approximating the Minimum Spanning Tree Weight in Sublinear Time.* SIAM J. Comput. 34(6):1370-1379, 2005.

[ChienRS-03]
Steve Chien, Lars Eilstrup Rasmussen, and Alistair Sinclair. *Clifford algebras and approximating the permanent.* J. Comput. Syst. Sci. 67(2), 2003.

[ChitnisCEHMMV-16]
Rajesh Chitnis, Graham Cormode, Hossein Esfandiari, MohammadTaghi Hajiaghayi, Andrew McGregor, Morteza Monemizadeh, and Sofya Vorotnikova. *Kernelization via Sampling with Applications to Finding Matchings and Related Problems in Dynamic Graph Streams.* In *SODA*, pages 1326-1344, 2016.

[CormodeDIM-03]
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.

[CormodeKMS-06]
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.

[CormodeM-05]
Graham Cormode and S. Muthukrishnan. *An improved data stream summary: the count-min sketch and its applications. J. Algorithms*, 55(1):58-75, 2005.

[CormodeM-05a]
Graham Cormode and S. Muthukrishnan. *What's new: finding significant differences in network data streams. IEEE/ACM Trans. Netw.*, 13(6):1219-1232, 2005.

[CormodeM-05b]
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.

[CormodeM-06]
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.

[CormodePSV-00]
Graham Cormode, Mike Paterson, Suleyman Cenk Sahinalp, and Uzi Vishkin. *Communication complexity of document exchange.* In *SODA*, 2000.

[CrouchM-11]
Michael S. Crouch and Andrew McGregor. *Periodicity and Cyclic Shifts via Linear Sketches.* In *APPROX*, 2011.

[CrouchS-14]
Michael Crouch and Daniel Stubbs. In Andrew McGregor's presentation at the 2014 Bertinoro Workshop on Sublinear Algorithms.

[CzumajS-09]
Artur Czumaj and Christian Sohler. *Estimating the Weight of Metric Minimum Spanning Trees in Sublinear Time.* In *SIAM J. Comput.*, 39(3):904–922, 2009.

[DaskalakisDS-11]
Constantinos Daskalakis, Ilias Diakonikolas, and Rocco A. Servedio. *Learning transformed product distributions.* CoRR, abs/1103.0598, 2011.

[DasSarmaGP-08]
Atish Das Sarma, Sreenivas Gollapudi, and Rina Panigrahy. *Estimating PageRank on graph streams.* In *PODS*, pages 69-78, 2008.

[DeanG-04]
Jeffrey Dean and Sanjay Ghemawat. *MapReduce: Simplified data processing on large clusters.* In *OSDI*, pages 137-150, 2004.

[DemaineLM-02]
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.

[DemetrescuFR-06]
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.

[DodisGLRRS-99]
Yevgeniy Dodis, Oded Goldreich, Eric Lehman, Sofya Raskhodnikova, Dana Ron, and Alex Samorodnitsky. *Improved testing algorithms for monotonicity.* In *RANDOM-APPROX*, pages 97-108, 1999.

[DoerrK-16]
Benjamin Doerr, Marvin Kunnemann. *Improved Protocols and Hardness Results for the Two-Player Cryptogenography Problem.* In *ICALP*, 2016.

[Donoho-06]
David L. Donoho. *Compressed sensing.* IEEE Transactions on Information Theory, 52(4):1289-1306, 2006.

[DrakeH-03]
Doratha E. Drake and Stefan Hougardy. *Improved linear time approximation algorithms for weighted matchings.* In *RANDOM-APPROX*, pages 14-23, 2003.

[DrineasMM-06]
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.

[DrineasMM-06a]
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.

[DrineasMM-06b]
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.

[Eddy-04]

Sean R Eddy. *How do RNA folding algorithms work?* Nature Biotechnology, 22, pages 1457-1458, 2004.

[Edmonds-65]
Jack Edmonds. *Maximum matching and a polyhedron with 0,1-vertices.* J. Res. Nat. Bur. Standards, 69(B):125-130, 1965.

