Stable Distributions, Pseudorandom Generators, Embeddings and Data Stream Computation

Piotr Indyk *

Abstract

In this paper we show several results obtained by combining the use of stable distributions *with* pseudorandom generators for bounded space*. In particular:*

- *we show how to maintain (using only $O(\log n/\epsilon^2)$ words of storage) a* sketch *$C(p)$ of a point $p \in l_1^n$ under dynamic updates of its coordinates, such that given sketches $C(p)$ and $C(q)$ one can estimate $|p-q|_1$ up to a factor of $(1+\epsilon)$ with large probability. This solves the main open problem of [10].*
- *we obtain another sketch function C' which maps l_1^n into a* normed space *l_1^m (as opposed to C), such that $m = m(n)$ is much smaller than n; to our knowledge this is the first dimensionality reduction lemma for l_1 norm*
- *we give an explicit embedding of l_2^n into $l_1^{n^{O(\log n)}}$ with distortion $(1 + 1/n^{\Theta(1)})$ and a non-constructive embedding of l_2^n into $l_1^{O(n)}$ with distortion $(1+\epsilon)$ such that the embedding can be represented using only $O(n \log^2 n)$ bits (as opposed to at least n^2 used by earlier methods)*

1 Introduction

Stable distributions [26] are defined as limits of normalized sums of independent identically distributed variables (see also Preliminaries for an alternative definition). The most well-known example of a stable distribution is Gaussian (or normal) distribution. However, the class is much wider; for example, it includes heavy-tailed distributions. Stable distribution have found numerous applications in various fields (see the survey [24] for more details).

In this paper we show that the combination of stable distributions and *bounded space pseudorandom generators* [23] forms a powerful tool for proving a variety of embedding-like results. The basic idea behind this combination is as follows. It is known [11, 25, 20] that an inner product of a vector $u \in l_p^d$ with a sequence of n i.i.d. random variables having stable distribution (with parameter p, see Preliminaries) is a good estimator of l_p norm of u; in particular, one can use several such products to embed l_p into some other space. Since inner product can be computed in a small space, we can use pseudorandom generators to reduce the number of required random bits. This in turn translates into reduction of storage/dimensionality/number of non-uniform bits or other parameters of interest, depending on the application.

In the following we describe in more detail applications of this technique to computing with data streams, dimensionality reduction in l_1 and embeddings of l_2 into l_1; we also describe the relevant algorithmic implications.

Stream computation. The first problem we address is defined as follows [10] (see also [14] for a background on stream computation). Assume that we have an access to a stream S of data, where each chunk of data is of the form (i, a), where $i \in [n] = \{0 \dots n-1\}$ and $a \in \{-M \dots M\}$. We want to approximate (up to the multiplicative factor $(1 \pm \epsilon)$) the quantity $L_1(S)$, where

$$L_p(S) = (\sum_{i \in [n]} | \sum_{(i,a) \in S} a|^p)^{1/p}.$$

The problem has a variety applications to estimating the size of self-join [1, 13] and potential applications to estimation of statistics of Net-Flow data [10]. An obvious solution to this problem is to maintain a counter c_i for each i and compute the sum of $|c_i|$'s at the end. Unfortunately, this solution requires $\Theta(n)$ words of storage. In their

*Stanford University. E-mail: `indyk@cs.stanford.edu`
Part of this work was done while the author was visiting AT&T Shannon Labs.

influential paper, Alon, Matias and Szegedy [1] proposed a very nice and simple scheme for approximating $L_2(S)$[1] in space $O(1/\epsilon^2)$ with (arbitrarily large) constant probability. Feigenbaum, Kannan, Strauss and Viswanathan [10] proposed an algorithm (using similar amount of memory) for $L_1(S)$ for the case where (roughly) for each i the stream S contains at most two pairs (i, a). An alternative way to view their result is to assume two streams, one (S_r) containing red pairs and another one (S_b) containing blue pairs; for each i there is at most one pair (i, a) of each color. The goal is to compute *sketches* $C(S_r)$ and $C(S_b)$ of small size, such that the approximate value $L_1(S_r, S_b) = \sum_i |\sum_{(i,a)\in S_r} a - \sum_{(i,a)\in S_b} a|$ can be quickly evaluated from $C(S_r)$ and $C(S_b)$ by applying some function F (see [10] for more details of the model). Computing sketches of normed vectors enables to compress the data and speed-up computation, e.g., see [18] where this was approach was shown to give up to an order of magnitude speed-up for various data-mining problems; see also [4, 3, 6] (where a somewhat different similarity measure has been used).

In this paper we propose a unified framework for approximating $L_p(S)$ for $p \in \{1, 2\}$[2], using $O(\log n/\epsilon^2)$ memory words. Our algorithm does not have the aforementioned restrictions of [10]; thus, it solves the main open problem from that paper (see [10], comments after Corollary 16). Moreover, our algorithm maintains only linear combinations of the input values, and therefore extends also to the sketch model (again, without the restrictions of [10]). Since the algorithm is simple and free of large constants, it can be used to extend the methods of [18] to l_1 norm and it is also likely to find practical uses for the compression applications mentioned in [10].

