Data stream algorithms via expander graphs

Sumit Ganguly
sganguly@cse.iitk.ac.in

IIT Kanpur, India

Abstract. We present a simple way of designing deterministic algorithms for problems in the data stream model via lossless expander graphs. We illustrate this by considering two problems, namely, k-sparsity testing and estimating frequency of items.

1 Introduction

We say that an n-dimensional vector f from $\mathbb{Z}^n$ is *k-sparse* if it has at most k non-zero entries. The problem is to test whether f is k-sparse or no after it has been subject to a sequence of coordinate wise updates in arbitrary order, that is, f is the frequency vector of a data stream. More formally, a data stream over the domain $[n] = \{1, 2, \ldots, n\}$ is a sequence σ of records of the form $(index, i, v)$, where, *index* is the position of the record in the sequence, $i \in [n]$ and $v \in \mathbb{Z}$. Associated with each data stream σ is an n-dimensional *frequency vector* $f(\sigma)$, such that $f_i(\sigma)$ is the frequency of i, or the cumulative sum of the updates to $f_i(\sigma)$, made by the sequence σ. That is,

$$f_i(\sigma) = \sum_{(index,i,v)\in\sigma} v, \quad i \in [n] \ .$$

The k-sparsity testing problem is as follows: design a data structure, referred to as a *k-sparsity tester*, that (a) processes any stream σ of updates over the domain $[n]$, and, (b) provides a test to check whether $f(\sigma)$ is k-sparse, that is, whether, f has at most k non-zero entries. The problem is to obtain solutions whose space requirement is $o(n)$.

We first review work on the following well-studied and closely related problem, namely, *k-sparse vector reconstruction* problem, where it is required to design a structure that can process a data stream σ and can retrieve the frequency vector $f(\sigma)$ provided $f(\sigma)$ is k-sparse. However, the structure is not required to actually test whether $f(\sigma)$ is k-sparse or not and may present an incorrect answer if $f(\sigma)$ is not k-sparse. Let m denote $\max_{i=1}^{n}|f_i|$. It is easy to show [15] that the k-sparse reconstruction problem requires $\Omega(k \log(mn/k))$ bits. Minsky et. al. [22] study a constrained version of the k-sparse vector reconstruction problem where $f(\sigma) \in \{-1, 0, 1\}^n$ and present a space-optimal algorithm for this scenario. Eppstein and Goodrich [11] present a space-optimal solution for the case when $f(\sigma) \in \{0, 1\}^n$. The k-set structure [15] presents a k-sparse vector reconstruction technique for the general case when $f \in \{-m, \ldots, m\}^n$. We reproduce their theorem since we will refer to it later.

Theorem 1 ([15]). *For vectors $f \in \{-m, \ldots, m\}^n$, there exists a data structure for the k-sparse reconstruction problem that requires space $O(k \log(mn) \log(n/k))$ bits. The time taken to process any coordinate-wise update to f is $O(k(\log(n/k)))$ elementary arithmetic operations over a finite field of size $O(mn)$ and characteristic at least $mn + 1$.* □

The work on compressed sensing [3, 10] has independently considered the problem of k-sparse vector reconstruction. Based on previous work in [3, 10, 18], Indyk [20] presented the first deterministic algorithm in the compressed sensing framework for k-sparse vector reconstruction using space $O(k2^{(\log \log n)^E} \log(mn))$ bits, where, $E > 1$ is a constant that depends on the best known construction of a class of extractors.

We now review prior work on k-sparsity testing. It is known that k-sparsity testing for vectors in $\{-m, \ldots, m\}^n$ requires $\Omega(n)$ space, for any $m \geq 1$ and $k \geq 0$ [11, 16]. In view of this negative result, in this paper, we will restrict our attention to non-negative frequency vectors from $\{0, \ldots, m\}^n$, that is, $0 \leq f_i(\sigma) \leq m$, for each $i \in [n]$. A space-optimal 1-sparsity tester was presented in [12] that requires $O(\log(mn))$ bits. A k-sparsity tester can be constructed by using *strongly selective families* [6, 8] as follows. An (n, k) strongly selective family is a family of sets $\{S_i\}_{1 \leq i \leq t}$ such that for any $A \subset \{1, 2, \ldots, n\}$ such that $|A| \leq k$ and for any $x \in A$, there exists a member S_j of the family such that $S_j \cap A = \{x\}$. In other words, each member of the set A is *selected via intersection* by some member of the family. Constructive solutions for (n, k)-strongly selective families are known for which the size of the family $t = O(k^2 \cdot \text{polylog}(n))$. A k-sparsity test can be designed by keeping a 1-sparsity tester [12] for each of the sets $\{S_i\}_{1 \leq i \leq t}$. The space requirement is $O(k^2 \cdot \text{polylog}(n) \log(mn))$ bits. This line of work cannot be used to obtain significantly more space efficient k-sparsity tests, since, there is a space lower bound of $\Omega(k^2(\log(n/k))/(\log k))$ for the size of (n, k)-strongly selective family [6]. The k-set structure [15] presents a technique that can be used to test k-sparsity of vectors in $\{0, \ldots, m\}^n$ using space $O(k^2 \log n + k \log(mn))$. Finite fields based solutions to k-sparse vector construction of Minsky et.al. [22], Eppstein and Goodrich [11], our previous work in [15], and, the compressed sensing approach of Indyk [20] are not known to directly extend to deterministic k-sparsity testing.

