Optimal Bounds for Johnson-Lindenstrauss Transforms and Streaming Problems with Sub-Constant Error

T.S. Jayram * David Woodruff †

Abstract

The Johnson-Lindenstrauss transform is a dimensionality reduction technique with a wide range of applications to theoretical computer science. It is specified by a distribution over projection matrices from $\mathbb{R}^n \to \mathbb{R}^k$ where $k \ll d$ and states that $k = O(\varepsilon^{-2} \log 1/\delta)$ dimensions suffice to approximate the norm of any fixed vector in $\mathbb{R}^d$ to within a factor of $1 \pm \varepsilon$ with probability at least $1 - \delta$. In this paper we show that this bound on k is optimal up to a constant factor, improving upon a previous $\Omega((\varepsilon^{-2} \log 1/\delta)/\log(1/\varepsilon))$ dimension bound of Alon. Our techniques are based on lower bounding the information cost of a novel one-way communication game and yield the first space lower bounds in a data stream model that depend on the error probability δ.

For many streaming problems, the most naïve way of achieving error probability δ is to first achieve constant probability, then take the median of $O(\log 1/\delta)$ independent repetitions. Our techniques show that for a wide range of problems this is in fact optimal! As an example, we show that estimating the ℓ_p-distance for any $p \in [0, 2]$ requires $\Omega(\varepsilon^{-2} \log n \log 1/\delta)$ space, even for vectors in $\{0, 1\}^n$. This is optimal in all parameters and closes a long line of work on this problem. We also show the number of distinct elements requires $\Omega(\varepsilon^{-2} \log 1/\delta + \log n)$ space, which is optimal if $\varepsilon^{-2} = \Omega(\log n)$. We also improve previous lower bounds for entropy in the strict turnstile and general turnstile models by a multiplicative factor of $\Omega(\log 1/\delta)$. Finally, we give an application to one-way communication complexity under product distributions, showing that unlike in the case of constant δ, the VC-dimension does not characterize the complexity when $\delta = o(1)$.

*IBM Almaden, jayram@almaden.ibm.com
†IBM Almaden, dpwoodru@us.ibm.com

1 Introduction

The Johnson-Linderstrauss transform is a fundamental dimensionality reduction technique with applications to many areas such as nearest-neighbor search [2, 25], compressed sensing [14], computational geometry [17], data streams [7, 24], graph sparsification [40], machine learning [32, 39, 43], and numerical linear algebra [18, 22, 37, 38]. It is given by a projection matrix that maps vectors in $\mathbb{R}^n$ to $\mathbb{R}^k$, where $k \ll d$, while seeking to approximately preserve their norm. The classical result states that $k = O(\frac{1}{\varepsilon^2} \log 1/\delta)$ dimensions suffice to approximate the norm of any fixed vector in $\mathbb{R}^n$ to within a factor of $1 \pm \varepsilon$ with probability at least $1 - \delta$. This is a remarkable result because the target dimension is *independent* of n. Because the transform is linear, it also preserves the pairwise distances of the vectors in this set, which is what is needed for most applications. The projection matrix is itself produced by a random process that is oblivious to the input vectors. Since the original work of Johnson and Lindenstrauss, it has been shown [1, 8, 21, 25] that the projection matrix could be constructed element-wise using the standard Gaussian distribution or even uniform ± 1 variables [1]. By setting the size of the target dimension $k = O(\frac{1}{\varepsilon^2} \log 1/\delta)$, the resulting matrix, suitably scaled, is guaranteed to approximate the norm of any single vector with failure probability δ.

Due to its algorithmic importance, there has been a flurry of research aiming to improve upon these constructions that address both the time needed to generate a suitable projection matrix as well as to produce the transform of the input vectors [2, 3, 4, 5, 33]. In the area of data streams, the Johnson-Lindenstrauss transform has been used in the seminal work of Alon, Matias and Szegedy [7] as a building block to produce *sketches* of the input that can be used to estimate norms. For a stream with poly(n) increments/decrements to a vector in $\mathbb{R}^n$, the size of the sketch can be made to be $O(\frac{1}{\varepsilon^2} \log n \log 1/\delta)$. To achieve even better update times, Thorup and Zhang [42], building upon the Count Sketch data structure of Charikar, Chen, and Farach-Colton [16], use an ultra-sparse transform to estimate the norm, but then have to

take a median of several estimators in order to reduce the failure probability. This is inherently non-linear but suggests the power of such schemes in addressing sparsity as a goal; in contrast, a single transform with constant sparsity per column fails to be an (ε, δ)-JL transform [20, 34].

In this paper, we consider the central lower bound question of Johnson Lindenstrauss transforms: how good is the upper bound on the target dimension of $k = O(\frac{1}{\varepsilon^2} \log 1/\delta)$ to approximate the norm of a fixed vector in $\mathbb{R}^n$? Alon [6] gave a near-tight lower bound of $\Omega(\frac{1}{\varepsilon^2}(\log 1/\delta)/\log(1/\varepsilon))$, leaving an asymptotic gap of $\log(1/\varepsilon)$ between the upper and lower bounds. In this paper, we close the gap and resolve the optimality of Johnson Lindenstrauss transforms by giving a lower bound of $k = \Omega(\frac{1}{\varepsilon^2} \log 1/\delta)$ dimensions. More generally, we show that any sketching algorithm for estimating the norm (whether linear or not) of vectors in $\mathbb{R}^n$ must use space at least $\Omega(\frac{1}{\varepsilon^2} \log n \log 1/\delta)$ to approximate the norm within a $1 \pm \varepsilon$ factor with a failure probability of at most δ. By a simple reduction, we show that this result implies the aforementioned lower bound on Johnson Lindenstrauss transforms.

Our results come from lower-bounding the information cost of a novel one-way communication complexity problem. One can view our results as a strengthening of the augmented-indexing problem [9, 10, 18, 28, 35] to very large domains. Our technique is far-reaching, implying the first lower bounds for the space complexity of streaming algorithms that depends on the error probability δ. In many cases, our results are tight. For instance, for estimating the ℓ_p-norm for any $p \geq 0$ in the turnstile model, we prove an $\Omega(\varepsilon^{-2} \log n \log 1/\delta)$ space lower bound for streams with poly(n) increments/decrements. This resolves a long sequence of work on this problem [26, 28, 44] and is simultaneously optimal in ε, n, and δ. For $p \in [0, 2]$, this matches the upper bound of [28]. Indeed, in [28] it was shown how to achieve $O(\varepsilon^{-2} \log n)$ space and constant probability of error. To reduce this to error probability δ, run the algorithm $O(\log 1/\delta)$ times in parallel and take the median. Surprisingly, this is optimal! For estimating the number of distinct elements in a data stream, we prove an $\Omega(\varepsilon^{-2} \log 1/\delta + \log n)$ space lower bound, improving upon the previous $\Omega(\log n)$ bound of [7] and $\Omega(\varepsilon^{-2})$ bound of [26, 44]. In [28, 29], an $O(\varepsilon^{-2} + \log n)$-space algorithm is given with constant probability of success. We show that if $\varepsilon^{-2} = \Omega(\log n)$, then running their algorithm in parallel $O(\log 1/\delta)$ times and taking the median of the results is optimal. On the other hand, we show that for constant ε and sub-constant δ, one can achieve $O(\log n)$ space, ruling out an $\Omega(\log n \log 1/\delta)$ bound. Similarly, we improve the known $\Omega(\varepsilon^{-2} \log n)$ bound for estimating the entropy in the turnstile model to $\Omega(\varepsilon^{-2} \log n \log 1/\delta)$, and we improve the previous $\Omega(\varepsilon^{-2} \log n/\log 1/\varepsilon)$ bound [28] for estimating the entropy in the strict turnstile model to $\Omega(\varepsilon^{-2} \log n \log 1/\delta/\log 1/\varepsilon)$. Entropy has become an important tool in databases as a way of understanding database design, enabling data integration, and performing data anonymization [41]. Estimating this quantity in an efficient manner over large sets is a crucial ingredient in performing this analysis (see the recent tutorial in [41] and the references therein).