[Elkin-06]
Michael Elkin. *A near-optimal fully dynamic distributed algorithm for maintaining sparse spanners.* 2006.

[ElkinZ-06]
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.

[EpsteinLMS-11]
Leah Epstein, Asaf Levin, Julian Mestre, and Danny Segev. *Improved Approximation Guarantees for Weighted Matching in the Semi-streaming Model.* SIAM J. Discrete Math. 25(3), 2011.

[ErgunJ-08]
Funda Ergün and Hossein Jowhari. *On distance to monotonicity and longest increasing subsequence of a data stream.* In *ACM-SIAM Symposium on Discrete Algorithms*, pages 730-736, 2008.

[FalahatgarJOPS-15]
Moein Falahatgar, Ashkan Jafarpour, Alon Orlitsky, Venkatadheeraj Pichapathi, and Ananda Theertha Suresh. *Faster Algorithms for Testing under Conditional Sampling.* In *CoRR*, abs/1504.04103, 2015.

[FeigenbaumKMSZ-05]
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.

[FeigenbaumKMSZ-05a]
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.

[FeigenbaumKMSZ-08]
Joan Feigenbaum, Sampath Kannan, Andrew McGregor, Siddharth Suri, and Jian Zhang. *Graph distances in the data-stream model.* SIAM J. Comput., 38(5):1709-1727, 2008.

[FeigenbaumKSV-02]
Joan Feigenbaum, Sampath Kannan, Martin Strauss, and Mahesh Viswanathan. *An approximate* L^1 *difference algorithm for massive data streams. Journal on Computing*, 32(1):131-151, 2002.

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

[FeldmanMSSS-10]
Jon Feldman, S. Muthukrishnan, Anastasios Sidiropoulos, Clifford Stein, and Zoya Svitkina. *On distributing symmetric streaming computations.* ACM Transactions on Algorithms, 6(4), 2010.

[FichtenbergerPS-15]
Hendrik Fichtenberger, Pan Peng, and Christian Sohler. *On constant-size graphs that preserve the local structure of high-girth graphs.* In *APPROX-RANDOM*, pages 786-799, 2015.

[Forster-01]
Jurgen Forster. *A Linear Lower Bound on the Unbounded Error Probabilistic Communication Complexity.* In *IEEE Conference on Computational Complexity*, pages 100-106, 2001.

[GabizonH-10]
Ariel Gabizon and Avinatan Hassidim. *Derandomizing algorithms on product distributions and other applications of order-based extraction.* In *ICS*, pages 397-405, 2010.

[Gabow-90]
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.

[GalG-07]
Anna Gál and Parikshit Gopalan. *Lower bounds on streaming algorithms for approximating the length of the longest increasing subsequence.* In *IEEE Symposium on Foundations of Computer Science*, pages 294-304, 2007.

[Ganguly-06]
Sumit Ganguly and Anirban Majumder. *CR-precis: A deterministic summary structure for update data streams.* In *ESCAPE*, 2007.

[GilbertKMS-02]
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.

[GilbertSTV-07]
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.

[GoldreichR-02]
Oded Goldreich and Dana Ron. *Property Testing in Bounded Degree Graphs.* Algorithmica, 32(2):302-343, 2002

[GolubV-89]
G.H. Golub and C.F. Van Loan. *Matrix Computations.* Johns Hopkins University Press, Baltimore, 1989.

[GomoryH-61]
R. E. Gomory and T. C. Hu. *Multi-terminal network flows.* Journal of the Society for Industrial and Applied Mathematics, 9(4):551-570, 1961.

[GopalanJKK-07]
Parikshit Gopalan, T. S. Jayram, Robert Krauthgamer, and Ravi Kumar. *Estimating the sortedness of a data stream.* In *ACM-SIAM Symposium on Discrete Algorithms*, pages 318-327, 2007.