Deterministic estimation of data stream frequency. We consider deterministic solution to the ApproxFreq(ϵ) problem, namely, to design a low-space data structure that can (a) process any stream σ over the domain $[n]$, and, (b) given any $i \in [n]$, it returns a deterministic estimate $\hat{f}_i$ for $f_i(\sigma)$ satisfying $|\hat{f}_i - f_i(\sigma)| \leq \epsilon \|f(\sigma)\|_1$, where, $\|f(\sigma)\|_1$ is the ℓ_1 norm of the frequency vector $f(\sigma)$. The ApproxFreq(ϵ) problem is a well-studied and basic problem in data stream processing. Deterministic algorithms requiring $O(\epsilon^{-1} \log m)$ space are known for insert-only (i.e., no deletions) streams [9, 23, 21]. For streams with arbitrary insertions and deletions, the CR-Precis algorithm solves the problem ApproxFreq(ϵ) [14] using space $\tilde{O}(\epsilon^{-2}(\log^{-2}(1/\epsilon))(\log^2 n)(\log mn))$. A space lower bound of $\Omega(\epsilon^{-2} \log m)$ for deterministic algorithms is known for solving ApproxFreq(ϵ) over streams with frequency vector in $\{-m, \ldots, m\}^n$ [13].

Contributions. We present a simple paradigm for designing deterministic algorithms over data streams by using appropriately chosen lossless expander graphs. The paradigm consists of two steps, namely, identifying the expansion properties needed to solve the problem at hand, and, a low space representation of the expander graph (or an object that closely resembles it). We illustrate our paradigm by designing algorithms for k-sparsity testing and estimating item frequencies.

We first present a novel solution for deterministic k-sparsity tester for the frequency vector $f(\sigma)$ of a data stream σ, when, $f(\sigma) \in \{0, \ldots, m\}^n$. This technique, based on lossless expander graphs, requires space $O(k \cdot D(1/4, n, k))$, where, $D(\epsilon, n, r) = o(n)$ is the smallest known degree function in the construction of (k, ϵ)-lossless expanders. We subsequently improve upon this algorithm to present a space upper bound of $O(k(\log(n/k))(\log mn))$ bits. This improves the current upper bound of $O(k^2 \log n + k \log(mn))$ space [15] and nearly matches the space lower bound up to a logarithmic factor, which we show to be $\Omega(k \log(mn/k))$. We also use the expander graphs based approach to design a family of deterministic algorithms for ApproxFreq(ϵ), of which the CR-Precis algorithm [14] is a special case. The algorithms derived in this manner are slightly more efficient in space and update time than the CR-Precis algorithm.

Organization. The remainder of the paper is organized as follows. In Section 2, we consider the k-sparsity testing problem over data stream and and in Section 3 we consider the ApproxFreq(ϵ) problem. Finally, we conclude in Section **??**.

2 Testing k-sparsity

In this section, we design a deterministic k-sparsity tester for frequency vectors in $\{0, \ldots, m\}^n$ based on lossless expander graphs. We first present a space lower bound for k-sparsity testing of vectors over $\{0, \ldots, m\}^n$ in the data stream model.

Lemma 1. *For each value of $1 \leq k < n/2$, a deterministic k-sparsity tester for vectors over $\{0, \ldots, m\}^n$ requires $\Omega(k \log(mn/k))$ bits.*

Proof (Of Lemma 1). Let $k < n/2$. Suppose f, g are distinct k-sparse vectors such that $\|f\|_\infty \leq m/2$ and $\|g\|_\infty \leq m/2$. We first show that there exists $h \in \{-m, \ldots, m\}^n$ such that both $f + h$ and $g + h$ are non-negative and one of them is k-sparse and the other is not k-sparse. This would imply that any k-sparsity tester must map distinct k-sparse vectors in $\{0, \ldots, \lfloor m/2 \rfloor\}^n$ to distinct summaries. Since, the number of k-sparse vectors in $\{0, \ldots, \lfloor m/2 \rfloor\}^n$ is $\sum_{r=0}^{k} \binom{n}{k} (\lfloor m/2 \rfloor + 1)^r$, the space requirement would be at least the logarithm of this quantity, which is $O(k \log(mn/k))$.