Kremer, Nisan and Ron [30] showed the surprising theorem that for constant error probability δ, the one-way communication complexity of a function under product distributions coincides with the VC-dimension of the communication matrix for the function. We show that for sub-constant δ, such a nice characterization is not possible. Namely, we exhibit two functions with the same VC-dimension whose communication complexities differ by a multiplicative $\log 1/\delta$ factor.

Organization: In Section 2, we give preliminaries on communication and information complexity. In Section 3, we give our lower bound for augmented-indexing over larger domains. In Section 4, we give the improved lower bound for Johnson-Lindenstrauss transforms and the streaming and communication applications mentioned above.

2 Preliminaries

Let $[a, b]$ denote the set of integers $\{i \mid a \leq i \leq b\}$, and let $[n] = [1, n]$. Random variables will be denoted by upper case Roman or Greek letters, and the values they take by (typically corresponding) lower case letters. Probability distributions will be denoted by lower case Greek letters. A random variable X with distribution μ is denoted by $X \sim \mu$. If μ is the uniform distribution over a set $\mathcal{U}$, then this is also denoted as $X \in_R \mathcal{U}$.

2.1 One-way Communication Complexity Let $\mathcal{D}$ denote the input domain and $\mathcal{O}$ the set of outputs. Consider the two-party communication model, where Alice holds an input $x \in \mathcal{D}$ and Bob holds an input $y \in \mathcal{D}$. Their goal is to solve some relation problem $Q \subseteq \mathcal{D} \times \mathcal{D} \times \mathcal{O}$. For each $(x, y) \in \mathcal{D}^2$, the set $Q_{xy} = \{z \mid (x, y, z) \in Q\}$ represents the set of possible answers on input (x, y). Let $\mathcal{L} \subseteq \mathcal{D}^2$ be the set of legal or *promise* inputs, that is, pairs (x, y) such that $Q_{xy} \neq \emptyset$. Q is a (partial) function on $\mathcal{D}^2$ if for every (x, y), Q_{xy} has size at most 1. In a *one-way communication protocol* $\mathcal{P}$, Alice sends a single message to Bob, following which Bob outputs an answer in $\mathcal{O}$. The maximum length of Alice's message (in bits) over all all inputs is the

communication cost of the protocol $\mathcal{P}$. The protocol is allowed to be randomized in which the players have *private* access to an unlimited supply of random coins. The protocol solves the communication problem Q if the answer on any input $(x, y) \in \mathcal{L}$ belongs to Q_{xy} with failure probability at most δ. Note that the protocol is legally defined for all inputs, however, no restriction is placed on the answer of the protocol for non-promise inputs. The *one-way communication complexity* of Q, denoted by $R^{\rightarrow}_{\delta}(Q)$, is the minimum communication cost of a protocol for Q with failure probability at most δ. A related complexity measure is distributional complexity $D^{\rightarrow}_{\mu,\delta}(Q)$ with respect to a distribution μ over $\mathcal{L}$. This is the cost of the best *deterministic* protocol for Q that has error probability at most δ when the inputs are drawn from distribution μ. By Yao's lemma, $R^{\rightarrow}_{\delta}(Q) = \max_{\mu} D^{\rightarrow}_{\mu,\delta}(Q)$. Define $R^{\rightarrow,\|}_{\delta}(Q) = \max_{\text{product } \mu} D^{\rightarrow}_{\mu,\delta}(Q)$, where now the maximum is taken only over product distributions μ on $\mathcal{L}$ (if no such distribution exists then $R^{\rightarrow,\|}_{\delta}(Q) = 0$). Here, by product distribution, we mean that Alice and Bob's inputs are chosen independently.

Another restricted model of communication is *simultaneous* or *sketch*-based communication, where Alice and Bob each send a message (sketch) depending only on her/his own input (as well as private coins) to a referee. The referee then outputs the answer based on the two sketches. The communication cost is the maximum sketch sizes (in bits) of the two players.

Note: When δ is fixed (say $1/4$) we will usually suppress it in the terms involving δ.

2.2 Information Complexity We summarize basic properties of entropy and mutual information (for proofs, see Chapter 2 of [19]).

PROPOSITION 2.1.

1. *Entropy Span: If X takes on at most s values, then* $0 \le \mathrm{H}(X) \le \log s$.
2. $\mathrm{I}(X : Y) \ge 0$, *i.e.,* $\mathrm{H}(X \mid Y) \le \mathrm{H}(X)$.
3. *Chain rule:* $\mathrm{I}(X_1, X_2, \ldots, X_n : Y \mid Z) = \sum_{i=1}^{n} \mathrm{I}(X_i : Y \mid X_1, X_2, \ldots X_{i-1}, Z)$
4. *Subadditivity:* $\mathrm{H}(X, Y \mid Z) \le \mathrm{H}(X \mid Z) + \mathrm{H}(Y \mid Z)$, *and equality holds if and only if X and Y are independent conditioned on Z.*
5. *Fano's inequality: Let A be a "predictor" of X, i.e, there is a function g such that* $\Pr[g(A) = X] \ge 1 - \delta$ *for some $\delta < 1/2$. Let $\mathcal{U}$ denote the support of X, where $|\mathcal{U}| \ge 2$. Then,* $\mathrm{H}(X \mid A) \le \delta \log(|\mathcal{U}| - 1) + h_2(\delta)$, *where $h_2(\delta) \triangleq \delta \log \frac{1}{\delta} + (1 - \delta) \log \frac{1}{1-\delta}$ is the binary entropy function.*

Recently, the *information complexity* paradigm, in which the information about the inputs revealed by the message(s) of a protocol is studied, has played a key role in resolving important communication complexity problems [11, 13, 15, 23, 27]. We do not need the full power of these techniques in this paper. There are several possible definitions of information complexity that have been considered depending on the application. Our definition is tuned specifically for one-way protocols, similar in spirit to [11] (see also [13]).

DEFINITION 2.1. *Let $\mathcal{P}$ be a one-way protocol. Suppose μ is a distribution over its input domain $\mathcal{D}$. Let Alice's input X be chosen according to μ Let A be the random variable denoting Alice's message on input $X \sim \mu$; A is a function of X and Alice's private coins. The* information cost *of $\mathcal{P}$ under μ is defined to be* $\mathrm{I}(X : A)$.

The one-way information complexity *of a problem Q w.r.t. μ and δ, denoted by* $\mathrm{IC}^{\rightarrow}_{\mu,\delta}(Q)$, *is defined to be the minimum information cost of a one-way protocol under μ that solves Q with failure probability at most δ.*

By the entropy span bound (Proposition 2.1),

$$\mathrm{I}(X : A) = \mathrm{H}(A) - \mathrm{H}(A \mid X) \le \mathrm{H}(A) \le |A|,$$

where $|A|$ denotes the length of Alice's message.