[GoreinovT-01]
S. A. Goreinov and E. E. Tyrtyshnikov. *The maximum-volume concept in approximation by low-rank matrices.* Contemporary Mathematics, 280, pages 47-51, 2001.

[GoreinovTZ-97]
S. A. Goreinov, E. E. Tyrtyshnikov, and N. L. Zamarashkin. *A theory of pseudoskeleton approximations.* Linear Algebra and its Applications, 261, pages 1-21, August 1997.

[Gould-96]
Stephen Jay Gould.*The Mismeasure of Man.* W. W. Norton and Company, 1996.

[GreenwaldK-01]
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.

[GrigorescuKS-08]
Elena Grigorescu, Tali Kaufman, and Madhu Sudan. *2-transitivity is insufficient for local testability.* In *IEEE Conference on Computational Complexity*, pages 259-267, 2008.

[GroheGLSTV-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.

[GuchtWWZ-15]
Dirk Van Gucht, Ryan Williams, David P. Woodruff, and Qin Zhang. *The Communication Complexity of Distributed Set-Joins with Applications to Matrix Multiplication.* In *PODS*, pages 199-212, 2015.

[GuE-96]
M. Gu and S.C. Eisenstat. *Efficient algorithms for computing a strong rank-revealing QR factorization.* SIAM Journal on Scientific Computing, 17, pages 848-869, 1996.

[GuhaIM-07]
Sudipto Guha, Piotr Indyk, and Andrew McGregor. *Sketching information divergences.* In *Conference on Learning Theory*, 2007.

[GuhaM-06]
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.

[GuhaM-07]
Sudipto Guha and Andrew McGregor. *Lower bounds for quantile estimation in random-order and multi-pass streaming.* Manuscript, 2007.

[GuhaM-07a]
Sudipto Guha and Andrew McGregor. *Space-efficient sampling.* In *AISTATS*, pages 169-176, 2007.

[GuhaM-08]
Sudipto Guha and Andrew McGregor. *Tight lower bounds for multi-pass stream computation via pass elimination.* In *International Colloquium on Automata, Languages and Programming*, pages 760-772, 2008.

[GuhaMV-06]
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.

[GuruswamiO-13]
Venkatesan Guruswami and Krzysztof Onak. *Superlinear lower bounds for multipass graph processing.* In *IEEE Conference on Computational Complexity*, 2013.

[GuruswamiR-05]
Venkatesan Guruswami and Atri Rudra. *Tolerant Locally Testable Codes.* In *Proceedings of the 9th International Workshop on Randomization and Computation (RANDOM)*, 2005.

[HagerupKNR-98]
Torben Hagerup, Jyrki Katajainen, Naomi Nishimura, and Prabhakar Ragde. *Characterizing Multiterminal Flow Networks and Computing Flows in Networks of Small Treewidth.* In J. Comput. Syst. Sci. (JCSS) 57(3):366-375, 1998.

[HarveyNO-08]
Nicholas J. A. Harvey, Jelani Nelson, and Krzysztof Onak. *Sketching and streaming entropy via approximation theory.* In *IEEE Symposium on*

Foundations of Computer Science, pages 489-498, 2008.

[HassidimKNO-09]
Avinatan Hassidim, Jonathan A. Kelner, Huy N. Nguyen, and Krzysztof Onak. *Local graph partitions for approximation and testing.* In *IEEE Symposium on Foundations of Computer Science*, pages 22-31, 2009.

[HershbergerSST-04]
John Hershberger, Nisheeth Shrivastava, Subhash Suri, and Csaba D. Tóth. *Adaptive spatial partitioning for multidimensional data streams.* In *ISAAC*, pages 522-533, 2004.

[HoffmannMR-04]
M. Hoffmann, S. Muthukrishnan, and R. Raman. *Location streams: Models and algorithms.* Technical Report 2004-28, DIMACS, May 2004.

[HopcroftK-73]
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.