Let S_f and S_g denote the set of coordinates of f and g respectively with non-zero entries. Let T be any set of size $k - |S_g|$ such that $T \cap (S_f \cup S_g) = \phi$. Such a T exists since, $k < n/2$. Denote by 1_T the characteristic vector of T. Let

$h = g + 1_T$. Then $g + h = 2g + 1_T$ and is k-sparse. Further, $f + h = f + g + 1_T$ and therefore,

$$|S_{f+g+1_T}| = |(S_f \cup S_g)| + |T| > |S_g| + 1 + |T| = k + 1$$

and so, $f + h$ is not k-sparse. □

2.1 Sparsity separator structure

We first define the (k, l)-*sparsity separator structure* that will be used later to test k-sparsity in sub-linear space.

Definition 1. *A (k, l)-sparsity separator structure, where, $k \leq l$, is a data structure that (a) supports updates corresponding to any stream σ over $[n]$, and, (b) supports a deterministic operation called* SeparateSparsity *that returns* TRUE *if the sparsity of $f(\sigma)$ is at most k and returns* FALSE *if the sparsity of $f(\sigma)$ is at least l.* □

There is an indeterminate region, namely, if the sparsity of $f(\sigma)$ is between $k+1$ and $l - 1$, then the function SeparateSparsity(f) is allowed to return either TRUE or FALSE.

Lossless expander graphs. We design a $(k, 2k)$ sparsity separator structure using lossless expander graphs. We first recall some standard concepts from expander graphs [19]. Let $G = (V_L, V_R, E, d)$ denote a left-regular bipartite graph where, $V_L = \{v_1, \ldots, v_n\}$ is the set of vertices in the left partition, $V_R = \{u_1, \ldots, u_r\}$ is the set of vertices in the right partition, E is the set of edges of G and d is the degree of each vertex in the left partition.

Definition 2 (Lossless Expanders [19].). *A left-regular bipartite graph $G = (V_L, V_R, E, d)$ is said to be be a $(K_{\max}, \epsilon)$-lossless expander if every set of $K \leq K_{\max}$ vertices from the left partition has at least $(1 - \epsilon)dK$ neighbors in V_R.* [1]

The work in [4] presents non-trivial, explicit constructions of lossless expanders using the zig-zag product of expanders.

Theorem 2 ([4]). *For any $\epsilon > 0$ and $r \leq n$, there is an explicit family of left-regular bipartite graphs with $|V_L| = n$, $|V_R| = r$ that is an $(c'\epsilon r/d, \epsilon)$-lossless expander with left degree $D(\epsilon, n, r) \leq (n/\epsilon r)^c$ for some constants $c, c' > 0$. The neighbors of any left vertex may be found in time $O(d \cdot \log^{O(1)}(n))$.* □

Denote by $R(\epsilon, n, k)$ the smallest value of r for which there is a known efficiently constructible (k, ϵ)-lossless expander with n left vertices and r right vertices. Theorem 2 is optimized for the case $r = \Theta(n)$, in which case, the degree $d = d(n)$ is constant. Our approach will be the following. We will be interested in (K, ϵ)-lossless expander graphs with r as small as possible. We use Theorem 2 to obtain

[1] More accurately, an expander is a family of bipartite graphs $\{G_n\}_{n \geq n_0}$, for some n_0, where, $K_{\max} = K_{\max}(n), \epsilon = \epsilon(n)$ and $d = d(n)$.

expander graphs with the desired lossless expansion property to give us the wire-frame of an algorithm for the problem. We then replace the expander by a more explicit and low-space construction of a bipartite graph G with a smaller value of r and that has the desired lossless expansion property.

A $(k, 2k)$-sparsity separator using lossless expander graphs. Given n and using Theorem 2, we consider a left-regular bipartite graph $G = (V_L, V_R, E, d)$ that is a $(2k, \epsilon = 1/4)$-lossless expander such that

$$|V_L| = n, |V_R| = r \text{ and left-degree } d = D(\epsilon, n, r) \ .$$

Keep $r = |V_R|$ integer counters denoted as an r-dimensional vector $g = [g_1, \ldots, g_r]$, where, g_s is the counter associated with vertex $u_s \in V_R$. All counters are initialized to 0. The counter g_s maintains the following sum over the data stream.

$$g_s = \sum_{(v_i, u_s) \in E} f_i(\sigma), \quad s = 1, 2, \ldots, r \ .$$

Alternatively, if we let B be the $r \times n$ matrix such that $A_{s,i} = 1$ if v_i is adjacent to u_s and $A_{s,i} = 0$ otherwise, then, $g = A(f(\sigma))$. The counters are easily updated corresponding to a stream update (pos, i, v) as follows:

$$g_s := g_s + v, \quad \forall s \in [r] \text{ such that } u_s \text{ is adjacent to } v_i \ .$$

Equivalently, in matrix notation, $g := g + A_i$, where, A_i is the column corresponding to vertex v_i. Since, the neighbors of any left vertex v_i can be computed in $d \cdot \text{polylog}(n)$ time and $d = D(\epsilon, n, r)$, the update can be performed in time $D(\epsilon, n, 4k) \cdot \text{polylog}(n)$.