Dimensionality reduction. The above stream algorithms, especially those operating in the sketch model, can be viewed as dimensionality reduction techniques. Indeed, the streams S_b and S_r can be viewed as points in n-dimensional space and $L_p(S_r, S_b)$ is just a norm (for $p \geq 1$). Thus the sketch operator C can be viewed as an approximate embedding of l_p^n into the sketch space (say $\mathcal{C}$), such that

- each point from $\mathcal{C}$ can be described using only small number (say m) of numbers (so we can assume $\mathcal{C} \subset \Re^m$).
- the value of $L_p(S_r, S_b)$ is approximately equal to $F(C(S_r), C(S_b))$

However, all of the above algorithms have the unfortunate property that the pair $(\mathcal{C}, F)$ is not a *normed space.* Specifically, the definition of F involves the median operator[3]; e.g. for L_1

$$F((x_1, \ldots, x_m), (y_1, \ldots, y_m)) = \text{median}(|x_1 - y_1|, \ldots, |x_m - y_m|)$$

Since F is not a norm, none of the large number of algorithms designed for normed spaces can be used. Thus, if one would like to perform any non-trivial operation on the set of points in the sketch space (e.g. clustering, similarity search, regression etc), not being able to apply algorithms designed for normed spaces is a serious disadvantage.

In this paper we attempt to overcome this difficulty. For L_2, one can observe that we can replace median by sum in our algorithm without significantly increasing the probability of error (this follows from the proof of Johnson-Lindenstrauss dimensionality reduction lemma as in [16]). For L_1, the situation is more complicated, since for sketch points $(x_1, \ldots, x_m), (y_1, \ldots, y_m)$ the expectation $E[|x_i - y_i|]$ is undefined (i.e. is equal to ∞). However, we were able to show that there exists a sketch function C which maps the points into $m = (\ln(1/\delta)/\epsilon)^{O(1/\epsilon)}$-dimensional space with l_1 norm, such that for any pair of points p, q:

- $|C(p) - C(q)|_1 \geq (1 - \epsilon)|p - q|_1$ with probability at least $1 - \delta$ (i.e. C is almost non-contractive with high probability)
- $|C(p) - C(q)|_1 \leq (1 + \epsilon)|p - q|_1$ with probability at least $1 - 1/(1 + \epsilon)$ (i.e. is almost non-expansive with a constant probability)

Note, that this can be viewed as a one-sided analog of Johnson-Lindenstrauss dimensionality reduction for l_1 (to our knowledge this is the first dimensionality reduction theorem for l_1). Although we cannot ensure that the mapping does not *expand* a fixed pair of points with high probability, the one-sided guarantee is good enough for several purposes. In particular, consider searching for the nearest neighbor (say of point q): if the distance from q to its nearest neighbor p does not expand much, and the distance to any other point p' does not contract much, we are still guaranteed to return

[1] In their original paper they assumed all pairs are of the form $(i, +1)$, but it was shown in [10] that their algorithm actually works for the general case.

[2] We also discuss the extension to any $p \in (0, 2]$.

[3] For L_2 the algorithms in [1, 10] *can* be implemented with median replaced by a sum; unfortunately, in that case the sketch size depends *polynomially*, not *logarithmically* on the probability of error. This makes the modified algorithm unsuitable for the applications mentioned below.

an approximate nearest neighbor of q (note that we can ensure this happens with constant probability, which can be amplified by using multiple data structures). By reductions in [8, 9, 16, 17, 5, 12] solving approximate nearest neighbor gives us efficient algorithms for hierarchical clustering, Minimum Spanning Tree clustering, diameter and other forms of clustering. Thus our dimensionality reduction technique is sufficient for a large class of algorithmic problems.

Deterministic embeddings of l_2 into l_1. It is known (e.g. see [11] and references therein) that l_2^n can be embedded into $l_1^{O(n)}$ with distortion $(1+\epsilon)$ (the $O()$ constant depends on the ϵ). Unfortunately, none of those proofs is constructive. To our knowledge, the only constructive result of this type [2, 21] embeds l_2^n into $l_1^{O(n^2)}$ with $\sqrt{3}$ distortion. In this paper we provide:

- an explicit embedding of l_2^n into $l_1^{n^{O(\log n)}}$ with distortion $(1+1/n^{\Theta(1)})$
- a non-constructive embedding of l_2^n into $l_1^{O(n)}$ with distortion $(1+\epsilon)$ such that the embedding can be represented using only $O(n\log^2 n)$ bits (as opposed to at least n^2 used by earlier methods); this reduces the non-uniformity and space requirements of the embedding

By combining the first result with the result of [15] we obtain a $(3+\epsilon)$-approximate *deterministic* algorithm for the nearest neighbor search in l_2^n with polynomial preprocessing/storage and $\tilde{O}(n^{\log n})$ query time. Note that for not-so-large dimension n (e.g. polylogarithmic in the data set size) this gives a sublinear query time.

2 Preliminaries

Stable distributions. A distribution $\mathcal{D}$ over $\Re$ is called *p-stable*, if there exists $p \geq 0$ such that for any n real numbers $a_1 \ldots a_n$ and i.i.d. variables $X_1 \ldots X_n$ variables with distribution $\mathcal{D}$, the random variable $\sum_i a_i X_i$ has the same distribution as the variable $(\sum_i |a_i|^p)^{1/p} X$, where X is a random variable with distribution $\mathcal{D}$.