[Indyk-00]
Piotr Indyk. *Stable distributions, pseudorandom generators, embeddings and data stream computation.* In *IEEE Symposium on Foundations of Computer Science*, pages 189-197, 2000.

[Indyk-04]
Piotr Indyk. *Algorithms for dynamic geometric problems over data streams.* In *ACM Symposium on Theory of Computing*, pages 373-380, 2004.

[IndykP-11]
Piotr Indyk and Eric Price. *K-median clustering, model-based compressive sensing, and sparse recovery for earth mover distance.* In *ACM Symposium on Theory of Computing*, pages 627-636, 2011.

[IndykR-08]
Piotr Indyk and Milan Ruzic. *Near-optimal sparse recovery in the ℓ_1 norm.* In *IEEE Symposium on Foundations of Computer Science*, pages 199-207, 2008.

[IndykR-13]
Piotr Indyk and Ilya Razenshteyn. *On Model-Based RIP-1 Matrices.* In *International Colloquium on Automata, Languages and Programming (ICALP)*, pages 564-575, 2013.

[IndykT-03]
Piotr Indyk and Niten Thaper. *Fast color image retrieval via embeddings.* Workshop on Statistical and Computational Theories of Vision (at ICCV), 2003.

[IndykW-03]
Piotr Indyk and David P. Woodruff. *Tight lower bounds for the distinct elements problem.* In *IEEE Symposium on Foundations of Computer Science*, pages 283-288, 2003.

[IndykW-05]
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.

[JainN-10]
Rahul Jain and Ashwin Nayak. *The space complexity of recognizing well-parenthesized expressions.* Electronic Colloquium on Computational Complexity (ECCC), 17:71, 2010.

[Jakobsen-14]
Sune Jakobsen. *Information Theoretical Cryptogenography.* In *ITCS*, 2014.

[JayramKS-07]
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.

[JayramW-09]
T. S. Jayram and David P. Woodruff. *The data stream space complexity of cascaded norms.* In *IEEE Symposium on Foundations of Computer Science*, 2009.

[JhaR-11]
Madhav Jha and Sofya Raskhodnikova. *Testing and reconstruction of Lipschitz functions with applications to data privacy.* Electronic Colloquium on Computational Complexity (ECCC), 18:57, 2011.

[Jowhari-12]
Hossein Jowhari. *Efficient Communication Protocols for Deciding Edit Distance.* In *ESA*, 2012.

[JowhariG-05]
Hossein Jowhari and Mohammad Ghodsi. *New streaming algorithms for counting triangles in graphs.* In *COCOON*, pages 710-716, 2005.

[KalantariS-95]
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.

[KaneMSS-12]
Daniel M. Kane, Kurt Mehlhorn, Thomas Sauerwald, and He Sun. *Counting Arbitrary Subgraphs in Data Streams.* In *Proceedings of the 39th International Colloquium on Automata, Languages, and Programming (2)*, 2012.

[KannanMY-16]
Sampath Kannan, Elchanan Mossel, and Grigory Yaroslavtsev. *Linear Sketching over $\mathbb{F}_2$.* In *CoRR*, abs/1611.01879, 2016.

[Kapralov-12]
Michael Kapralov. *Improved lower bounds for matchings in the streaming model.* In *CoRR*, abs/1206.2269, 2012.

[KapralovKS-15]
Michael Kapralov, Sanjeev Khanna, and Madhu Sudan. *Streaming Lower Bounds for Approximating MAX-CUT.* In *SODA*, pages 1263-1282, 2015.

[KapralovKSV-17]
Michael Kapralov, Sanjeev Khanna, Madhu Sudan, and Ameya Velingker. *$(1+\Omega(1))$-Approximation to MAX-CUT Requires Linear Space.* In *SODA*, pages 1703-1722, 2017.

[KarloffSV-10]
Howard J. Karloff, Siddharth Suri, and Sergei Vassilvitskii. *A model of computation for MapReduce.* In *ACM-SIAM Symposium on Discrete Algorithms*, pages 938-948, 2010.

