Stable Distributions, Pseudorandom Generators, Embeddings, and Data Stream Computation

Piotr Indyk
MIT

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 that, for any $p \in (0, 2]$, one can maintain (using only $O(\log n/\epsilon^2)$ words of storage) a *sketch* $C(q)$ of a point $q \in l_p^n$ under dynamic updates of its coordinates. The sketch has the property that, given $C(q)$ and $C(s)$, one can estimate $\|q - s\|_p$ up to a factor of $(1 + \epsilon)$ with large probability. This solves the main open problem of [Feigenbaum et al. 1999].

—We show that the aforementioned sketching approach directly translates into an approximate algorithm that, for a fixed linear mapping A, and given $x \in \Re^n$ and $y \in \Re^m$, estimates $\|Ax-y\|_p$ in $O(n + m)$ time, for any $p \in (0, 2]$. This generalizes an earlier algorithm of [Wasserman and Blum 1997] which worked for the case $p = 2$.

—We obtain another sketch function C' which probabilistically embeds l_1^n into a *normed space* l_1^m. The embedding guarantees that, if we set $m = \log(1/\delta)^{O(1/\epsilon)}$, then for any pair of points $q, s \in l_1^n$, the distance between q and s does not *increase* by more than $(1 + \epsilon)$ with constant probability, and it does not *decrease* by more than $(1 - \epsilon)$ with probability $1 - \delta$. This is the only known dimensionality reduction theorem for the l_1 norm. In fact, stronger theorems of this type (i.e., that guarantee very low probability of expansion as well as of contraction) cannot exist [Brinkman and Charikar 2003].

—We give an explicit embedding of l_2^n into $l_1^{n^{O(\log n)}}$ with distortion $(1 + 1/n^{\Theta(1)})$.

Categories and Subject Descriptors: F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Numerical Algorithms and Problems

General Terms: Algorithms, Theory

Additional Key Words and Phrases: sketching, dimensionality reduction, embeddings, data streams, norms

1. INTRODUCTION

Stable distributions [Zolotarev 1986] are defined as limits of normalized sums of independent identically distributed variables. In particular, a stable distribution with parameter p has the property that for any three independent random variables X, Y, Z drawn from that distribution, and any $a, b \in \Re$, the variables $aX + bY$ and $(|a|^p + |b|^p)^{1/p}Z$ are identically distributed. The most well-known example of a

Part of this work was done while the author was at Stanford University and visiting AT&T Shannon Labs. The author was supported in part by NSF ITR grant CCR-0220280, David and Lucille Packard Fellowship and Alfred P. Sloan Fellowship.

stable distribution is Gaussian (or normal) distribution. However, the class is much wider; for example, it includes heavy-tailed distributions. Stable distributions have found numerous applications in many areas. They are particularly useful in local theory of Banach spaces [Lindenstrauss and Milman], where, e.g., they have been used to show a low-distortion embedding of l_p into l_1 for $p \in (1, 2]$ [Johnson and Schechtman 1982]. However, prior to this work, few applications to theoretical computer science have been known.

In this paper we show that the combination of stable distributions and *bounded space pseudorandom generators* [Nisan 1990] forms a powerful tool for proving a variety of algorithmic results. The high-level idea behind this approach is as follows. Assume that we would like to construct a compact representation of a vector $u \in l_p^n$. It is known (e.g., see [Johnson and Schechtman 1982]) that an inner product of u with a sequence r of n i.i.d. random variables drawn from p-stable distribution has "magnitude" proportional to $\|u\|_p$. This implies that the dot product can be used to recover an approximate value of $\|u\|_p$. Since the inner product $u \cdot r$ can be computed in small space, one can use pseudorandom generators to reduce the number of required truly random bits. This in turn translates into reduction of storage or dimensionality 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, space-efficient dimensionality reduction in l_1 and l_2 and explicit embeddings of l_2 into l_1. Further applications are sketched in Section 6.