It is known [26] that stable distributions exist for any $p \in (0,2]$. In particular:

- a *Cauchy distribution* $\mathcal{D}_C$, defined by the density function $c(x) = \frac{1}{\pi}\frac{1}{1+x^2}$, is 1-stable
- a *Gaussian (normal) distribution* $\mathcal{D}_G$, defined by the density function $g(x) = \frac{1}{\sqrt{2\pi}}e^{-x^2/2}$, is 2-stable

Pseudorandom generators (PRGs). As in [23] we consider PRGs which fool any Finite State Machine (FSM) which uses at most $O(S)$ bits of space (or $2^{O(S)}$ states). Assume that a FSM $Q \in \text{space}(S)$ uses at most k chunks of random bits, where each chunk is of length b. The generator $G: \{0,1\}^m \to (\{0,1\}^b)^k$ expands a "small number" m of "truly random" bits into kb bits which "look random" for Q. Formally, it is defined as follows. Let $\mathcal{D}^t$ be a uniform distribution over $\{0,1\}^t$. For any (discrete) random variable X let $\mathcal{D}[X]$ be the distribution of X, interpreted as a vector of probabilities. Let $Q(x)$ denote the state of Q after using the random bits sequence x. Then we say that G is a PRG *with parameter* $\epsilon > 0$ for a class $\mathcal{C}$ of FSMs, if for every $Q \in \mathcal{C}$

$$|\mathcal{D}[Q_{x\in\mathcal{D}^{bk}}(x)] - \mathcal{D}[Q_{x\in\mathcal{D}^m}(G(x))]|_1 \leq \epsilon$$

where $|y|_1$ denotes an l_1 norm of a vector y.

Fact 1 ([23]) *There exists a PRG G for space(S) with parameter $\epsilon = 2^{-O(S)}$ such that:*

- *G expands $O(S\log R)$ bits into $O(R)$ bits*
- *G requires only $O(S)$ bits of storage (in addition to its random input)*
- *any length-$O(S)$ chunk of $G(x)$ can be computed using $O(\log R)$ arithmetic operations on $O(S)$-bit words*

Other assumptions and notation. To simplify expressions we assume that $M \geq n$. Also, we will assume that the processor can operate on $\log M$-bit words in unit cost. One can easily modify our upper bounds for the case when either of these assumptions is not true.

All $O()$ constants in the paper are absolute, except when it is clearly stated (in which case we use $O_t()$ to denote dependence on t).

3 Approximation of l_p difference for data streams

Let S be the data stream sequence containing pairs (i,a), for $i \in [n]$ and $a \in \{-M \ldots M\}$. We present the algorithm for calculating $L_1(S)$; the extension to $p \neq 1$ is discussed at the end.

We present our algorithm in three step. In the first step we present an algorithm which approximates well $L_1(S)$, but suffers from two major drawbacks:

1. It assumes infinite precision of the calculations (i.e. uses arithmetic operations on real numbers)

2. Although it uses only $O(1/\epsilon^2)$ words for storage, it performs random (and multiple) access to as many as $\Theta(n)$ random numbers. Thus a natural implementation of the algorithm would require $\Theta(n)$ storage.

Despite these limitations, the algorithm will serve well as an illustration of our main ideas. In the next two steps, we will remove its limitations.

An ideal algorithm. Let $l = c/\epsilon^2 \log 1/\delta$ for a constant $c > 1$ specified later. The algorithm works as follows.

1. Initialize nl independent random variables $X_i^j, i \in [n], j \in [l]$ with Cauchy distribution; set $S^j = 0$, for $j \in [l]$
2. For each new pair (i, a): perform $S^j = S^j + aX_i^j$ for all $j \in [l]$
3. Return median$(|S^0|, \dots |S^{l-1}|)$

Let $c_i = \sum_{(i,a)\in S} a$; if there is no $(i, a) \in S$, we define $c_i = 0$. Thus $L_1(S) = C = \sum_i |c_i|$. The following claim justifies the correctness of the algorithm.

Claim 1 *Each S^j has the same distribution as CX where X has Cauchy distribution.*

Proof: Follows from the 1-stability of Cauchy distribution. □

Therefore, it is sufficient to estimate C from independent samples of CX, i.e. from $S^0 \dots S^{l-1}$. To this end, we use the following Lemmas.

Lemma 1 *If X has Cauchy distribution, then median$(|X|) = 1$. Therefore, median$(a|X|) = a$, for any $a > 0$.*

Proof: If X has Cauchy distribution, then the density function of $|X|$ is $f(x) = \frac{2}{\pi}\frac{1}{1+x^2}$. Therefore, the distribution function of X is equal to

$$F(z) = \int_0^z f(x)dx = \frac{2}{\pi}\arctan(z)$$

Since $\tan(\pi/4) = 1$, we have $F(1) = 1/2$. Thus median$(X) = 1$. □

Lemma 2 *For any distribution D on $\Re$ with the distribution function F, take $l = c/\epsilon^2 \log 1/\delta$ independent samples $X_0 \dots X_{l-1}$ of $\mathcal{D}$; also, let $X =$ median$(X_0 \dots X_{l-1})$. Then for a suitable constant c we have*

$$\Pr[F(X) \in [1/2 - \epsilon, 1/2 + \epsilon]] > 1 - \delta$$

Proof: Folklore. □

Lemma 3 *Let F be the distribution function of $|X|$ where X has Cauchy distribution, and let $z > 0$ be such that $F(z) \in [1/2 - \epsilon, 1/2 + \epsilon]$. Then, if ϵ is small enough, we have $z \in [1 - 4\epsilon, 1 + 4\epsilon]$.*

Proof: Follows from the fact that $F^{-1}(x) = \tan(x\pi/2)$ has bounded derivative around the point $1/2$. In particular, $(F^{-1})'(1/2) = \pi$. □

Therefore, for a suitable constant c, we have the following Theorem.