[KerenidisLLRX-12]
Iordanis Kerenidis, Sophie Laplante, Virginie Lerays, Jérémie Roland, David Xiao. *Lower bounds on information complexity via zero-communication protocols and applications.* In *FOCS*, 2012.

[KhanR-14]
Arindam Khan and Prasad Raghavendra. *On mimicking networks representing minimum terminal cuts.* Inf. Process. Lett. (IPL) 114(7):365-371, 2014.

[KhotN-06]
Subhash Khot and Assaf Naor. *Nonembeddability theorems via Fourier analysis.* Math. Ann., 334(4):821-852, 2006.

[KoganK-15]
Dmitry Kogan and Robert Krauthgamer. *Sketching Cuts in Graphs and Hypergraphs.* In *ITCS*, pages 367-376, 2015.

[KonradMM-12]
Christian Konrad, Frederic Magniez, and Claire Mathieu. *Maximum Matching in Semi-Streaming with Few Passes.* In *Proceedings of 15th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems*, 2012.

[KrahmerW-11]
Felix Krahmer, Rachel Ward. *New and improved Johnson-Lindenstrauss embeddings via the Restricted Isometry Property.* SIAM J. Math. Anal., 43(3):1269-1281, 2011.

[KuroseR-04]
James F. Kurose and Keith W. Ross. *Computer Networking: A Top-Down Approach Featuring the Internet.* Addison Wesley, 2004.

[KushilevitzOR-00]
Eyal Kushilevitz, Rafail Ostrovsky, and Yuval Rabani. *Efficient search for approximate nearest neighbor in high dimensional spaces.* SIAM J. Comput., 30(2):457-474, 2000.

[LarsenW-17]
Kasper Green Larsen and Ryan Williams. *Faster Online Matrix-Vector Multiplication.* In *SODA*, pages 2182-2189, 2017.

[LeviR-13]
Reut Levi and Dana Ron. *A Quasi-Polynomial Time Partition Oracle for Graphs with an Excluded Minor.* In *CoRR*, abs/1302.3417, 2013.

[Li-06]
Ping Li. *Very sparse stable random projections, estimators and tail bounds for stable random projections.*

[LiHC-06]
Ping Li, Trevor Hastie, and Kenneth Ward Church. *Very sparse random projections.* In *KDD*, pages 287-296, 2006.

[LibenNowellVZ-06]
David Liben-Nowell, Erik Vee, and An Zhu. *Finding longest increasing and common subsequences in streaming data.* J. Comb. Optim., 11(2):155-175, 2006.

[LibertySS-16]
Edo Liberty, Ram Sriharsha, and Maxim Sviridenko. *An Algorithm for Online K-Means Clustering.* In *ALENEX*, pages 81-89, 2016.

[MagniezMN-10]
Frédéric Magniez, Claire Mathieu, and Ashwin Nayak. *Recognizing well-parenthesized expressions in the streaming model.* In *ACM Symposium on Theory of Computing*, pages 261-270, 2010.

[MahoneyMD-06]
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.

[ManjunathMPS-11]
Madhusudan Manjunath, Kurt Mehlhorn, Konstantinos Panagiotou, and He Sun. *Approximate Counting of Cycles in Streams.* In *Proceedings of the 19th Annual European Symposium*, 2011.

[MankuRL-98]
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.

[MankuRL-99]
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.

[Matousek-02]
Jiří Matoušek. *Lectures on Discrete Geometry*. Springer, 2002.

[McGregor-05]
Andrew McGregor. *Finding graph matchings in data streams.* In *APPROX-RANDOM*, pages 170-181, 2005.

[McGregorRU-11]
Andrew McGregor, Atri Rudra, and Steve Uurtamo. *Polynomial Fitting of Data Streams with Applications to Codeword Testing.* In *Proceedings of the 28th International Symposium on Theoretical Aspects of Computer Science (STACS)*, 2011.

[MetwallyAA-05]
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.

[MicaliV-80]
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.