PROPOSITION 2.2. *For every μ,*

$$R^{\rightarrow}_{\delta}(Q) \ge \mathrm{IC}^{\rightarrow}_{\mu,\delta}(Q).$$

2.3 JL Transforms

DEFINITION 2.2. *A random family $\mathcal{F}$ of $k \times n$ matrices A, together with a distribution μ on $\mathcal{F}$, forms a Johnson-Lindenstrauss transform with parameters ε, δ, or (ε, δ)-JLT for short, if for any fixed vector $x \in \mathbb{R}^n$,*

$$\Pr_{A \sim \mu}[(1 - \varepsilon)\|x\|_2^2 \le \|Ax\|_2^2 \le (1 + \varepsilon)\|x\|_2^2] \ge 1 - \delta.$$

We say that k is the dimension of the transform.

3 Augmented Indexing on Large Domains

Let $\mathcal{U} \cup \{\perp\}$, where $\perp \notin \mathcal{U}$, denote the input domain for some universe $\mathcal{U}$ which is sufficiently large. Consider the decision problem known as *augmented indexing* with respect to $\mathcal{U}$ ($\mathrm{IND}^{\mathsf{a}}_{\mathcal{U}}$) as shown in Figure 1.

Let μ be the uniform distribution on $\mathcal{U}$ and let μ^N denote the product distribution on $\mathcal{U}^N$.

THEOREM 3.1. *Suppose the failure probability $\delta \le \frac{1}{4|\mathcal{U}|}$. Then,*

$$\mathrm{IC}^{\rightarrow}_{\mu^N,\delta}(\mathrm{IND}^{\mathsf{a}}_{\mathcal{U}}) \ge N \log |\mathcal{U}|/2$$

Problem: $\textsc{Ind}^{\mathsf{a}}_{\mathcal{U}}$

Promise Inputs:
Alice gets $\mathbf{x} = (x_1, x_2, \ldots, x_N) \in \mathcal{U}^N$.

Bob gets $\mathbf{y} = (y_1, y_2, \ldots, y_N) \in (\mathcal{U} \cup \{\perp\})^N$ such that for some (unique) i:

1. $y_i \in \mathcal{U}$,
2. $y_k = x_k$ for all $k < i$,
3. $y_{i+1} = y_{i+2} = \cdots = y_N = \perp$

Output:
Does $x_i = y_i$ (YES/NO)?

Figure 1: Communication problem $\textsc{Ind}^{\mathsf{a}}_{\mathcal{U}}$

Proof. The proof uses some of the machinery developed for direct sum theorems in information complexity.

Let $\mathbf{X} = (X_1, X_2, \ldots, X_N) \sim \mu^N$, and let A denote Alice's message on input $\mathbf{X}$ in a protocol for $\textsc{Ind}^{\mathsf{a}}_{\mathcal{U}}$ with failure probability δ. By the chain rule for mutual infomation (Proposition 2.1),

$$\begin{aligned}\mathrm{I}(\mathbf{X} : A) &= \sum_{i=1}^{N} \mathrm{I}(X_i : A \mid X_1, X_2, \ldots, X_{i-1}) \\ &= \sum_{i=1}^{N} \mathrm{H}(X_i \mid X_1, X_2, \ldots, X_{i-1}) \\ &\quad - \mathrm{H}(X_i \mid A, X_1, X_2, \ldots, X_{i-1}) \qquad (3.1)\end{aligned}$$

Fix a coordinate i within the sum in the above equation. By independence, the first expression: $\mathrm{H}(X_i \mid X_1, X_2, \ldots, X_{i-1}) = \mathrm{H}(X_i) = \log|\mathcal{U}|$. For the second expresson, fix an element $a \in \mathcal{U}$ and let $\mathbf{Y}_a$ denote $(X_1, X_2, \ldots, X_{i-1}, a, \perp, \ldots, \perp)$. Note that when Alice's input is $\mathbf{X}$, the input that Bob is holding is *exactly* $\mathbf{Y}_a$ for some i and a. Let $B(A, \mathbf{Y}_a)$ denote Bob's output on Alice's message A. Then

$$\Pr[B(A, \mathbf{Y}_a) = 1 \mid X_i = a] \geq 1 - \delta$$

and for every $a' \neq a$,

$$\Pr[B(A, \mathbf{Y}_{a'}) = 0 \mid X_i = a] \geq 1 - \delta$$

Therefore, by the union bound,

$$\Pr\Big[B(A, \mathbf{Y}_a) = 1 \wedge \bigwedge_{a' \neq a} B(A, \mathbf{Y}_{a'}) = 0 \mid X_i = a\Big]$$

is at least $1 - \delta|\mathcal{U}| \geq \frac{3}{4}$. Thus, there is a predictor for X_i using $X_1, X_2, \ldots, X_{i-1}$ and A with failure probability at most $1/4$. By Fano's inequality,

$$\begin{aligned}&\mathrm{H}(X_i \mid A, X_1, X_2, \ldots, X_{i-1}) \\ &\quad \leq \frac{1}{4}\log(|\mathcal{U}| - 1) + h_2(1/4) \\ &\quad \leq \frac{1}{2}\log(|\mathcal{U}|),\end{aligned}$$

since $|\mathcal{U}|$ is sufficiently large. Substituting in (3.1), we conclude

$$\mathrm{I}(\mathbf{X} : A) \geq N \log(|\mathcal{U}|)/2$$

COROLLARY 3.1. *Let $|\mathcal{U}| = 1/4\delta$. Then $R^{\rightarrow}_{\delta}(\textsc{Ind}^{\mathsf{a}}_{\mathcal{U}}) = \Omega(N \log 1/\delta)$.*

REMARK 3.1. *Consider a variant of $\textsc{Ind}^{\mathsf{a}}_{\mathcal{U}}$ where for the index i of interest, Bob does not get to see all of the prefix $x_1, x_2, \ldots, x_{i-1}$ of $\mathbf{x}$. Instead, for every such i, there is a subset $J_i \subseteq [i-1]$ depending on i such that he gets to see only x_k for $k \in J_i$. In this case, he has even less information than what he had for $\textsc{Ind}^{\mathsf{a}}_{\mathcal{U}}$ so every protocol for this problem is also a protocol for $\textsc{Ind}^{\mathsf{a}}_{\mathcal{U}}$. Therefore, the one-way communication lower bound of Corollary 3.1 holds for this variant.*

REMARK 3.2. *Now, consider the standard indexing problem* IND *where Bob gets an index i and a single element y, and the goal is to determine whether $x_i = y$. This is equivalent to the setting of the previous remark where $J_i = \emptyset$ for every i. The proof of Theorem 3.1 can be adapted to show that $R^{\rightarrow,\|}_{\delta}(\textsc{Ind}) = \Omega(N \log 1/\delta)$ for $|\mathcal{U}| = \frac{1}{8\delta}$. Let μ be the distribution where Alice gets $\mathbf{X}$ uniformly chosen in $\mathcal{U}^N$ and Bob's input (I, Y) is uniformly chosen in $[N] \times \mathcal{U}$. As in the proof of the theorem, let A be the message sent by Alice on input $\mathbf{X}$. Let δ_i denote the expected error of the protocol conditioned on $I = i$. By an averaging argument, for at least half the indices i, $\delta_i \leq 2\delta$. Fix such an i. Look at the last expression bounding the information cost in* (3.1)*. Using $H(X_i \mid X_1, X_2, \ldots, X_{i-1}) \leq H(X_i \mid A)$ and then proceeding as before, there exists an estimator β_i such that*

$$\Pr[\beta_i(A) \neq X_i \mid I = i] \leq |\mathcal{U}|\delta_i \leq 2|\mathcal{U}|\delta \leq \tfrac{1}{4},$$

implying that $\mathrm{I}(X_i : A) \geq (1/2)\log(|\mathcal{U}|)$. The lower bound follows since there are at least $N/2$ such indices.