Stream computation. The first problem we address is defined as follows (see [Henzinger et al. 1998; Muthukrishnan 2003] 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 see the elements of the stream one by one. Our goal is to approximate (up to the multiplicative factor $(1 \pm \epsilon)$) the l_p norm of the stream S, i.e., the quantity $L_p(S) = \|V(S)\|_p$, where

$$V(S)_i = \sum_{(i,a) \in S} a$$

Estimating the norm of a stream is a fundamental primitive in the growing area of data stream computation, and is used as a subroutine in many streaming algorithms (Section 6.2 for examples). An obvious solution to this problem is to maintain a counter c_i for each i and compute the sum of $|c_i|^p$'s at the end. Unfortunately, this solution requires $\Theta(n)$ words of storage.

In their seminal paper, Alon, Matias and Szegedy [Alon et al. 1996] proposed a randomized scheme for approximating $L_2(S)$ using $O(1/\epsilon^2)$ integers, each $O(\log(n+M))$-bits long. Feigenbaum, Kannan, Strauss and Viswanathan [Feigenbaum et al. 1999] proposed a different algorithm for estimating $L_1(S)$. Their algorithm works in a restricted setting 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 [Feigenbaum et al. 1999] for more details of the model). Computing sketches of normed vectors enables us to compress the data and speed-up computation, e.g., see [Indyk et al. 2000] where this approach was shown to give up to an order of magnitude speed-up for various data-mining problems; see also [Broder et al. 1997; Broder 1998; Cohen et al. 2000] (where a somewhat different similarity measure has been used).

In this paper we propose a unified framework for approximating $L_p(S)$ for $p \in (0, 2]$ in small space. As indicated earlier, our algorithm proceeds by maintaining a dot product of the vector $V(S)$ with a vector r of n independent random variables, each drawn from a p-stable distribution. Since the dot product can be computed in small space, we can generate the random variables using only small number of truly random bits. In this way we make sure that the total storage use is low.

Our algorithm does not have the aforementioned restrictions of [Feigenbaum et al. 1999]; thus, it solves the main open problem from that paper. Moreover, our algorithm maintains only linear combinations of the input values, and therefore extends also to the sketch model.

We note that the algorithms of [Alon et al. 1996] also maintained a dot product $r \cdot V(S)$. However, in their case, the vector r had entries from $\{-1, 1\}$ and was instead drawn from a 4-wise independent family. In this case, the distribution of the dot product $s = V(S) \cdot r$ is not easy to predict. However, it can be shown [Alon et al. 1996] that the second moment of s is equal to $\|V(S)\|_2^2$, so $L_2(S)$ can be estimated (roughly) from the median of squares of several dot products. The advantage of that approach is that 4-wise independent random variables can be generated from only $O(\log(n + M))$ random bits. The disadvantage is that it is not known how to generalize their technique to other $L_p(S)$'s for $p < 2$.

Dimensionality reduction. Dimensionality reduction is a technique that enables to map a set of high-dimensional points into a set of points in low-dimension, such that both sets have similar "distance properties". This technique, especially the result of Johnson and Lindenstrauss [Johnson and Lindenstrauss 1984] for the l_2 norm, found numerous applications in theoretical computer science (cf. [Indyk 2001]). We observe that the aforementioned sketching results can be viewed as low-storage dimensionality reduction theorems. 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 the l_p distance between the points. Then the sketch operator C can be viewed as a mapping of l_p^n into the "sketch space" (say $\mathcal{C}$), such that

—each point in $\mathcal{C}$ can be described using only m numbers, where m is "small"
—the value of $L_p(S_r, S_b)$ is approximately equal to $F(C(S_r), C(S_b))$

Unfortunately, our sketches (as well as the sketches of [Alon et al. 1996]) have the undesirable property that the pair $(\mathcal{C}, F)$ is not a *normed space.* Specifically, the definition of F involves the median operator; e.g. for l_1 we have

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

The fact that F is not a norm significantly restricts the applications of the mapping C as a dimensionality reduction technique. This is because it prohibits the usage of a large number of algorithms designed for normed spaces. To overcome this obstacle we proceed as follows. For l_2, we observe that if we modify our algorithm by replacing the median by $\|\cdot\|_2$, then the accuracy of the estimation does not change (this follows by observing that the dimensionality reduction lemma of Johnson-Lindenstrauss requires few truly random bits). This gives a small-space/streaming version of the Johnson-Lindenstrauss lemma.