Theorem 1 *The "ideal" algorithm correctly estimates $L_1(S)$ up to the factor $(1 \pm \epsilon)$ with probability at least $1 - \delta$.*

Bounded precision. Now we show how to remove the assumption that the numbers on which we perform operations have infinite precision. Since the number is in the data stream are integers, we only need to take care of the random variables X_i^j. Specifically, we show that it is sufficient to assume that they take values in the set $V_L = \{p/q : p, q \in \{-L, L\}, q \neq 0\}$, where L is small.

Consider the following way of generating X_i^j. Let Y_i^j be a random number from the set $[0, 1)$. We define $X_i^j = F^{-1}(Y_i^j) = \tan(\pi Y_i^j/2)$. Now we define an approximation $\tilde{X}_i^j$ of X_i^j. Let $\tilde{Y}_i^j$ be equal to Y_i^j rounded to the nearest multiple of $1/L$. We define $\tilde{X}_i^j$ to be $F^{-1}(\tilde{Y}_i^j)$, again rounded to the nearest multiple of $1/L$. Consider the case when $\tilde{Y}_i^j < 1 - K/L = 1 - \alpha$ (K to be specified later). Since the derivative of $F^{-1}(x)$ for $x < 1 - \alpha$ is $O(1/\alpha^2)$, it follows that in this case $\tilde{X}_i^j = X_i^j + E_i^j$, where $|E_i^j| = O(\frac{1}{\alpha^2 L}) = O(K^2/L) = \beta$.

Now we set K and L such that $K/L < 1/(n/\delta)^{\Theta(1)}$ and $\beta << \epsilon$, in which case we know that $\tilde{X}_i^j = X_i^j \pm \beta$ for all i, j with high probability. Then the value

$$\tilde{S}^j = \sum_i \sum_{(i,a)\in S} a\tilde{X}^j = \sum_i c_i \tilde{X}_i^j = \sum_i c_i(X_i^j \pm \beta) = S^j \pm \beta \sum_i c_i$$

Since median$(S^j) = \sum_i |c_i|$, by making β to be sufficiently smaller than ϵ, we can ignore the contribution of $\beta \sum_i c_i$ to the estimated quantity.

Randomness reduction. Consider a fixed S^j. From the above it follows that the value of S^j can be represented using $O(\log M)$ bits; also, we need only $O(\log n)$ bits to generate each $\tilde{X}_i^j$. Unfortunately, we still need $O(n)$ memory words to make sure that if we access a specific $\tilde{X}_i^j$ several times, its value is always the same. We avoid this problem in the following way. Assume for a moment that the pairs (i, a) are coming in the increasing order of i. In this case we do not have to store X_j^i, since we can generate them on the fly. Thus, the algorithm uses only $O(\log M)$ storage and $O(n)$ chunks of randomness, and thus there exists a PRG G which given a random seed of size $O(\log M \log(n/\delta))$ expands it to a sequence $\overline{X}_0^j \dots \overline{X}_{n-1}^j$, such that using $\overline{X}_i^j$ instead of $\tilde{X}_i^j$ results in negligible probability of error and therefore the resulting value of $\tilde{S}_i^j$ (call it $\overline{S}_i^j$) can be used to estimate $L_1(S)$. However, observe that for a fixed random seed r, the value S^j *does not depend* on the order in which the pairs (i, a) come (since addition is commutative). Therefore, G is good as well if the input is unsorted, i.e. the pairs come in arbitrary order. Since we use l random seeds for each $j \in [l]$, we obtain the following result.

Theorem 2 *There is an algorithm which estimates $L_1(S)$ up to a factor $(1 \pm \epsilon)$ with probability $1 - \delta)$ and uses*

- $O(\log M \log(1/\delta)/\epsilon^2)$ *bits of random access storage*
- $O(\log M \log(n/\delta) \log(1/\delta)/\epsilon^2)$ *random bits (which can be stored in a random access storage)*
- $O(\log(n/\delta))$ *arithmetic operations per pair (i, a)*

Computing $L_p(S)$. For $p = 2$, the algorithm and analysis remains essentially the same, with Cauchy distribution replaced by Gaussian. For general $p \in (0, 2]$ the algorithm and analysis become more involved, mainly due to the fact that no closed formulas are known for densities and/or distribution functions of general p-stable distribution. However, it is known [7] that one can generate p-stable random variables essentially from two independent variables distributed uniformly over $[0, 1]$; therefore, one can implement our algorithm for general p. As far as the analysis is concerned, it seems (we did not perform a rigorous verification of this fact) that the distribution functions of p-stable are Lipschitz around the median (i.e. an analog of Lemma 3 holds); also their median can be computed numerically for any p. Therefore, it seems likely that the algorithm is provably correct also for general p. However, since we are not aware of any application which involves p different from 1 or 2, we skip further details.

4 Dimensionality reduction for l_1

In this section we show how to obtain the sketch function C which maps the points into a normed space l_1^m. We will describe the mapping in terms of dimensionality reduction of l_1^n; the adaptation to the stream model can be done as in the previous section. Specifically, we prove the following Theorem.