3.1 An encoding scheme Let $\Delta(\mathbf{x}, \mathbf{y}) \triangleq |\{i \mid x_i \neq y_i\}|$ denote the Hamming distance between two vectors $\mathbf{x}, \mathbf{y}$ over some domain. We present an encoding scheme that transforms the inputs of $\textsc{Ind}^{\mathsf{a}}_{\mathcal{U}}$ into well-crafted gap instances of the Hamming distance problem. This

will be used in the applications to follow. The proof uses well-known machinery but is somewhat technical, therefore, we postpone it to the appendix.

LEMMA 3.1. *Consider the problem* $\text{IND}^{\mathsf{a}}_{\mathcal{U}}$ *on length* $N = bm$, *where* $m = \frac{1}{4\varepsilon^2}$ *is odd and* b *is some parameter. Let* $\alpha \geq 2$ *denote a decay factor. Let* $(\mathbf{x}, \mathbf{y})$ *be a promise input to the problem to* $\text{IND}^{\mathsf{a}}_{\mathcal{U}}$. *Then there exist encoding functions* $\mathbf{x} \rightsquigarrow \mathbf{u} \in \{0,1\}^n$ *and* $\mathbf{y} \rightsquigarrow \mathbf{v} \in \{0,1\}^n$, *where* $n = O(\alpha^b \cdot \frac{1}{\varepsilon^2} \cdot \log 1/\delta)$, *depending on a shared random string* $\mathbf{s}$ *that satisfy the following: suppose the index* i *(which is determined by* $\mathbf{y}$*) for which the players need to determine whether* $x_i = y_i$ *belongs to* $[(p-1)m+1, pm]$, *for some* p. *Then,* $\mathbf{u}$ *can be written as* $(\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3) \in \{0,1\}^{n_1} \times \{0,1\}^{n_2} \times \{0,1\}^{n_3}$ *and* $\mathbf{v}$ *as* $(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3) \in \{0,1\}^{n_1} \times \{0,1\}^{n_2} \times \{0,1\}^{n_3}$ *such that:*

1. $n_2 = n \cdot \alpha^{-p}(\alpha - 1)$ *and* $n_3 = n \cdot \alpha^{-p}$;
2. *each of the* $\mathbf{u}_i$*'s and* $\mathbf{v}_i$*'s have exactly half of their coordinates set to 1;*
3. $\Delta(\mathbf{u}_1, \mathbf{v}_1) = 0$ *and* $\Delta(\mathbf{u}_3, \mathbf{v}_3) = n_3/2$;
4. *if* $(\mathbf{x}, \mathbf{y})$ *is a* NO *instance, then with probability at least* $1 - \delta$,
$$\Delta(\mathbf{u}_2, \mathbf{v}_2) \geq n_2(\tfrac{1}{2} - \tfrac{\varepsilon}{3});$$
5. *if* $(\mathbf{x}, \mathbf{y})$ *is a* YES *instance, then with probability at least* $1 - \delta$,
$$\Delta(\mathbf{u}_2, \mathbf{v}_2) \leq n_2(\tfrac{1}{2} - \tfrac{2\varepsilon}{3}).$$

4 Applications

Throughout we assume that $n^{1-\gamma} \geq \frac{1}{\varepsilon^2} \log 1/\delta$ for an arbitrarily small constant $\gamma > 0$. For several of the applications below, the bounds will be stated in terms of communication complexity which can be translated naturally to memory lower bounds for analogous streaming problems.

4.1 Approximating the Hamming Distance

Consider the problem HAM where Alice gets $\mathbf{x} \in \{0,1\}^n$, Bob gets $\mathbf{y} \in \{0,1\}^n$, and their goal is to produce a $1 \pm \varepsilon$-approximation of $\Delta(x, y)$.

THEOREM 4.1. $R^{\rightarrow}_{\delta}(\text{HAM}) = \Omega(\frac{1}{\varepsilon^2} \cdot \log n \cdot \log 1/\delta)$

Proof. We reduce $\text{IND}^{\mathsf{a}}_{\mathcal{U}}$ to HAM using the encoding given in Lemma 3.1 with $\alpha = 2$ so that $n_2 = n_3 = n \cdot 2^{-p}$. With probability at least $1 - \delta$, the YES instances are encoded to have Hamming distance at most $n \cdot 2^{-p}(1 - \frac{2\varepsilon}{3})$ while the NO instances have distance at least $n \cdot 2^{-p}(1 - \frac{\varepsilon}{3})$. Their ratio is at least $1 + \varepsilon/3$. Using a protocol for HAM with approximation factor $1 + \varepsilon/3$ and failure probability δ, we can distinguish the two cases with probability at least $1 - 2\delta$.

Since we assume that $\frac{1}{\varepsilon^2} \cdot \log 1/\delta < n^{1-\gamma}$ for a constant $\gamma > 0$, we can indeed set $b = \Omega(\log n)$, as needed here to fit the vectors into n coordinates. Now apply Corollary 3.1 to finish the proof.

4.2 Estimating ℓ_p-distances

Since $\Delta(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_p^p$, for $\mathbf{x}, \mathbf{y} \in \{0,1\}^n$, Theorem 4.1 immediately yields the following for any constant p:

THEOREM 4.2. *The one-way communication complexity of the problem of approximating the* $\|\cdot\|_p$ *difference of two vectors of length* n *to within a factor* $1 + \varepsilon$ *with failure probability at most* δ *is* $\Omega(\frac{1}{\varepsilon^2} \cdot \log n \cdot \log 1/\delta)$.

4.3 JL Transforms

THEOREM 4.3. *Any* (ε, δ)*-JLT* (F, μ) *has dimension* $\Omega(\frac{1}{\varepsilon^2} \log 1/\delta)$.

Proof. The public-coin one-way communication complexity, that is, the one-way communication complexity in which the parties additionally share an infinitely long random string and denoted $R^{\rightarrow,pub}_{\delta}$, is at least $R^{\rightarrow}_{\delta} - O(\log I)$, where I is the sum of input lengths to the two parties [30]. By Theorem 4.2,

$$\begin{aligned} R^{\rightarrow,pub}_{\delta}(\ell_2) &= \Omega\left(\frac{1}{\varepsilon^2} \log n \log 1/\delta\right) - O(\log n) \\ &= \Omega\left(\frac{1}{\varepsilon^2} \log n \log 1/\delta\right). \end{aligned}$$

Consider the following public-coin protocol for ℓ_2. The parties use the public-coin to agree upon a $k \times n$ matrix A sampled from $\mathcal{F}$ according to μ. Alice computes $A\mathbf{x}$, rounds each entry to the nearest additive multiple of $\varepsilon/(2\sqrt{k})$, and send the rounded vector $\tilde{A\mathbf{x}}$ to Bob. Bob then computes $A\mathbf{y}$, and outputs $\|\tilde{A\mathbf{x}} - A\mathbf{y}\|$. By the triangle inequality,

$$\begin{aligned} \|A\mathbf{y} - A\mathbf{x}\| - \|\tilde{A\mathbf{x}} - A\mathbf{x}\| &\leq \|\tilde{A\mathbf{x}} - A\mathbf{y}\| \\ &\leq \|A\mathbf{y} - A\mathbf{x}\| + \|\tilde{A\mathbf{x}} - A\mathbf{x}\|, \end{aligned}$$

or using the definition of $\tilde{A\mathbf{x}}$,

$$\|A\mathbf{y} - A\mathbf{x}\| - \frac{\varepsilon}{2} \leq \|\tilde{A\mathbf{x}} - A\mathbf{y}\| \leq \|A\mathbf{y} - A\mathbf{x}\| + \frac{\varepsilon}{2}.$$