For l_1, the situation is more complicated, since for sketch points $(x_1, \ldots, x_m), (y_1, \ldots, y_m)$ the expectation $E[\sum_i |x_i - y_i|]$ is undefined (i.e. is equal to ∞). Thus, we cannot simply replace the median by $\|\cdot\|_1$. However, we are able to show that for any $\gamma > 0$ there exists a sketch function C which maps the points into $m = \Theta(\ln(1/\delta)^{1/(\epsilon-\gamma)})$-dimensional space $\Re^m$ 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 $1-(1+\gamma)/(1+\epsilon)$ (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 the l_1 norm. The two-sided analog is impossible by the result of [Brinkman and Charikar 2003]. 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 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).

Deterministic embeddings of l_2 into l_1. The study of low-distortion embeddings between normed spaces is a rich area of study in mathematics. One of the major results in that area (e.g. see [Figiel et al. 1977] and references therein) is that l_2^n can be embedded into $l_1^{O(n)}$ with distortion $(1+\epsilon)$ (the $O()$ constant depends on ϵ). Unfortunately, none of the many proofs of this theorem is constructive, since they use probabilistic method to construct the embeddings. To our knowledge, the only constructive result of this type [Berger 1997; Linial et al. 1994] embeds l_2^n into $l_1^{O(n^2)}$ with $\sqrt{3}$ distortion.

We provide an explicit embedding of l_2^n into $l_1^{n^{O(\log n)}}$ with distortion $(1+1/n^{\Theta(1)})$. By combining the result with the deterministic nearest neighbor algorithm of [Indyk 2000] we obtain a $(3+\epsilon)$-approximate deterministic algorithm for the nearest neighbor search in l_2^n. The algorithm uses preprocessing and storage that is polynomial in the number N of input points and $n^{\log n}$, and has query time that is polynomial in $n^{\log n}$ and $\log N$.

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 [Zolotarev 1986] 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

In general, a random variable X from a p-stable distribution can be generated [Chambers et al. 1976] by taking:

$$X = \frac{\sin(p\Theta)}{\cos^{1/p}\Theta}\left(\frac{\cos(\Theta(1-p))}{-\ln r}\right)^{(1-p)/p}$$

where Θ is uniform on $[-\pi/2, \pi/2]$ and r is uniform on $[0, 1]$.

Pseudorandom generators (PRGs). To reduce the randomness needed to generate a vector of random variables from a stable distribution, we use pseudorandom generators for bounded space computation [Nisan 1990]. The intuition is that our algorithms perform only a dot product of the random vector with the vector $V(S)$, and the dot product can be computed in small space.

As in [Nisan 1990] 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 the 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 $Q \in \mathcal{C}$

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

FACT 1 [NISAN 1990]. *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 $M \ge n$, and that the number of pairs in the stream S is polynomial in 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.

3. APPROXIMATION OF THE L_P NORM OF 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. For simplicity, we focus exclusively on the problem of estimating $L_p(S)$; the algorithms automatically generate the sketches of S as well.

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

(1) It assumes infinite precision of the calculations (i.e. it 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 the 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 $X_i^j, i \in [n], j \in [l]$ drawn from 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$(|S^0|, \ldots |S^{l-1}|)$

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 $S^0 \ldots S^{l-1}$. 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$(|X|) = 1$. □

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

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

Proof. An easy application of Chernoff bound. □

Lemma 2. *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 have infinite precision. Since the numbers in the data stream are integer, we only need to take care of the random variables X_i^j. Specifically, we need to show that it is sufficient to assume that the random variables can be represented using $O(\log(n + M))$ bits.