A procedure for SeparateSparsity$(k, 2k)$ can be designed as follows. It first checks if g is not $(1 - 1/4)2dk = \frac{3dk}{2}$-sparse in which case it returns FALSE. Otherwise, the procedure returns TRUE.

procedure SeparateSparsity$(k, 2k)$
Data Structure: A $(k, 2k)$-sparsity separator structure.
if g is not $\left(\frac{3}{2}dk - 1\right)$-sparse **return** FALSE **else return** TRUE

We now show that the algorithm SeparateSparsity$(k, 2k)$ correctly solves the approximate sparsity problem with parameter $k, 2k$.

Lemma 2. *Algorithm* SeparateSparsity$(k, 2k)$ *correctly solves the problem* SeparateSparsity$(k, 2k)$.

Proof. Suppose that f is not $2k - 1$-sparse. Then, it has at least $2k$ non-zero entries. Let $S_f = \{v_i \in V_L \mid f_i > 0\}$. Then, $|S_f| \geq 2k$. Choose any subset $T \subset S_f$ such that $|T| = 2k$. Let $\Gamma(S)$ denote the neighbors of any set $T \subset V_L$. By property of the $(2k, 1/4)$-lossless expander graph, $(1-1/4)d(2k) \leq |\Gamma(T)| \leq 2dk$, that is, $1.5dk \leq |\Gamma(T)| \leq 2dk$. For each $s \in \Gamma(T)$, $g_s > 0$, since, g_s is the sum of the (positive) f_i's of those i's such that v_i is adjacent to u_s. Therefore, g

procedure SPARSITYTEST(k)
Input: Data Stream σ with frequency vector $f(\sigma) \in \{0, \ldots, m\}^n$.
Output: Returns TRUE if f is k-sparse and FALSE otherwise.
Data Structure: (a) A $(k, 2k)$-sparsity separator structure over $\{0, \ldots, m\}^n$, and,
(b) a $2k$-set structure over $\{-m, \ldots, m\}^n$ that supports operation RETRIEVEVECTOR($2k$) [15].
begin

1. **if** SEPARATESPARSITY$(k, 2k)$ = FALSE **return** FALSE
2. **else if** SPARSITY(RETRIEVEVECTOR$(2k)) \leq k$ **return** TRUE
3. **else return** FALSE

end.

Fig. 1. Procedure for testing k-sparsity

has at least $1.5dk$ non-zero entries and the algorithm returns FALSE. On the other hand, if f is k-sparse, $|\Gamma(S_f)| \leq kd$ and therefore g is kd-sparse and the algorithm returns TRUE. Hence the algorithm satisfies the properties of testing SEPARATESPARSITY$(k, 2k)$. □

The space requirement of Algorithm SEPARATESPARSITY consists of the r-dimensional vector g, each of whose entries is an integer between 0 and mn. Then, the space requirement is $O(R(\epsilon, n, 2k) \log(mn))$. The time requirement to process each stream update is $D(\epsilon, n, R) \cdot O(\log^{O(1)}(n))$.

2.2 Algorithm for testing *k*-sparsity

We now use the sparsity separator $(k, 2k)$ of Section 2.1 together with the k-set reconstruction procedure of [15] to design an algorithm for k-sparsity test.

We keep two data structures, namely, a $2k$-set structure for $2k$-set reconstruction as presented in [15] and a $(k, 2k)$-sparsity separator structure, presented in Section 2.1. Both structures are maintained independently and in parallel in the face of stream updates. The procedure k-SPARSITYTEST is presented in Figure 1 and is described as follows. It first uses the $(k, 2k)$ sparsity separator test on the r-dimensional vector g. If the approximate sparsity test returns FALSE, then, we know that f cannot be k-sparse. (Otherwise, the $(k, 2k)$-sparsity separator test would have returned TRUE, by definition.) However, if the sparsity separator procedure returns TRUE, then, f is $2k$-sparse (otherwise, the sparsity-separator$(k, 2k)$ would have returned FALSE). The reconstruction procedure of the $2k$-set structure [15] is then invoked to obtain f, and from f, we obtain its sparsity. If the sparsity is at most k, then the procedure returns TRUE, and otherwise returns FALSE. We summarize these properties and the space and time bounds in the following theorem.