With probability $\geq 1 - \delta$, we have $\|A(\mathbf{y} - \mathbf{x})\|^2 = (1 \pm \varepsilon)\|\mathbf{y} - \mathbf{x}\|^2$, or $\|A\mathbf{y} - A\mathbf{x}\| = (1 \pm \varepsilon/2)\|\mathbf{y} - \mathbf{x}\|$. Using that $\|\mathbf{y} - \mathbf{x}\| \geq 1$ in Theorem 4.2 if $\|\mathbf{y} - \mathbf{x}\| \neq 0$, we have $\|\tilde{A\mathbf{x}} - A\mathbf{y}\| = (1 \pm \varepsilon)\|\mathbf{x} - \mathbf{y}\|$. Hence,

$$kB = \Omega\left(\frac{1}{\varepsilon^2} \log n \log 1/\delta\right),$$

where B is the maximum number of bits needed to describe an entry of $\tilde{A}\mathbf{x}$. With probability at least $1-\delta$, $\|A\mathbf{x}\|^2 = (1\pm\varepsilon)\|\mathbf{x}\|^2$, and so using that $\mathbf{x} \in \{0,1\}^n$, no entry of $A\mathbf{x}$ can be larger than $2n$. By rescaling δ by a constant, this event also occurs, and so $B = O(\log n + \log 1/\varepsilon + \log k)$. Since we assume that $n \geq \frac{1}{\varepsilon^2}\log n \log 1/\delta$, we have $B = O(\log n)$, and so $k = \Omega\left(\frac{1}{\varepsilon^2}\log 1/\delta\right)$, finishing the proof.

4.4 Estimating Distinct Elements We improve the lower bound for estimating the number F_0 of distinct elements in an insertion-only data stream up to a $(1\pm\varepsilon)$-factor with probability at least $1-\delta$. We let n be the universe size, that is, the total possible number of distinct elements.

THEOREM 4.4. *Any 1-pass streaming algorithm that outputs a $(1\pm\varepsilon)$-approximation to F_0 in an insertion-only stream with probability at least $1-\delta$ must use $\Omega(\varepsilon^{-2}\log 1/\delta + \log n)$ bits of space.*

REMARK 4.1. *This improves the previous $\Omega(\varepsilon^{-2} + \log n)$ lower bound of [7, 26, 44].*

Proof. It is enough to show an $\Omega(\frac{1}{\varepsilon^2}\cdot\log 1/\delta)$ bound since the $\Omega(\log n)$ bound is in [7]. We reduce $\text{IND}^{\mathsf{a}}_{\mathcal{U}}$ to approximating F_0 in a stream. Apply Lemma 3.1 with $\alpha = 2$ and $b = 1$ to obtain $\mathbf{u}$ and $\mathbf{v}$ of length $k = O(\frac{1}{\varepsilon^2}\cdot\log 1/\delta)$. With $b = p = 1$, with probability at least $1-\delta$, the Hamming distance for NO instances is at least $\frac{k}{2}(1-\frac{\varepsilon}{3})$ while for the YES instances it is at most $\frac{k}{2}(1-\frac{2\varepsilon}{3})$.

Alice inserts a token i corresponding to each i such that $u_i = 1$. Bob does the same w.r.t. v_i. Since the Hamming weights of $\mathbf{u}$ and $\mathbf{v}$ are exactly half, by a simple calculation, $2F_0 = \Delta(\mathbf{u},\mathbf{v}) + k$. Thus, there is a gap of at least $1+\Theta(\varepsilon)$.

REMARK 4.2. *The best known upper bound for estimating F_0 in an insertion-only stream is $O(\varepsilon^{-2}+\log n)$ bits of space [29], and this holds with constant probability. Naïvely repeating this $O(\log 1/\delta)$ times and taking the median would give space $O(\varepsilon^{-2}\log 1/\delta + \log n \log 1/\delta)$, which matches our lower bound unless $\varepsilon^{-2} = o(\log n)$. However, in this case it is possible to improve this naïve upper bound with a more careful algorithm. Here we sketch a simple way to achieve $O(\log n)$ space with error probability $\delta = O(\log\log n/\log n)$ whenever ε is constant. Notice that this rules out the possibility of proving an $\Omega(\log n \log 1/\delta)$ bound. We leave a finer analysis of the upper bound to future work.*

To do this, the algorithm of [29] has a subroutine ROUGHESTIMATOR *which provides an $O(1)$-approximation to F_0 at every point in the stream using $O(\log n)$ space. Without increasing the space by more than a constant factor, it is possible to achieve error probability $1 - 1/\text{poly}(\log n)$ for any polynomial, by replacing the set $[3] = \{1,2,3\}$ in lines 2-5 of Figure 2 of [29] with a set $[C]$ for a larger constant $C > 0$.*

Next, we combine this constant-factor approximation with the second algorithm of [12], which, given such a constant-factor approximation, has space $O(\log n)$ for constant ε. Importantly, in their analysis they have $O(\varepsilon^{-2})$ pairwise-independent hash functions h_i and maintain for each i the maximum number of trailing zeros of any item j for which $h_i(j) = 0$. By Chebyshev's inequality, their argument shows with constant probability, this number can be used to obtain a $(1\pm\varepsilon)$-approximation to F_0. This contributes an additive $O(\varepsilon^{-2}\log\log n)$ in their space bound. Since ε is a constant, we can instead afford to have $O(\log n/\log\log n)$ such hash functions h_i (instead of ε^{-2}) and still maintain $O(\log n)$ space. It now follows that with probability $1 - O(\log\log n/\log n)$, the output is a $(1\pm\varepsilon)$-approximation, as desired.

4.5 Estimating Entropy Our technique improves the lower bound for additively estimating the entropy of a stream. To capture this, the entropy difference of two n-dimensional vectors x and y is the problem of computing

$$H(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{n} \frac{|x_i - y_i|}{\|\mathbf{x}-\mathbf{y}\|_1} \log_2 \frac{\|\mathbf{x}-\mathbf{y}\|_1}{|x_i - y_i|}.$$

As usual with entropy, if $x_i - y_i = 0$, the corresponding term in the above sum is 0.

THEOREM 4.5. *The one-way communication complexity of the problem of estimating the entropy difference up to an additive ε with probability at least $1-\delta$ is $\Omega(\varepsilon^{-2}\log n \log 1/\delta)$.*

REMARK 4.3. *This improves the previous $\Omega(\varepsilon^{-2}\log n)$ lower bound implied by the work of [28].*

Proof. We reduce from HAM. Since the input vectors $\mathbf{x},\mathbf{y}$ to HAM are in $\{0,1\}^n$, $\|\mathbf{x}-\mathbf{y}\|_1 = \Delta(x,y)$. Also, if $x_i = y_i$, then the contribution to the entropy is 0. Otherwise, the contribution is $\frac{\log_2 \Delta(x,y)}{\Delta(x,y)}$. Hence, $H(\mathbf{x},\mathbf{y}) = \log_2 \Delta(x,y)$, or $\Delta(x,y) = 2^{H(\mathbf{x},\mathbf{y})}$. Given an approximation $\tilde{H}(\mathbf{x},\mathbf{y})$ with $|\tilde{H}(\mathbf{x},\mathbf{y}) - H(\mathbf{x},\mathbf{y})| \leq \varepsilon$ and with probability at least $1-\delta$,

$$\begin{aligned}
(1-\Theta(\varepsilon))\Delta(x,y) &\leq 2^{-\varepsilon}\Delta(x,y) \\
&\leq 2^{\tilde{H}(\mathbf{x},\mathbf{y})} \\
&\leq 2^{\varepsilon}\Delta(x,y) \\
&\leq (1+\Theta(\varepsilon))\Delta(x,y).
\end{aligned}$$

so one obtains a $(1 \pm \Theta(\varepsilon))$-approximation to HAM with the same probability. The lower bound now follows from Theorem 4.1.