Firstly, we state the following

CLAIM 3. *Let $f : [0, 1]^d \to \Re$ be a function computed by an algorithm that, given an input $x \in [0, 1]^d$ where each coordinate x_i is represented using b bits of precision, computes $f(x)$ with an additive error $\beta > 0$, using $O(d(b+\log(1/\beta)))$ bits of space. In addition, assume that there exists $P > 0$, such that for all $x \in [P, 1 - P]^d$ the absolute values of the first order partial derivatives of f at x are at most B. Define a random variable $X = f(U)$, where U is chosen from the uniform distribution over $[0, 1]^d$.*

There is an algorithm A that for any $\alpha > 0$ generates a random variable $\tilde{X}$, such that there is a joint probability space of X and $\tilde{X}$ so that:

—$\Pr[|X - \tilde{X}| > \alpha] \leq 2d(P + \frac{\alpha}{Bd})$
—*A uses only $O(d[\log(1/\alpha + B + d) + b])$ random and storage bits*

PROOF. Follows from a standard discretization argument. Let $s = \frac{\alpha}{Bd}$. Impose a cubic grid on $[0, 1]^d$, where each cell has side length s. Note that the total volume of cells fully contained in $[P, 1 - P]^d$ is at least $1 - 2d(P + s)$.

We define $\tilde{X} = f(\tilde{U})$, where each coordinate of $\tilde{U}$ is chosen uniformly at random from $\{0, 1/2^b, \ldots, 1-1/2^b\}$. This corresponds to choosing a random U from $[0, 1]^d$, and rounding each coordinate down to the nearest multiple of $1/2^b$. If the grid cell containing U is fully contained in $[P, 1 - P]^d$, then $|f(U) - f(\tilde{U})| \leq Bds = \alpha$. □

In our case, $U \in [0, 1]$, and we have $X = f(U) = \tan(\pi U/2)$. One can observe that the derivative of f is bounded by $O(1/P^2)$ in the interval $[P, 1-P]$. Thus, for each i, j, we can generate an approximation $\tilde{X}_i^j$ to each X_i^j using only $O(\log(1/P+1/\alpha)$ bits, such that for each of them $|\tilde{X}_i^j - X_i^j| \leq \alpha$ with probability $1 - P$. It follows that, with probability at least $1 - nlP$, for all j we have

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

Since $\text{median}(S^j) = \sum_i |c_i|$,we can set $\alpha = \epsilon$ to ensure that the estimation of $L_1(S)$ is within a factor of $(1 \pm 2\epsilon)$ from the true value. We also set $P = \frac{\delta}{nl}$ to ensure that the probability of correct estimation is at least $1-2\delta$. Finally, we need only $O(\log(n + 1/\delta + 1/\epsilon))$ random bits to generate each random variable.

Randomness reduction. Consider a fixed S^j. From the above it follows that the value of S^j can be represented using small number of bits; also, we need only small number of 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.

LEMMA 3. *Consider an algorithm A which, given a stream S of pairs (i, a), and a function $f : [n] \times \{0,1\}^R \times \{-M \dots M\} \to \{-M^{O(1)} \dots M^{O(1)}\}$, does the following:*

—*Set $O = 0$; Initialize length-R chunks $R_0 \dots R_M$ of independent random bits*
—*For each new pair (i, a): perform $O = O + f(i, R_i, a)$*
—*Output $A(S) = O$*

Assume that the function $f(\cdot,\cdot,\cdot)$ is computed by an algorithm using $O(C + R)$ space and $O(T)$ time. Then there is an algorithm A' producing output $A'(S)$, that uses only $O(C + R + \log(Mn))$ bits of storage and $O([C + R + \log(Mn)] \log(nR))$ random bits, such that

$$\Pr[A(S) \neq A'(S)] \leq 1/n$$

over some joint probability space of randomness of A and A'. Then the algorithm A' uses $O(T + \log(nR))$ arithmetic operations per each pair (i, a).

PROOF. Consider a stream $sort(S)$ in which $(i, a) \in S$ appear in the increasing order of i's. In this case we do not have to store the chunks R_i, since we can generate them on the fly. Thus, the algorithm uses only $O(\log(nM) + C + R)$ storage and $O(nR)$ bits of randomness. Therefore, by Fact 1, there exists a PRG which, given a random seed of size $O([C + R + \log(Mn)] \log(nR))$, expands it to a sequence $R'_0 \dots R'_{n-1}$, such that using R'_i's instead of R_i's results in negligible probability of error. That is, if A' is the algorithm using variables R'_i, then

$$\Pr[A(sort(S)) \neq A'(sort(S))] \leq 1/n$$

However, since the addition is commutative, we have $A(sort(S)) = A(S)$ and $A'(sort(S)) = A'(S)$. The lemma follows. □

The theorem stating the correctness of the final algorithm for estimating $L_1(S)$ is deferred to the next section.