Theorem 3 *For any $\epsilon, \delta > 0$, there is a probability space $\mathcal{D}$ over linear mappings $f : l_1^n \to l_1^k$, where $k = (\ln(1/\delta)/\epsilon)^{O(1/\epsilon)}$, such that for any pair of points $p, q \in l_1^n$:*

- *the probability that $|f(p) - f(q)|_1 \leq |p - q|_1$ is smaller than δ*
- *the probability that $|f(p) - f(q)|_1 \geq (1+\epsilon)|p - q|_1$ is smaller than $1 - \epsilon$*

Note that the embedding is randomized but asymmetric: the probability of small expansion is only ϵ, while the probability of small contraction is $1 - \delta$.

Proof: We define the random mapping f such that for $j = 1 \dots k$ the jth coordinate of $f((v_1, \dots, v_n))$ is equal to $\sum_i X_i^j v_i$, where X_i^j are i.i.d random variables having Cauchy distribution. Since f is linear, it sufficient to show the above for $q = 0$ and p such that $|p|_1 = 1$. In this case $|f(p) - f(q)| = \sum_j |\sum_i X_i^j v_i| = \sum_j |Y_j|$. Since the Cauchy distribution is 1-stable, each Y_j has a Cauchy distribution. Thus it is sufficient to prove the following fact: for any sequence $Y_1 \dots Y_k$ of i.i.d. variables with Cauchy distribution, let $Y = \sum_j |Y_i|$. Show that there exists a *threshold* $T = T(k, \delta, \epsilon)$, such that:

- $\Pr[Y < (1 - \epsilon)^c T] \leq \delta$, for some $c = O(1)$
- $\Pr[Y > (1 + \epsilon)T] \leq \frac{1+\epsilon/2}{1+\epsilon}$

We will first establish T which is "good" for the second condition. Let $U = a \cdot k$, for some $a \geq 1$. Since (what is easy to verify) $\Pr[|Y_i| > t] \leq b/t$ for some $b = O(1)$, by the "union bound" we have $\Pr[\exists_i |Y_i| \geq U] \leq kb/U = b/a$. We define

$$T = E[Y : \forall_i |Y_i| \leq U]$$

By the linearity of expectation

$$\begin{aligned} T &= \sum_i E[|Y_i| : |Y_i| \le U] \\ &\le \frac{k\frac{2}{\pi}\int_0^U \frac{x}{1+x^2}dx}{1-b/a} \\ &\le \frac{2}{\pi}k\frac{\ln(U^2+1)}{2(1-b/a)} \end{aligned}$$

Set $b/a = \frac{\epsilon/2}{1+\epsilon}$. Then

$$\begin{aligned} & \Pr[Y \ge (1+\epsilon)T] \\ \le\ & \Pr[\exists_i |Y_i| > U] \\ +\ & \Pr[Y \ge (1+\epsilon)T : \forall_i |Y_i| \le U] \\ \le\ & b/a + \frac{1}{1+\epsilon} \\ =\ & \frac{1+\epsilon/2}{1+\epsilon} \end{aligned}$$

where the second last inequality is an application of Markov inequality.

Now we need to show the first condition holds. Define $I_i = [(1+\epsilon)^{i-1}, (1+\epsilon)^i]$ for $i \ge 1$. Let $p_i = \Pr[Y_1 \in I_i]$. We can bound p_i from below as follows:

$$\begin{aligned} p_i &= \frac{2}{\pi}\int_{(1+\epsilon)^{i-1}}^{(1+\epsilon)^i} \frac{1}{1+x^2}dx \\ &\ge \frac{2}{\pi}[(1+\epsilon)^i - (1+\epsilon)^{i-1}] \cdot \frac{1}{1+(1+\epsilon)^{2i}} \\ &= \frac{2}{\pi}\frac{\epsilon(1+\epsilon)^{i-1}}{1+(1+\epsilon)^{2i}} \end{aligned}$$

Let n_i be the number of Y_i's falling to the interval I_i. In the following we introduce a parameter $L \ge 1$ such that for all $i \le l = \log_{1+\epsilon} L$ we have $n_i \ge (1-\epsilon)p_i k$. Note that this implies the following lower bound T' for $|Y|$

$$\begin{aligned} T' &\ge \sum_{i=1}^{l} (1+\epsilon)^{i-1} n_i \\ &\ge \sum_{i=1}^{l} (1+\epsilon)^{i-1}(1-\epsilon)p_i k \\ &\ge \frac{2}{\pi}k(1-\epsilon)\epsilon\sum_{i=1}^{l}\frac{(1+\epsilon)^{i-1}\cdot(1+\epsilon)^{i-1}}{1+(1+\epsilon)^{2i}} \\ &= \frac{2}{\pi}k\frac{1-\epsilon}{(1+\epsilon)^2}\epsilon\sum_{i=1}^{l}\frac{(1+\epsilon)^{2i}}{1+(1+\epsilon)^{2i}} \\ &\ge \frac{2}{\pi}k\frac{1-\epsilon}{(1+\epsilon)^2}\epsilon(l - (\log_{1+\epsilon} 1/\epsilon)/2)\frac{1}{1+\epsilon} \\ &\ge \frac{2}{\pi}k\frac{1-\epsilon}{(1+\epsilon)^3}\epsilon((\ln L)/\epsilon - (\log_{1+\epsilon} 1/\epsilon)/2) \\ &\ge \frac{2}{\pi}k\frac{1-\epsilon}{(1+\epsilon)^3}(\ln L - \epsilon(\log_{1+\epsilon} 1/\epsilon)/2) \end{aligned}$$

We will see later that

$$\ln L \ \ge\ (\log_{1+\epsilon} 1/\epsilon)/2 \qquad (1)$$

Therefore, the final lower bound for T' is $\frac{2}{\pi}(1-\epsilon)^5 k \ln L$.