Entropy estimation has also been studied in the strict turnstile model of streaming, in which one has a stream of tokens that can be inserted or deleted, and the number of tokens of a given type at any point in the stream is non-negative. We can show an $\Omega(\varepsilon^{-2} \log n \log 1/\delta / \log 1/\varepsilon)$ as follows.

We apply Lemma 3.1 with $\alpha = \varepsilon^{-2}$, $b = O(\log n / \log(1/\varepsilon))$ to obtain $\mathbf{u}$ and $\mathbf{v}$. For each coordinate i in $\mathbf{u}$, Alice inserts a token i if the value at the coordinate equals of 0 and a token of $n + i$ if the value equals 1. Let $\mathbf{u} = (\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3)$ and $\mathbf{v} = (\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$. Bob can compute the split of $\mathbf{v}$ because he can compute n_1, n_2, and n_3 based on p (which itself depends on i). Bob deletes all the tokens corresponding to coordinates in $\mathbf{u}_1$, which is possible because $\mathbf{v}_1 = \mathbf{u}_1$. For coordinates in $\mathbf{v}_2$ he mimics Alice's procedure i.e. a token i for 0 and a token $n+i$ for 1. For $\mathbf{v}_3$ he does nothing. The number of tokens equal $n_3 + 2n_2 = n \cdot \varepsilon^{-2p}(2\varepsilon^{-2} - 1)$. The tokens corresponding to $\mathbf{u}_3$ appear exactly once. For every coordinate where $\mathbf{u}_2$ and $\mathbf{v}_2$ differ, the stream consists of 2 distinct tokens, whereas for each of the remaining coordinates the stream consists of a token appearing twice. Therefore, number of tokens appearing exactly once equals $n_3 + 2\Delta(\mathbf{u}_2, \mathbf{v}_2) = n\varepsilon^{-2p} + 2\Delta(\mathbf{u}_2, \mathbf{v}_2)$. The number of tokens appearing twice equals $n_2 - \Delta(\mathbf{u}_2, \mathbf{v}_2) = n \cdot \varepsilon^{-2p}(\varepsilon^{-2} - 1) - \Delta(\mathbf{u}_2, \mathbf{v}_2)$. In the setting of Theorem A.3 of [28], if $\Delta = \Delta(\mathbf{u}_2, \mathbf{v}_2)$, then the entropy H satisfies

$$\Delta = \frac{HR}{2B} + C - C \log R - \frac{A}{2B} \log R,$$

where

$$\begin{aligned} A &= n\varepsilon^{-2p}, \\ B &= n\varepsilon^{-2p+2}(\varepsilon^{-2} - 1), \\ C &= \varepsilon^{-2}, \\ R &= A + 2BC. \end{aligned}$$

Notice that A, B, C, and R are known to Bob. Thus, to decide whether Δ is small or large, it suffices to have a $\frac{1}{\varepsilon}$-additive approximation to $HR/(2B)$, or since $B/R = \Theta(\varepsilon^2)$, it suffices to have an additive $\Theta(\varepsilon)$-approximation to H with probability at least $1 - \delta$. The theorem follows by applying Corollary 3.1.

THEOREM 4.6. *Any 1-pass streaming algorithm that outputs an additive ε-approximation to the entropy in the strict turnstile model with probability at least $1 - \delta$ must use $\Omega(\varepsilon^{-2} \log n \log 1/\delta / \log(1/\varepsilon))$ bits of space.*

REMARK 4.4. *This improves the previous $\Omega(\varepsilon^{-2} \log n / \log 1/\varepsilon)$ lower bound of [28].*

4.6 VC-Dimension and Sub-constant Error Recall that the VC-dimension $VC(f)$ of the communication matrix for a binary function f is the maximum number ℓ of columns for which all 2^ℓ possible bit patterns occur in the rows of the matrix restricted to those columns. In [30], Kremer, Nisan, and Ron show the surprising result that the VC-dimension $VC(f)$ of the communication matrix for f exactly characterizes $R^{\rightarrow,\parallel}_{1/3}(f)$, namely,

THEOREM 4.7. *([30]) $R^{\rightarrow,\parallel}_{1/3}(f) = \Theta(VC(f))$.*

We show that for sub-constant error probabilities δ, the VC-dimension does not capture $R^{\rightarrow,\parallel}_{\delta}(f)$.

THEOREM 4.8. *There exist problems f, g for which $VC(f) = VC(g) = N$, yet*

- $R^{\rightarrow,\parallel}_{\delta}(f) = \Theta(N)$.
- $R^{\rightarrow,\parallel}_{\delta}(g) = \Theta(N \log 1/\delta)$.

Proof. For f, we take the Indexing function. Namely, Alice is given $x \in \{0,1\}^N$, Bob is given $i \in [N]$, and $f(x, i) = x_i$. It is easy to see that $VC(f) = N$, and it is well-known [30, 31] that $R^{\rightarrow,\parallel}_{\delta}(f) = \Theta(N)$ in this case, if, say, $\delta < 1/3$.

For g, we take the problem IND with $|\mathcal{U}| = \frac{1}{8\delta}$. By Remark 3.2 following Corollary 3.1 (and a trivial upper bound), $R^{\rightarrow,\parallel}_{\delta}(\text{IND}) = \Theta(N \log 1/\delta)$. On the other hand, $VC(g) \leq N$ since for each row of the communication matrix, there are at most N ones. Also, $VC(g) = N$ since the matrix for f occurs as a submatrix for g. This completes the proof.

The separation in Theorem 4.8 is best possible, since the success probability can always be amplified to $1 - \delta$ with $O(\log 1/\delta)$ independent repetitions.

References

[1] Dimitris Achlioptas. Database-friendly random projections: Johnson-lindenstrauss with binary coins. *J. Comput. Syst. Sci.*, 66(4):671–687, 2003.

[2] Nir Ailon and Bernard Chazelle. The fast johnson–lindenstrauss transform and approximate nearest neighbors. *SIAM J. Comput.*, 39(1):302–322, 2009.

[3] Nir Ailon and Bernard Chazelle. Faster dimension reduction. *Commun. ACM*, 53(2):97–104, 2010.

[4] Nir Ailon and Edo Liberty. Fast dimension reduction using rademacher series on dual bch codes. *Discrete & Computational Geometry*, 42(4):615–630, 2009.

[5] Nir Ailon and Edo Liberty. Almost optimal unrestricted fast johnson-lindenstrauss transform. *CoRR*, abs/1005.5513, 2010.

[6] Noga Alon. Problems and results in extremal combinatorics–i. *Discrete Mathematics*, 273(1-3):31–53, 2003.

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

[8] Rosa I. Arriaga and Santosh Vempala. An algorithmic theory of learning: Robust concepts and random projection. In *FOCS*, pages 616–623, 1999.

[9] Khanh Do Ba, Piotr Indyk, Eric C. Price, and David P. Woodruff. Lower bounds for sparse recovery. In *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), to appear*, 2010.

[10] Ziv Bar-Yossef, T. S. Jayram, Robert Krauthgamer, and Ravi Kumar. The sketching complexity of pattern matching. In *8th International Workshop on Randomization and Computation (RANDOM)*, pages 261–272, 2004.