3.1 Computing $L_2(S)$

In this section we describe the modifications for the case of $p = 2$. Note that the algorithms given in this section use *more* space than the earlier algorithm of [Alon et al. 1996]. However, the second algorithm has the following appealing property: the sketch of the stream S is computed by taking $y = AV(S)$ where A is an (implicitly defined) matrix, and $L_2(S)$ is estimated by taking $\|y\|_2$. In other words, the algorithm provides a streaming version of the dimensionality reduction theorem by Johnson and Lindenstrauss [Johnson and Schechtman 1982], which has the benefits as stated in the introduction.

The first algorithm is obtained by replacing Cauchy distribution by Gaussian distribution. As before, the final estimator is a median of $|S^0| \dots |S^{l-1}|$. The distribution function of the normal distribution is differentiable and has non-zero derivative around its median, so the analog of Lemma 2 still holds. Moreover, a random variable having normal distribution can be generated from a pair chosen uniformly at random from $[0, 1]^2$ using the formula given in Preliminaries. Then, using elementary analysis, one can verify that Claim 3 holds with $B = 1/P^{O(1)}$.

Theorem 2. *There is an algorithm which for any $0 < \epsilon, \delta < 1$ estimates $L_1(S)$ or $L_2(S)$ up to a factor $(1 \pm \epsilon)$ with probability $1 - \delta - 1/n$ and uses*

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

However, a more elegant approach to estimating $L_2(S)$ is to replace the median operator in the algorithm by $\| \cdot \|_2$. Specifically, the modified algorithm returns $\|(S^0, \ldots, S^{l-1})\|_2$ as the estimation of $L_2(S)$. The correctness of the algorithm follows by a combination of two facts:

—If we use truly independent normal variables X_i^j, then the algorithm is correct [Indyk and Motwani 1998].
—If the random variables are instead created using Nisan's generator, the resulting difference in the probability of correctness is negligible. This can be shown in the same way as for the median-based algorithm.

This gives us the following streaming version of Johnson-Lindenstrauss lemma:

Theorem 3. *There is an algorithm that for any $0 < \epsilon, \delta < 1$ constructs an implicit representation of a $k \times n$ matrix A, $k = O(\log(1/\delta)/\epsilon^2)$, such that:*

—*Given any $i = 1 \ldots k$, $j = 1 \ldots n$, the algorithms returns $A[i, j]$ after performing $O(\log n)$ arithmetic operations.*
—*The algorithm uses $O(\log(Mn/(\delta\epsilon)) \log(n/(\epsilon\delta)) \log(1/\delta)/\epsilon^2)$ bits of space.*
—*Each entry of A can be represented using $O(\log(n/(\delta\epsilon))$ bits.*
—*For any fixed vector $x \in \Re^n$, we have*

$$\Pr[|\|Ax\|_2 - \|x\|_2| > \epsilon \|x\|_2] \leq \delta$$

3.2 Computing $L_p(S)$

For general $p \in (0, 2]$ the algorithm and analysis become more involved, mainly due to the fact that no exact formulas are known for densities and/or distribution functions of general p-stable distribution. However, one can generate p-stable random variables as in Preliminaries. Therefore, the algorithm from earlier section can be *implemented* for general p. As far as the analysis is concerned, it seems that an analog of Lemma 2 does hold for any $p \in (0, 2)$. Unfortunately, we are not aware of any proof of this fact. Instead, we show the following lemma.