Theorem 3. *There exists a k-sparsity tester for frequency vector of data stream in $\{0, \ldots, m\}^n$ using space $O(R(1/4, n, 2k) \log mn))$, where, $R = R(1/4, n, 2k)$ is the smallest value of r for which a $(2k, 1/4)$-lossless expander can be efficiently constructed with n vertices in the left vertex partition V_L. The time required to process each stream update is $O(D(1/4, n, R) \cdot \log^{O(1)}(n))$.* □

Improving the expander based sparsity test. The space requirement of the the expander based k-sparsity separator presented in Section 2 can be improved by using a different construction of an (approximate) expander graph than the one given by Theorem 2. The set of vertices adjacent to a given subset of vertices S in a graph is denoted as $\Gamma(S)$.

Lemma 3. *For any $n \geq 2$, $d > 3\log(ne/4k)$ and $r \geq 8kd$, there exist bipartite graphs $G = (V_L, V_R, E)$ with $|V_L| = n$, $|V_R| = r$ satisfying the following property: for any subset $S \subset V_L$ such that $|S| \leq k$, $|\Gamma(S)| \leq k$ and for any subset $S \subset V_L$ such that $|S| \geq 4k$, $|\Gamma(S)| > k$. Moreover, the bipartite graph can be succinctly represented by a string of size kd^2 bits. The adjacency of a vertex in the left partition may be computed in time $O(kd^2)$.*

Proof (Of Lemma 3). Let $V_L = \{1, 2, \ldots, n\}$ and $V_R = \{1, 2, \ldots, r\}$. Define d independently chosen random hash functions $h_1, \ldots, h_d$ each mapping $[n] \to [r]$. For $i \in V_L$ and $s \in V_R$, we say that there is an edge (i, s) provided there exists $j \in \{1, \ldots, d\}$ such that $h_j(i) = s$.

By construction, for any $S \subset V_L$ be a set of left vertices of size k, $|\Gamma(S)| \leq kd$. Now suppose $S \subset V_L$ and $|S| = 4k$. For $s \in V_R$, define an indicator variable y_s that is 1 if $u_s \in \Gamma(S)$ and 0 otherwise.

$$\mathsf{Pr}\{y_s = 1\} = 1 - (1 - |S|/r)^d = p \text{ (say)} \ .$$

Thus,

$$\frac{|S|d}{r} - \frac{d^2|S|^2}{2r^2} \leq p \leq \frac{kd}{r} \ .$$

Let

$$W_S = \sum_{s=1}^{r} y_s \ .$$

Therefore,

$$\mathsf{E}[W_S] = rp \geq kd - \frac{d^2k^2}{r} \ .$$

Further,

$$\begin{aligned}\mathsf{E}\left[W_S^2\right] &= \Big(\sum_{s=1}^{r} y_s\Big)^2 = \sum_{s=1}^{r} y_s + 2 \sum_{1 \leq s_1 < s_2 \leq r} y_{s_1} y_{s_2} \\ &= rp + 2\binom{r}{2} p^2 \ .\end{aligned}$$

Thus,

$$\begin{aligned}\sigma^2(W_S) = \mathbf{Var}\left[W_S\right] = \mathsf{E}\left[W_S^2\right] - (\mathsf{E}[W_S])^2 &= rp + 2\binom{r}{2}p^2 - (rp)^2 \\ &= rp - rp^2 \ .\end{aligned}$$

Therefore,

$$\sigma^2(W_S) \leq rp \leq kd \ .$$

If the hash functions $h_1, \ldots, h_d$ are each t-wise independent, then, the y_s's are at least t-wise independent. By Chernoff's bound for t-wise independent variables [24], we have,

$$\mathsf{Pr}\left\{|W_S - \mathsf{E}\left[W_S\right]| > T\right\} \leq \left(\frac{t \max(t, \sigma^2(W_S))}{e^{2/3}T^2}\right)^{t/2} \quad \textbf{[24]} \ .$$

Choose the degree of independence as $t = kd$ and let the deviation from the expectation be $T = \mathsf{E}\left[W_S\right] - (kd+1)$. Then,

$$T \geq rp - (kd+1) \geq 4kd - \frac{16k^2d^2}{r} \geq 2kd$$

by choosing $r \geq 8kd$. Substituting, we have

$$\mathsf{Pr}\left\{W_S \leq k\right\} \leq \mathsf{Pr}\left\{|W_S - \mathsf{E}\left[W_S\right]| > T\right\} \leq \left(\frac{(kd)(4kd)}{e^{2/3}(2kd)^2}\right)^{kd/2} = e^{-kd/3} \ .$$

Therefore,

$$\mathsf{Pr}\left\{W_S \leq k, \text{ for some } S \text{ s.t. } |S| = 4k\right\} \leq \binom{n}{4k} e^{-kd/3} \leq e^{k\ln(ne/4k) - kd/3} < 1$$

provided, $d \geq 3\ln(ne/4k)$ Thus,

$$\mathsf{Pr}\left\{\forall S, |S| = 4k, W_S > 4k\right\} > 0$$

This proves the existence of a bipartite graph with the properties as stated in the Lemma.