We will make L as close to U as possible (so that T' is close to T). However, we have to make sure that for all $i \le l$ we have $n_i \ge (1-\epsilon)p_i k$ with probability $1-\delta$. To this end, notice that if $i \le l$ then $p_i \ge q = \frac{2}{\pi}\frac{\epsilon}{1+L(1+\epsilon)}$. Therefore, by Chernoff bound, we just need to make sure that

$$\exp(-\epsilon^2/3 \cdot qk) \le \delta/l$$

The latter condition can be rewritten as

$$k \ge 3/\epsilon^3 \cdot 1/q \cdot (\ln(1/\delta) + \ln l)$$

Substituting for q we get

$$k \ge 3/\epsilon^3 \cdot \pi/2 \cdot (1 + L(1+\epsilon))[\ln\ln_{1+\epsilon} L + \ln 1/\delta]$$

Again, we will see later that

$$\ln\ln_{1+\epsilon} L \ \le\ \ln 1/\delta \qquad (2)$$

Since also $L > 1/\epsilon$, we get that it is sufficient to make sure that

$$k \ge 3\pi/\epsilon^3 \cdot (1+\epsilon)^2 L \ln 1/\delta$$

which can be satisfied by setting

$$L = \frac{\epsilon^3}{3\pi(1+\epsilon)^2}k/\ln(1/\delta) = Ck/\ln(1/\delta)$$

Finally, in order to show that $T' \ge T(1-\epsilon)^f$ for some $f = O(1)$, it is sufficient to ensure that $\ln U \le (1+\epsilon)\ln L$. Substituting for U and L we get the constraint

$$\ln(ak) \le \ln[Ck/\ln(1/\delta)](1+\epsilon)$$

which solves to

$$k \ge [a(\frac{\ln 1/\delta}{C})^{1+\epsilon}]^{1/\epsilon}.$$

Notice that if we take (without loss of generality) $1/\delta = 1/\epsilon^{\Omega(1)}$, then both inequalities (1) and (2) which we assumed on the way are satisfied. $\square$

5 Other results

In this section we sketch other results we obtained using the techniques introduced earlier.

Explicit embedding of l_2^n into $l_1^{n^{O(\log n)}}$ with $(1+1/n^{O(1)})$ distortion. We start from illustrating the embedding by providing an intuitive (although not exactly formal) embedding of l_2^d into l_1 with infinite dimension. To this end, notice that if $X_1 \dots X_n$ is a sequence of i.i.d. random variables with Gaussian distribution, then there exists a constant $c > 0$ such that for any $p = (u_1, \dots, u_n) \in l_2^n$ we have

$$E[|\sum_i u_i X_i|] = c|p|_2$$

(this easily follows from 2-stability of Gaussian distribution and properties of a norm). Thus if we create an "infinite matrix" A with n columns and "infinite" number of rows, one for "each" configuration of $(X_1, \dots, X_n)$, then $|Ap|_1$ is "proportional" to $|p|_2$, which is what we need.

To reduce the dimension of the host space, we proceed essentially as in Section 3. The only difference is that this time we are dealing with the expectation instead of low probability of error (i.e. we have to exclude the case that a small probability event has a significant contribution to the expectation). To this end, we proceed as follows. Let X_i' be i.i.d. variables having the "truncated Gaussian" distribution, i.e. such that:

- if $|X_i| \le t$, then $X_i' = X_i$
- if $|X_i| > t$, then $X_i' = 0$

We use $t = 2c\sqrt{\log n}$, so $\Pr[|X_i| > t] \le a/n^c$, for some $a > 0$. We will relate $E[\sum_i X_i u_i]$ and $E[\sum_i X_i' u_i]$ as follows. Let $p = \Pr[\exists i : |X_i| > t]$; notice that $p \le a/n^{c-1}$, i.e. is small. Then we can write

$$\begin{aligned} E &= E[\sum_i X_i u_i] \\ &= (1-p)E[|\sum_i X_i u_i| : \forall i |X_i| \le t] \\ &+ pE[|\sum_i X_i u_i| : \exists i |X_i| > t] \\ &= (1-p)E_1 + pE_2 \end{aligned}$$

and

$$\begin{aligned} E' &= E[\sum_i X_i u_i] \\ &= (1-p)E[|\sum_i X_i' u_i| : \forall i X_i' \neq 0] \\ &+ pE[|\sum_i X_i' u_i| : \exists i X_i' = 0] \\ &= (1-p)E_1' + pE_2' \end{aligned}$$

Notice that $E_1 = E_1'$. Moreover, it is easy to see that $E_2 = O(nt)$ and $E_2' = O(nt)$. Thus E and E' differ only by a factor of $(1 + 1/n^{\Theta(1)})$. The bounded precision issues are essentially the same as in Section 3, so we skip the details.

Embedding of l_2^n into $l_2^{O_\epsilon(n)}$ with $O_\epsilon(n \log^2 n)$ non-uniform bits and $(1+\epsilon)$ distortion. We follow the usual scheme for non-constructive embeddings [11], i.e.