[MisraG-82]
Jayadev Misra and David Gries. *Finding repeated elements. Sci. Comput. Program.*, 2(2):143-152, 1982.

[MitzenmacherV-08]
Michael Mitzenmacher and Salil P. Vadhan. *Why simple hash functions work: exploiting the entropy in a data stream.* In *ACM-SIAM Symposium on Discrete Algorithms*, pages 746-755, 2008.

[MontanaroO-09]
Ashley Montanaro and Tobias Osborne. *On the communication complexity of XOR functions.* In *CoRR*, abs/0909.3392, 2009.

[MunroP-80]
J. Ian Munro and Mike Paterson. *Selection and sorting with limited storage.* Theor. Comput. Sci., 12:315-323, 1980.

[Muthukrishnan-06]
S. Muthukrishnan. *Data streams: Algorithms and applications.* Now Publishers, 2006.

[Muthukrishnan-06a]
S. Muthukrishnan. *Some algorithmic problems and results in compressed sensing.* In *Allerton Conference*, 2006.

[NaorS-06]
Assaf Naor and Gideon Schechtman. *Planar earthmover is not in l_1.* In *FOCS*, pages 655-666, 2006.

[NeedellT-10]
Deanna Needell and Joel A. Tropp. *Cosamp: iterative signal recovery from incomplete and inaccurate samples.* Commun. ACM, 53(12):93-100, 2010.

[NguyenO-08]
Huy N. Nguyen and Krzysztof Onak. *Constant-time approximation algorithms via local improvements.* In *IEEE Symposium on Foundations of Computer Science*, pages 327-336, 2008.

[Onak-10]
Krzysztof Onak. *New Sublinear Methods in the Struggle Against Classical Problems.*PhD thesis, Massachusetts Institute of Technology, 2010.

[OnakRRR-12]
Krzysztof Onak, Dana Ron, Michal Rosen, and Ronitt Rubinfeld. *A near-optimal sublinear-time algorithm for approximating the minimum vertex cover size.* In *ACM-SIAM Symposium on Discrete Algorithms*, pages 1123-1131, 2012.

[PettieS-04]
Seth Pettie and Peter Sanders. *A simpler linear time $2/3 - \epsilon$ approximation for maximum weight matching.* Inf. Process. Lett., 91(6):271-276, 2004.

[RudelsonV-06]
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.

[RudraU-10]
Atri Rudra and Steve Uurtamo. *Data Stream Algorithms for Codeword Testing.* In *Proceedings of the 37th International Colloquium on Automata, Languages and Programming (ICALP)*, 2010.

[SeshadhriV-11]
C. Seshadhri and Jan Vondrák. *Is submodularity testable?* In *ICS*, 2011.

[ShrivastavaBAS-04]
Nisheeth Shrivastava, Chiranjeeb Buragohain, Divyakant Agrawal, and Subhash Suri. *Medians and beyond: new aggregation techniques for sensor networks.* In *SenSys*, pages 239-249, 2004.

[Stewart-99]
G.W. Stewart. *Four algorithms for the efficient computation of truncated QR approximations to a sparse matrix.* Numerische Mathematik, 83, pages 313-323, 1999.

[Stewart-04]
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.

[Thaler-16]
Justin Thaler. *Semi-Streaming Algorithms for Annotated Graph Streams.* In *ICALP*, 2016.

[Trevisan-09]
Luca Trevisan. *Max Cut and the smallest eigenvalue.* In *ACM Symposium on Theory of Computing*, pages 263-272, 2009.

[YoshidaYI-09]
Yuichi Yoshida, Masaki Yamamoto, and Hiro Ito. *An improved constant-time approximation algorithm for maximum matchings.* In *ACM Symposium on Theory of Computing*, pages 225-234, 2009.

[Woodruff-04]
David P. Woodruff. *Optimal space lower bounds for all frequency moments.* In *ACM-SIAM Symposium on Discrete Algorithms*, pages 167-175, 2004.