Such a bipartite graph may be generated as follows. The random seed length is $kd^2 \log n$ bits, since, each of the hash functions may be implemented as a degree $kd-1$ polynomial over a field of size $O(n)$. We iterate over the space of $kd^2 \log n$ bit strings, generate the corresponding bipartite graph and check for the property. If the property holds, then, the $kd^2 \log n$ bit seed is stored as the generator for the bipartite graph. The above proof assures us of the existence of such a seed. □

$(k, 4k)$-Sparsity separator. A $(k, 4k)$-sparsity separator can be designed based on a succinctly representable bipartite graph $G = (V_L, V_R, E)$ obtained using Lemma 3 such that $|V_L| = n, d = 4\log(ne/k)$ and $|V_R| = r = 16kd$. By Lemma 3, for any subset $S \subset V_L$, if $|S| \geq 4k$, then, $|\Gamma(S)| > 2k$ and if $|S| \leq k$, then, $|\Gamma(S)| \leq k$. A $(k, 4k)$-sparsity separator is designed as follows. Keep r counters, $g_1, \ldots, g_r$, one each corresponding to each right vertex $u_s \in V_R$; all initialized to 0. The counter g_s maintains the following sum: $g_s = \sum_{i:(v_i,u_s)\in E} f_i$.

Corresponding to update (pos, i, v) on the stream, the counters are updated as follows.

$$\textsc{Update}(pos, i, v): \quad g_s = g_s + v, \quad \forall s \text{ such that } (i, s) \in E \ .$$

The space requirement is $O(r) = O(k \log(n/k))$ counters of size $O(\log(mn))$ bits plus the succinct description length $O(kd^2) = O(k \log^2(n/k))$ bits. The time required to process each stream update of the form (pos, i, v) is to evaluate d polynomials of degree kd each to obtain the adjacency of vertex v_i; this requires time $O(kd^2) = O(k \log^2(n/k))$. The $(k, 4k)$-SeparateSparsity test is as follows.

procedure Bipartite-SeparateSparsity$(k, 4k)$
1. **if** g is not k-sparse **then** **return** FALSE **else** **return** TRUE.

Rephrasing Lemma 3, if f is k-sparse, then, g is k-sparse, and, if f is not $4k$-sparse, then, g is not k sparse. The problem of k-sparsity testing can now be readily solved as before. Keep a $(k, 4k)$-sparsity separator for vectors in $\{0, \ldots, m\}^n$ based on succinct bipartite graphs and a $4k$-set structure from [15]. The algorithm for k-sparsity testing is identical to that presented in Figure 1, with the only change being that the use of the $2k$-set structure is replaced by a $4k$-set structure. We therefore have the following theorem.

Theorem 4. *There exists a structure for testing k-sparsity of vectors in $\{0, \ldots, m\}^n$ updated coordinate-wise as a data stream using space $O(k \log(n/k) \log(mn))$. The time taken to process each coordinate-wise update is $O(k \log^2(n/k))$.* □

The space requirement of the succinct bipartite graph based k-sparsity tester is within a logarithmic factor of the space lower bound of $\Omega(k \log(mn/k))$ proved in Lemma 1.

3 Deterministic estimation of frequency vector

In this section, we present a novel, deterministic algorithm for approximating the frequency vector of a data stream based on the use of lossless expander graphs. Consider a $(2, \epsilon/2)$-lossless expander graph $G = (V_L, V_R, E, d)$, where, $V_L = \{v_1, \ldots, v_n\}$, $V_R = \{u_1, \ldots, u_r\}$. By Theorem 2, a $(2, \epsilon/2)$-lossless expander has $r = O(D(\epsilon, n, O(1))/\epsilon)$ and $d = D(\epsilon, n, r)$, where, $D(\epsilon, n, O(1))$ is the current best known degree function for an $(O(1), \epsilon)$-lossless expander given by Theorem 2. As before, we keep an integer counter g_s corresponding to each vertex $u_s \in V_R$. The counters are initialized to 0 and are updated corresponding to stream update exactly in the same manner as discussed in Section 2.1. The estimate $\hat{f}_i$ is the following.

$$\hat{f}_i = \frac{1}{d} \sum_{s:(v_i, u_s) \in E} g_s, \quad i \in [n] \ .$$

Lemma 4. $|\hat{f}_i - f_i(\sigma)| \leq \epsilon \|f(\sigma)\|_1$.