[11] Ziv Bar-Yossef, T. S. Jayram, Ravi Kumar, and D. Sivakumar. Information theory methods in communication complexity. In *Proceedings of the 17th Annual IEEE Conference on Computational Complexity (CCC)*, pages 93–102, 2002.

[12] Ziv Bar-Yossef, T. S. Jayram, Ravi Kumar, D. Sivakumar, and Luca Trevisan. Counting distinct elements in a data stream. In *Randomization and Approximation Techniques, 6th International Workshop (RANDOM)*, pages 1–10, 2002.

[13] Boaz Barak, Mark Braverman, Xi Chen, and Anup Rao. How to compress interactive communication. In *STOC*, pages 67–76, 2010.

[14] Emmanuel J. Candès and Terence Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Transactions on Information Theory*, 52(12):5406–5425, 2006.

[15] Amit Chakrabarti, Yaoyun Shi, Anthony Wirth, and Andrew Chi-Chih Yao. Informational complexity and the direct sum problem for simultaneous message complexity. In *FOCS*, pages 270–278, 2001.

[16] Moses Charikar, Kevin Chen, and Martin Farach-Colton. Finding frequent items in data streams. In *ICALP*, pages 693–703, 2002.

[17] Kenneth L. Clarkson. Tighter bounds for random projections of manifolds. In *Symposium on Computational Geometry*, pages 39–48, 2008.

[18] Kenneth L. Clarkson and David P. Woodruff. Numerical linear algebra in the streaming model. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC)*, pages 205–214, 2009.

[19] Thomas Cover and Joy Thomas. *Elements of Information Theory*. Wiley Interscience, 1991.

[20] Anirban Dasgupta, Ravi Kumar, and Tamás Sarlós. A sparse johnson: Lindenstrauss transform. In *STOC*, pages 341–350, 2010.

[21] Sanjoy Dasgupta and Anupam Gupta. An elementary proof of a theorem of johnson and lindenstrauss. *Random Struct. Algorithms*, 22(1):60–65, 2003.

[22] Petros Drineas, Michael W. Mahoney, S. Muthukrishnan, and Tamás Sarlós. Faster least squares approximation. *CoRR*, abs/0710.1435, 2007.

[23] Prahladh Harsha, Rahul Jain, David A. McAllester, and Jaikumar Radhakrishnan. The communication complexity of correlation. In *IEEE Conference on Computational Complexity*, pages 10–23, 2007.

[24] Piotr Indyk. Stable distributions, pseudorandom generators, embeddings, and data stream computation. *J. ACM*, 53(3):307–323, 2006.

[25] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *STOC*, pages 604–613, 1998.

[26] Piotr Indyk and David P. Woodruff. Tight lower bounds for the distinct elements problem. In *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 283–, 2003.

[27] Rahul Jain, Pranab Sen, and Jaikumar Radhakrishnan. Optimal direct sum and privacy trade-off results for quantum and classical communication complexity. *CoRR*, abs/0807.1267, 2008.

[28] Daniel M. Kane, Jelani Nelson, and David P. Woodruff. On the exact space complexity of sketching and streaming small norms. In *SODA*, 2010.

[29] Daniel M. Kane, Jelani Nelson, and David P. Woodruff. An optimal algorithm for the distinct elements problem. In *PODS*, pages 41–52, 2010.

[30] Ilan Kremer, Noam Nisan, and Dana Ron. On randomized one-round communication complexity. *Computational Complexity*, 8(1):21–49, 1999.

[31] Eyal Kushilevitz and Noam Nisan. *Communication Complexity*. Cambridge University Press, 1997.

[32] J. Langford, L. Li, and A. Strehl. Vowpal wabbit online learning. Technical report, 2007.

[33] Edo Liberty, Nir Ailon, and Amit Singer. Dense fast random projections and lean walsh transforms. In *APPROX-RANDOM*, pages 512–522, 2008.

[34] Jirí Matousek. On variants of the johnson-lindenstrauss lemma. *Random Struct. Algorithms*, 33(2):142–156, 2008.

[35] Peter Bro Miltersen, Noam Nisan, Shmuel Safra, and Avi Wigderson. On data structures and asymmetric communication complexity. *J. Comput. Syst. Sci.*, 57(1):37–49, 1998.

[36] S. Muthukrishnan. Data Streams: Algorithms and Applications. *Foundations and Trends in Theoretical Computer Science*, 1(2):117–236, 2005.

[37] V. Rokhlin, A. Szlam, and M. Tygert. A randomized algorithm for principal component analysis. *SIAM Journal on Matrix Analysis and Applications*, 31(3):1100, 1124, 2009.

[38] Tamás Sarlós. Improved approximation algorithms for large matrices via random projections. In *FOCS*, pages 143–152, 2006.

[39] Q. Shi, J. Petterson, G. Dror, J. Langford, A. J. Smola, A. Strehl, and V. Vishwanathan. Hash kernels. In *AISTATS 12*, 2009.

[40] Daniel A. Spielman and Nikhil Srivastava. Graph sparsification by effective resistances. In *STOC*, pages

563–568, 2008.

[41] Divesh Srivastava and Suresh Venkatasubramanian. Information theory for data management. In *SIGMOD '10: Proceedings of the 2010 international conference on Management of data*, pages 1255–1256, New York, NY, USA, 2010. ACM.

[42] Mikkel Thorup and Yin Zhang. Tabulation based 4-universal hashing with applications to second moment estimation. In *Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 615–624, 2004.

[43] Kilian Q. Weinberger, Anirban Dasgupta, Josh Attenberg, John Langford, and Alex J. Smola. Feature hashing for large scale multitask learning. *CoRR*, abs/0902.2206, 2009.

[44] David P. Woodruff. Optimal space lower bounds for all frequency moments. In *Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 167–175, 2004.

A Proof of Lemma 3.1

We first define and analyze a basic encoding scheme. Let $\mathbf{w} \in \mathcal{U}^m$. Let $s : \mathcal{U}^m \to \{-1,+1\}^m$ be a random hash function. We define $\mathrm{enc}_1(\mathbf{w}, s)$ to be the majority of the ± 1 values in the m components of $s(\mathbf{w})$. This is well-defined since m is odd. We contrast this with another encoding defined with an additional parameter $j \in [m]$. Define $\mathrm{enc}_2(\mathbf{w}, j, s)$ to be just the j-th component of $s(\mathbf{w})$.

To analyze this scheme, fix two vectors $\mathbf{w}, \mathbf{z} \in \mathcal{U}^m$ and an index j. If $w_j \neq z_j$, then

$$\Pr[\mathrm{enc}_1(\mathbf{w}, s) \neq \mathrm{enc}_2(\mathbf{z}, j, s)] = \tfrac{1}{2}.$$

On the other hand, suppose $w_j = z_j$. Then, by a standard argument involving the binomial coefficients,

$$\Pr[\mathrm{enc}_1(\mathbf{w}, s) \neq \mathrm{enc}_2(\mathbf{z}, j, s)] \leq \tfrac{1}{2}(1 - \tfrac{1}{2\sqrt{m}}) = \tfrac{1}{2} - \varepsilon.$$

We repeat the above scheme to amplify the gap between the two cases. Let $\mathbf{s} = (s_1, s_2, \ldots, s_k)$ be a collection of $k = \frac{10}{\varepsilon^2} \cdot \log 1/\delta$ i.i.d. random hash functions each mapping $\mathcal{U}^m$ to $\{-1,+1\}^a$. Define

$$\mathrm{enc}_1(\mathbf{w}, \mathbf{s}) = (\mathrm{enc}_1(\mathbf{w}, s_1), \mathrm{enc}_1(\mathbf{w}, s_2), \ldots, \mathrm{enc}_1(\mathbf{w}, s_k)),$$

and

$$\mathrm{enc}_2(\mathbf{z}, j, \mathbf{s}) = (\mathrm{enc}_1(\mathbf{z}, j, s_1), \ldots, \mathrm{enc}_2(\mathbf{z}, j, s_k)),$$

For ease of notation, let $\mathbf{w}' = \mathrm{enc}_1(\mathbf{u}, \mathbf{s})$ and $\mathbf{z}' = \mathrm{enc}_2(\mathbf{z}, j, \mathbf{s})$.