1. Generate a random $N \times n$ matrix, where $N = O_\epsilon(n)$ (in our case all entries of A are i.i.d. variables with Gaussian distribution)
2. Show that for any vector u such that $|u|_2 = 1$ the value of $|Au|_1$ is sharply concentrated around some $C = C(n)$, i.e. is within the range $[C, (1+\epsilon)C]$ with probability $1 - 2^{-\Omega_\epsilon(n)}$
3. Apply this bound to an epsilon-net of a unit ball in l_2^n and conclude that *all* vectors are distorted by at most $(1+\epsilon)$ factor

However, we are not aware of any proof of the Step 2 for our distribution of A. In particular:

- the matrix in [11] assumes dependence between the columns
- the proof for the $\{-1, 1\}$ matrix in [25] does not give arbitrarily small distortion $(1+\epsilon)$

Therefore, we prove the following

Lemma 4 *There exists a constant $A > 0$ such that*

1. $E(L) = k\sqrt{2/\pi}$
2. *For $0 < \epsilon < 1$ we have*

$$\Pr[L \ge (1+\epsilon)E[L]] \le \exp(-k\epsilon^2 A)$$

and

$$\Pr[L \le (1-\epsilon)E[L]] \le \exp(-k\epsilon^2 A)$$

Note that, modulo the constant A (which has not been optimized here) the dependence of the tail bound on k and ϵ is the same as in the previous section.

Proof: Firstly, recall that all X_i's are distributed according to $N(0,1)$. Let X be a random variable with $N(0,1)$ distribution. Clearly

$$\begin{aligned} E[|X|] &= 1/\sqrt{2\pi}\int_{-\infty}^{\infty} x\exp(-x^2/2)dx \\ &= 1/\sqrt{2\pi}\cdot 2 \\ &= \sqrt{2/\pi} \end{aligned}$$

By linearity of expectation, we get $E[L] = k\sqrt{2/\pi}$.

The sharp concentration of L around its mean is, unfortunately, somewhat more complicated than in the l_2 case, mainly due to the fact that the distribution of L does not seem to have a closed form. Therefore, we use exponential moment inequalities to upper bound the tail of L's distribution.

Let $\beta = 1+\epsilon$. Then for any $a>0$

$$\begin{aligned} & \Pr[\sum |X_i| \geq \beta E[\sum |X_i|]] \\ \leq\ & \Pr[\exp(a\sum |X_i|) \geq \exp(a\beta E[\sum |X_i|])] \\ \leq\ & \frac{E[\exp(a\sum |X_i|)]}{\exp(a\beta E[\sum |X_i|])} \\ =\ & \left(\frac{E[\exp(aX)]}{\exp(a\beta\sqrt{2/\pi})}\right)^k \\ =\ & (N/D)^k \end{aligned}$$

By using Taylor expansion we know that $D \geq 1 + a\beta\sqrt{2/\pi}$. It is sufficient to upper bound $N = \sqrt{2/\pi}\int_0^\infty \exp(ax)\exp(-x^2/2)dx$. To this end, we split N into the sum of N_1 and N_2, where for a $z>0$ defined later we have

$$\begin{aligned} N_1 &= \sqrt{2/\pi}\int_z^\infty \exp(ax)\exp(-x^2/2)dx \\ &= \sqrt{2/\pi}\int_0^\infty \exp(ax+az)\exp(-(x+z)^2/2)dx \\ &\leq \sqrt{2/\pi}\exp(az - z^2/2) \\ &\quad \cdot\int_0^\infty \exp(ax)\exp(-x^2/2)dx \\ &\leq \exp(az-z^2/2)N \end{aligned}$$

Since $\exp(az - z^2/2)$ is small for large z, we can focus only on the initial part of the integral N, namely $N_2 = \sqrt{2/\pi}\int_0^z \exp(ax)\exp(-x^2/2)dx$. In particular, let $z = \frac{1}{2a}$. Then for all $x \leq z$ we have $|ax| \leq 1/2$. Therefore, we know that $\exp(ax) \leq 1 + ax + (ax)^2$. Hence we can write

$$\begin{aligned} N_2 &\leq \sqrt{2/\pi}[\int_0^z \exp(-x^2/2)dx \\ &\quad + \int_0^z ax\exp(-x^2/2)dx \\ &\quad + \int_0^z (ax)^2\exp(-x^2/2)]dx \\ &\leq \sqrt{2/\pi}(\sqrt{\pi/2} + a + a^2\cdot I) \\ &= 1 + \sqrt{2/\pi}a + \sqrt{2/\pi}I\cdot a^2 \end{aligned}$$

where $I = \int_0^z x^2\exp(-x^2/2)dx$ is upper bounded by a constant. Thus we can bound

$$\begin{aligned} N_2/D &\leq \frac{1+\sqrt{2/\pi}a + \sqrt{2/\pi}I\cdot a^2}{1 + a\beta\sqrt{2/\pi}} \\ &= \frac{1+\sqrt{2/\pi}a + \sqrt{2/\pi}I\cdot a^2}{1 + \sqrt{2/\pi}a + \sqrt{2/\pi}\epsilon a} \\ &= 1 - \frac{\sqrt{2/\pi}a(\epsilon - Ia)}{1 + \sqrt{2/\pi}a + \sqrt{2/\pi}\epsilon a} \end{aligned}$$

If we set $a = \frac{\epsilon}{2I}$ then

$$N_2/D \leq 1 - \frac{\sqrt{2/\pi}a(\epsilon/2)}{1 + \sqrt{2/\pi}a + \sqrt{2/\pi}\epsilon a}$$