Proof. For simplicity, fix the input stream σ and let f denote $f(\sigma)$. Fix i. By property of $(2,\epsilon)$-lossless expander, for any $i, j \in [n]$, $i \neq j$,

$$|\Gamma(v_i) \cap \Gamma(v_j)| = 2d - |\Gamma(v_i) \cup \Gamma(v_j)| \leq 2d - (1-\epsilon/2)(2d) \leq \epsilon d \ . \qquad (1)$$

Therefore,

$$\begin{aligned}\sum_{s:(v_i,u_s)\in E} g_s &= \sum_{s:(v_i,u_s)\in E} \sum_{j:(v_j,u_s)\in E} f_j = \sum_{j=1}^{n} f_j |\Gamma(v_j) \cap \Gamma(v_i)| \\ &= df_i + \sum_{\substack{1\leq j\leq n \\ j\neq i}} f_j |\Gamma(v_j) \cap \Gamma(v_i)| = df_i + \sum_{\substack{1\leq j\leq n \\ j\neq i}} f_j(\epsilon d), \quad \text{by (1)} \ .\end{aligned}$$

Dividing by d, transposing and taking absolute values, we have,

$$\Big|\frac{1}{d} \sum_{s:(v_i,u_s)\in E} g_s - f_i\Big| \leq \Big|\sum_{j\neq i} \epsilon f_j\Big| \leq \epsilon \|f\|_1 - |f_i| \ .$$

Since, $\frac{1}{d}\sum_{s:(v_i,u_s)\in E} g_s = \hat{f}_i$, this proves the lemma. □

Theorem 2 can be applied by setting $r = O(\frac{1}{\epsilon} D(\epsilon, n, 1))$ and $d = D(\epsilon, n, r)$, thereby obtaining a (K,ϵ)-lossless expander, for some $K \geq 2$. We summarize this in the following theorem.

Theorem 5. *There exists a deterministic algorithm for solving* ApproxFreq(ϵ) *over a data stream using space* $O(R(\epsilon, n, 2)\log(mn))$. *The time taken to process each stream update is* $O(D(\epsilon, n, R)\log^{O(1)} n)$. □

An exercise along the lines of producing a succinctly representable bipartite graph using the probabilistic method instead of using the lossless expander family of Theorem 2 can be carried out (and has a slightly simpler argument). We state this in the following lemma.

Lemma 5. *There exists a bipartite graph* $G(n,\epsilon) = (V_L, V_R, E, d)$ *such that* $|V_L| = n$, $|V_R| = O((1/\epsilon^2)\log(n/\epsilon))$, $d = O(\log n)$ *such that the degree of every vertex in* V_L *is between* $(1-\epsilon)d$ *and* d *and for any* $v_i, v_j \in V_L$, $i \neq j$, *the number of common neighbors of* v_i *and* v_j *do not exceed* ϵd. *Moreover, such a bipartite graph can be succinctly represented using* $O((\log(n/\epsilon))(\log n))$ *bits. The neighbors of any vertex in* V_L *can be computed in time* $O(\log(n/\epsilon)(\log n))$. □

Note that the bipartite graph of Lemma 5 is not left-regular, but rather almost left-regular: $(1-\epsilon)d \leq \deg(v_i) \leq d$. The analysis of Lemma 4 goes through with a slight modification, since the division by d gives rise to a factor that lies between 1 and $1/(1-\epsilon)$, thereby, increasing the error factor by $O(\epsilon)$. Replacing the $(2,\epsilon/2)$-expander graph by the succinct bipartite graph $G(n,\epsilon/4)$ from Lemma 5 yields an algorithm for ApproxFreq(ϵ). This is summarized in the following theorem.

Theorem 6. *There exists a deterministic algorithm for solving* ApproxFreq(ϵ) *over a data stream using space* $O(\epsilon^{-2}\log(n/\epsilon)\log(mn))$ *bits. The time taken to process each stream update is* $O(\log(n/\epsilon)(\log n))$. □

The only known previous algorithm for deterministic estimation of frequency is the CR-Precis algorithm [14] which requires space $O(\epsilon^{-2}(\log^{-2}(1/\epsilon))(\log^2 n)\cdot(\log mn))$. The algorithm of Theorem 6 is slightly better in its space requirement than the CR-Precis algorithm [14] by a small poly-logarithmic factor.

It is interesting to note that the primes residue based structure used by the CR-Precis structure [15] is an explicit construction of a $(2, \epsilon/2)$-lossless expander as follows. For t distinct primes $p_1, \ldots, p_t$, we define the bipartite graph $G_{CR}(p_1, \ldots, p_t) = (V_L, V_R, E, d)$ where, $V_L = \{v_1, \ldots, v_n\}$ and $V_R = \{u_{j,l} \mid 1 \leq j \leq t \text{ and } 0 \leq l \leq p_j - 1\}$. There is an edge between left vertex v_i and right vertex $u_{j,l}$ if and only if $l = i \mod p_j$. The degree of each left vertex is by construction t, since, each number has exactly one residue respectively for $p_1, \ldots, p_t$. For any $1 \leq i < j \leq n$, each common neighbor $u_{j,l}$ of v_i and v_j means that $p_l | j - i$. If there are s distinct common neighbors, then, there are s distinct primes that divides $j - i$. Since, $j - i \leq n - 1$, and each $p_l \geq 2$, $s < \log n$. This shows that the graph is a $(2, \epsilon/2)$-lossless expander for $t = 2(\log n)/\epsilon$ and for any choice of primes $p_1, \ldots, p_t$. Since, $r = |V_R| = p_1 + \ldots + p_t$, r is minimized by choosing the first $2\log n/\epsilon$ primes as $p_1, \ldots, p_t$. The well-known prime number theorem then guarantees that $p_1 + \ldots + p_t = O(p_t t) = O(t^2 \ln t) = O((\log^2 n/\epsilon^2)\log((\log n)/\epsilon))$.