FACT A.1. *Let $X_1, X_2, \ldots, X_k$ be a collection of i.i.d. 0-1 Bernoulli random variables with success probability p. Set $\bar{X} = \sum_i X_i/k$. Then,*

$$\Pr[\bar{X} < p - h] < exp(-2h^2k), \text{ and}$$
$$\Pr[\bar{X} > p + h] < exp(-2h^2k).$$

In the above fact, with $k = 10\varepsilon^{-2} \log 1/\delta$ and $h = \varepsilon/3$, we obtain that the tail probability is at most δ. In the case $w_j \neq z_j$ we have $p = \frac{1}{2}$, so

$$\Pr[\Delta(\mathbf{w}', \mathbf{z}') < k(\tfrac{1}{2} - \tfrac{\varepsilon}{3})] \leq \delta. \tag{A.1}$$

In the second case, $p = \frac{1}{2} - \varepsilon$,

$$\Pr[\Delta(\mathbf{w}', \mathbf{z}') > k(\tfrac{1}{2} - \tfrac{2\varepsilon}{3})] \leq \delta. \tag{A.2}$$

The two cases differ by a factor of at least $1 + \varepsilon/3$ for ε less than a small enough constant.

Divide $[N]$ into b blocks where the q-th block equals $[(q-1)m + 1, qm]$ for every $q \in [b]$. We use the above to define an encoding for promise inputs $(\mathbf{x}, \mathbf{y})$ to the problem $\mathrm{IND}^{\mathsf{a}}_{\mathcal{U}}$, where the goal is to decide for an index i belonging to block p whether $x_i = y_i$. Let $j = i - (p-1)m$ denote the offset of i within block p. We also think of $\mathbf{x}$ and $\mathbf{y}$ as being analogously divided into b blocks $\mathbf{x}_{[1]}, \mathbf{x}_{[2]}, \ldots, \mathbf{x}_{[b]}$ and $\mathbf{y}_{[1]}, \mathbf{y}_{[2]}, \ldots, \mathbf{y}_{[b]}$ respectively. Thus, the goal is to decide whether the j-th components of $\mathbf{x}_{[p]}$ and $\mathbf{y}_{[p]}$ are equal.

Fix a block index q. Let $\mathbf{s}_{[q]}$ denote a vector of k i.i.d. random hash functions corresponding to block q. Compute $\mathrm{enc}_1(\mathbf{x}_{[q]}, \mathbf{s}_{[q]})$ and then repeat each coordinate of this vector α^{b-q} times. Call the resulting vector $\mathbf{x}'_{[q]}$. For $\mathbf{y}_{[q]}$, the encoding $\mathbf{y}'_{[q]}$ depends on the relationship of q to p and and additionally on j (both p and j are determined by $\mathbf{y}$). If $q < p$, we use the same encoding function as that for $\mathbf{x}_{[q]}$, i.e. $\mathrm{enc}_1(\mathbf{y}_{[q]}, \mathbf{s}_{[q]})$ repeated α^{b-q} times. If $q > p$, the encoding is a 0 vector of length $\alpha^{b-q} \cdot k$. If $q = p$, the encoding equals $\mathrm{enc}_2(\mathbf{y}_{[p]}, j, \mathbf{s}_{[q]})$ using the second encoding function, again repeated α^{b-p} times. For each q, the lengths of both $\mathbf{x}'_{[q]}$ and $\mathbf{y}'_{[q]}$ equal $\alpha^{b-q} \cdot k$. Finally, define a dummy vector $\mathbf{x}_{[b+1]}$ of length $k/(\alpha-1)$ all of whose components equal 1, and another dummy vector $\mathbf{y}_{[b+1]}$ of the same length all of whose components equal 0. We define the encoding $\mathbf{x} \rightsquigarrow \mathbf{u}$ to be the concatenation of all $\mathbf{x}'_{[q]}$ for all $1 \leq q \leq b+1$. Similarly for $\mathbf{y} \rightsquigarrow \mathbf{v}$. The encodings have length

$$\begin{aligned} n &= k/(\alpha-1) + \sum_{1 \leq q \leq b} \alpha^{b-q} \cdot k \\ &= \alpha^b \cdot k/(\alpha-1) \\ &= O(\alpha^b \cdot \tfrac{1}{\varepsilon^2} \cdot \log 1/\delta). \end{aligned}$$

Moreover, the values are in $\{-1, 0, +1\}$ but a simple fix to be described at the end will transform this into a 0-1 vector.

We now define the split $\mathbf{u} = (\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3)$ and $\mathbf{v} = (\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$. Define $\mathbf{u}_1$ (respectively $\mathbf{u}_2$, $\mathbf{u}_3$) to be the

concatenation of all $\mathbf{x}'_{[q]}$ for $q < p$ (respectively $q = p$, $q > p$). Define $\mathbf{v}_c$ for $c = 1, 2, 3$ analogously.

First, note that $\mathbf{u}_1 = \mathbf{v}_1$ because $\mathbf{x}'_{[q]} = \mathbf{y}'_{[q]}$ for $q < p$. Next, the lengths of $\mathbf{u}_3$ and $\mathbf{v}_3$ equal

$$\begin{aligned} k/(\alpha-1) + \sum_{p+1 \le q \le b} \alpha^{b-q} \cdot k &= \alpha^{b-p} \cdot k/(\alpha-1) \\ &= n \cdot \alpha^{-p} = n_3. \end{aligned}$$

Since $\mathbf{u}_3$ is a ± 1 vector while $\mathbf{v}_3$ is a 0 vector, $\|\mathbf{u}_3 - \mathbf{v}_3\|_1 = n_3$. Last, we look at $\mathbf{u}_2$ and $\mathbf{v}_2$. Their lengths equal $\alpha^{b-p} \cdot k = n \cdot \alpha^{-p}(\alpha - 1) = n_2$. We now analyze $\|\mathbf{u}_2 - \mathbf{v}_2\|_1 = 2\Delta(\mathbf{x}'_{[p]}, \mathbf{y}'_{[p]})$. We distinguish between the YES and NO instances via (A.1) and (A.2). For an NO instance, $x_i \neq y_i$, so by (A.1), with probability at least $1 - \delta$,

$$\|\mathbf{u}_2 - \mathbf{v}_2\|_1 \geq 2n_2(\tfrac{1}{2} - \tfrac{\varepsilon}{3}).$$

For a YES instance, a similar calculation using (A.2) shows that with probability at least $1 - \delta$,

$$\|\mathbf{u}_2 - \mathbf{v}_2\|_1 \leq 2n_2(\tfrac{1}{2} - \tfrac{2\varepsilon}{3}).$$

To obtain the required 0-1 vectors, apply a simple transformation of $\{-1 \to 0101, 0 \to 0011, +1 \to 1010\}$ to $\mathbf{u}$ and $\mathbf{v}$. This produces 0-1 inputs having a relative Hamming weight of exactly half in each of the $\mathbf{u}_i$'s and $\mathbf{v}_i$'s. The length quadruples while a norm distance of d translates to a Hamming distance of $2d$, which translates to the bounds stated in the lemma.