Recall that $N = N_1 + N_2 = \exp(az - z^2/2)N + N_2$, and thus $N = N_2/(1-\exp(az - z^2/2))$. Also $z = \frac{1}{2a} = I/\epsilon$ and therefore

$$\begin{aligned} N/D &\leq \left(1 - \frac{\sqrt{2/\pi}\frac{\epsilon}{2I}(\epsilon/2)}{1+\sqrt{2/\pi}\frac{\epsilon}{2I} + \sqrt{2/\pi}\epsilon\frac{\epsilon}{2I}}\right) \\ &\quad /\left(1 - \exp(1/2 - \frac{I^2}{2\epsilon^2})\right) \\ &\leq (1 - C_1\epsilon^2)/(1-\exp(1/2 - C_2/\epsilon^2)) \\ &\leq (1 - C_3\epsilon^2) \end{aligned}$$

Therefore

$$\Pr[\sum |X_i| \geq \beta E[\sum |X_i|]] \leq (1 - C_3\epsilon^2)^k$$

which was to be shown.

The second inequality can be proved in *exactly* the same way as shown above, with the difference that $a<0$ and $\beta = 1-\epsilon$. □

Once we have the sharp concentration lemma, we can show that each row of A can be in fact generated using small random seed (as in Section 3). Thus, we need only $O_\epsilon(n\log^2 n)$ bits to represent A.

Acknowledgements. The author would like to thank Martin Strauss and Joan Feigenbaum for helpful discussions.

References

[1] N. Alon, Y. Matias, M. Szegedy, "The space complexity of approximating the frequency moments", STOC'96, pp. 20-29.

[2] B. Berger, "The Fourth Moment Method", *SIAM J. Comput.* 26(4): 1188-1207 (1997).

[3] A. Broder, M. Charikar, A. Frieze, M. Mitzenmacher, "Min-wise independent permutations", STOC'98.

[4] A. Broder, S. Glassman, M. Manasse, and G. Zweig, "Syntactic clustering of the Web", *Proceedings of the Sixth International World Wide Web Conference*, pp. 391-404, 1997.

[5] A. Borodin, R. Ostrovsky and Y. Rabani "Subquadratic Approximation Algorithms For clustering Problems in High Dimensional Spaces", STOC'99.

[6] E. Cohen, M. Datar, S. Fujiwara, A. Gionis, P. Indyk, R. Motwani, J. Ullman and C. Yang, "Finding Interesting Associations Without Support Pruning", ICDE'00.

[7] J. M. Chambers, C. L. Mallows and B. W. Stuck, "A Method for Simulating Stable Random Variables", J. Amer. Statist. Assoc., 71 (1976), pp. 340-344.

[8] D. Eppstein, "Dynamic Euclidean minimum spanning trees and extrema of binary functions", *Disc. Comp. Geom.* 13, pp. 111-122, 1995.

[9] D. Eppstein, "Fast hierarchical clustering and other applications of dynamic closest pairs", SODA'98.

[10] J. Feigenbaum, S. Kannan, M. Strauss, M. Viswanathan, "An Approximate L1-Difference Algorithm for Massive Data Streams", FOCS'99.

[11] T. Figiel, J. Lindenstrauss, V.D. Milman, "The dimension of almost spherical sections of convex bodies", *Acta Math.* 139 (1977), no. 1-2, 53–94.

[12] A. Goel, P. Indyk, K. Varadarajan, "Reductions Among High Dimensional Proximity Problems", manuscript, 1999.

[13] P. Gibbons, Y. Matias, "Synopsis data structures for massive data sets", SODA'99, pp. S909 - S910.

[14] M. Henzinger, P. Raghavan and S. Rajagopalan, "Computing on data streams", Technical Note 1998-011, Digital Systems Research Center, Palo Alto, CA, May 1998.

[15] P. Indyk, "Dimensionality Reduction Techniques for Proximity Problems", accepted to the 11th Symposium on Discrete Algorithms, 2000.

[16] P. Indyk, R. Motwani, "Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality", STOC'98, pp. 604-613.

[17] P. Indyk, R. Motwani, a draft of the final version of the above. Available at `http://theory.stanford.edu/~indyk/nndraft.ps`

[18] P. Indyk, N. Koudas and S. Muthukrishnan, "Identifying Representative Trends in Massive Time Series Datasets Using Sketches", *Proc. VLDB'00*, to appear.

[19] W.B. Johnson and J. Lindenstrauss, "Extensions of Lipshitz mapping into Hilbert space", *Contemporary Mathematics*, 26(1984), pp. 189–206.

[20] W.B. Johnson and G. Schechtman. Embedding l_p^m into l_1^n. *Acta Mathematica*, 149(1982):71–85.

[21] N. Linial, E. London, and Y. Rabinovich. The geometry of graphs and some of its algorithmic applications. In: *Proceedings of 35th Annual IEEE Symposium on Foundations of Computer Science*, 1994, pp. 577–591.

[22] K. Mulmuley, "Randomized Geometric Algorithms and Pseudorandom Generators", *Algorithmica* 16(4/5), pp. 450-463 (1996)

[23] N. Nisan, "Pseudorandom generators for Space-Bounded Computation", STOC'90, pp. 204-212.

[24] J. P. Nolan, "An introduction to stable distributions", available at `http://www.cas.american.edu/~jpnolan/chap1.ps`

[25] G. Schechtman, "Random embeddings of Euclidean spaces in sequence spaces", *Israel J. Math.* 40 (1981), no. 2, 187–192.

[26] V.M. Zolotarev, "One-Dimensional Stable Distributions", Vol. 65 of Translations of Mathematical Monographs , American Mathematical Society (1986).