References

1. Noga Alon. "Perturbed identity matrices have high rank: proof and applications". Available from http://www.math.tau.ac.il/∼nogaa/identity.pdf, 2006.
2. P. Bose, E. Kranakis, P. Morin, and Y. Tang. "Bounds for Frequency Estimation of Packet Streams". In *Proc. of SIROCCO*, pp. 33–42, 2003.
3. Emmanuel Candès, Justin Romberg, and Terence Tao. "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information". *IEEE Trans. Inf. Theory*, 52(2):489–509, February 2006.
4. M. Capalbo, O. Reingold, S. Vadhan, and A. Wigderson. "Randomness conductors and constant degree lossless expanders". In *Proc. of ACM STOC*, 2002.
5. Moses Charikar, Kevin Chen, and Martin Farach-Colton. "Finding frequent items in data streams". In *Proc. (ICALP)*, pp. 693–703, 2002.
6. Bogdan Cheblus and Dariusz R. Kowalski. "Almost Optimal Explicit Selectors". In *Proc. (FCT) LNCS 3623*, pp. 270–280, 2005.
7. Graham Cormode and S. Muthukrishnan. "An Improved Data Stream Summary: The Count-Min Sketch and its Applications". *J. Algo.*, 55(1).
8. A. De Bonis, L. Gàsieniec, and U. Vaccaro. "Generalized framework for selectors with applications in optimal group testing". In *Proc. (ICALP)*, pp. 81–96, 2003.
9. E. D. Demaine, A. López-Ortiz, and J. I Munro. "Frequency estimation of internet packet streams with limited space". In *Proc. ESA*, pp. 348–360, 2002.
10. David Donoho. "Compressed sensing". *IEEE Trans. Inf. Theory*, 52(4):1289–1306, April 2006.
11. David Eppstein and Michael T. Goodrich. "Space-Efficient Straggler Identification in Round-Trip Data Streams". In *Proc. WADS*, 2007.

12. S. Ganguly. "Counting distinct items over update streams". In *Proc. of Int'l Symp. on Algo. Automata, Comp. (ISAAC)*, pp. 505–514, 2005.
13. S. Ganguly. "Lower bounds for frequency estimation over data streams". In *Proc. CSR (LNCS 5010)*, pp. 204–215, Moscow, June 2008.
14. S. Ganguly and Majumder A. "CR-precis: A Deterministic Summary Structure for Update Streams". In *Proc. ESCAPE, Springer LNCS 4614*, 2007.
15. S. Ganguly and A. Majumder. "Deterministic K-set Structure". In *Proc. ACM PODS*, pp. 280–289, 2006.
16. S. Ganguly and A. Majumder. "Deterministic K-set Structure". Manuscript under review. Available from http://www.cse.iitk.ac.in/users/sganguly, July 2006.
17. Anna Gilbert, Sudipto Guha, Piotr Indyk, Y. Kotidis, S. Muthukrishnan, and Martin Strauss. "Fast Small-space Algorithms for Approximate Histogram Maintenance". In *Proc. ACM STOC*, pp. 152–161, 2002.
18. Anna Gilbert, Martin Strauss, Joel Tropp, and Roman Vershynin. "One sketch for all: Fast algorithms for Compressed Sensing". In *Proc. ACM STOC*, 2007.
19. Shlomo Hoory, Nathan Linial, and Avi Wigderson. "Expander graphs and their applications". Draft of book, 2006.
20. Piotr Indyk. "Explicit Constructions for Compressed Sensing of Sparse Signals". In *Proc. ACM SODA*, 2008.
21. R.M. Karp, S. Shenker, and C.H. Papadimitriou. "A Simple Algorithm for Finding Frequent Elements in Streams and Bags". *ACM TODS*, 28(1):51–55, 2003.
22. Y. Minsky, A. Trachtenberg, and R. Zippel. "Set Reconciliation with Nearly Optimal Communication Complexity". *IEEE Trans. Inf. Theory*, 49(9):2213–2218.
23. J. Misra and Gries. D. "Finding repeated elements". *Sci. Comput. Programm.*, 2:143–152, 1982.
24. J. Schmidt, A. Siegel, and A. Srinivasan. "Chernoff-Hoeffding Bounds with Applications for Limited Independence". In *Proc. ACM SODA*, pp. 331–340, 1992.