Lemma 4. *Let F be a c.d.f. of a random variable $|Z|$, where Z is drawn from a p-stable distribution. There exist constants $c_1, c_2, c_3 > 0$, such that for any p and ϵ, there exists $t \in [c_1, c_2]$ such that $|F^{-1}(t - \epsilon/c_3) - F^{-1}(t + \epsilon/c_3)| \leq \epsilon$.*

Proof. Let $b_1, b_2 > 0$ be constants such that $\Pr[|Z| \geq b_1] \leq 1 - b_2$. Consider $v > 0$ such that $F(v) = b_2$. Clearly, we have $v \leq b_1$. Also, let $u > 0$ be such that $F(u) = b_2/2$. Decompose the interval $[b_2/2, b_2]$ into b_1/ϵ disjoint intervals of the form $[t - \epsilon/(4b_1/b_2), t + \epsilon/(4b_1/b_2)]$. Assume the lemma does not hold with constants $c_1 = b_2/2, c_2 = b_2, c_3 = (4b_1/b_2)$. This implies that F^{-1} increases on each of the intervals by more than ϵ. But this would imply that $F^{-1}(b_2) > b_1/\epsilon \cdot \epsilon = b_1$, which yields a contradiction.

□

Given the lemma, we can estimate the value of $L_p(S)$ by taking the t-quantile (instead of the median) of variables $|S^0| \dots |S^{l-1}|$. Note that, unlike for $p = 1, 2$, the value of t (and therefore the algorithm) depends on $\epsilon > 0$. Moreover, we do not specify a method to compute t given p and ϵ, although presumably this task can be accomplished by using numerical approximations of the densities of p-stable distributions. This means that our algorithm is not uniform.

Other issues are taken care of as for $p = 2$. We only observe that for general p, the derivative of F^{-1} depends on p. However, since we consider p to be a constant, we suppress the dependence on p in the $O(\cdot)$ notation below.

THEOREM 4. *For any $p \in (0, 2)$ and any $0 < \epsilon, \delta < 1$, there is a non-uniform algorithm that estimates $L_p(S)$ up to a factor $(1 \pm \epsilon)$ with probability $1 - \delta$ and uses*

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

The $O(\cdot)$ notation subsumes constants depending on p.

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.

THEOREM 5. *For any $1/2 \geq \epsilon, \delta > 0$, and $\epsilon > \gamma > 0$, there is a probability space over linear mappings $f : l_1^n \to l_1^k$, where $k = (\ln(1/\delta))^{1/(\epsilon-\gamma)}/c(\gamma)$, for a function $c(\gamma) > 0$ depending only on γ, such that for any pair of points $p, q \in l_1^n$:*

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

Note that the embedding is randomized but asymmetric: the probability that the expansion is small is only about ϵ, while the probability that the contraction is small is $1 - \delta$. Also, note that the term $c(\gamma)$ in the definition of k enables us to assume that k is "large" compared to any function of γ.

PROOF. We define the random mapping f such that, for $j = 1 \dots k$ the jth coordinate of $f(q)$ for $q = (q_1, \dots, q_n)$ is equal to $Y_j = \sum_i X_i^j q_i$, where X_i^j are i.i.d random variables having Cauchy distribution. Since f is linear, it is sufficient to show the above for $p = 0$ and q such that $\|q\|_1 = 1$. In this case $\|f(p) - f(q)\| = \sum_j |\sum_i X_i^j q_i| = \sum_j |Y_j|$. 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 $Y = \sum_j |Y_j|$. Show that there exists a *threshold* $T = T(k, \gamma, \epsilon)$, such that:

—$\Pr[Y < (1-\epsilon)T] \le \delta$
—$\Pr[Y > (1+\epsilon)T] \le \frac{1+\gamma}{1+\epsilon}$

Our approach is to consider a "truncated" version of the variables Y_j. In particular, for $B > 0$ and any $j = 1 \ldots k$, let Z_j^B be equal to $|Y_j|$ if $|Y_j| \le B$; we will set $Z_j^B = B$ otherwise.

Claim 4. *Let $P = \Pr[Z_j^B = B]$. Then $P \le b/B$ for some constant $b > 0$.*

Lemma 5. *For any $B > 0$*

$$k \ln(B^2+1)/\pi \le E[\sum_j Z_j^B] \le k[\ln(B^2+1)/\pi + b]$$

Proof. We have

$$\begin{aligned} E[\sum_j Z_j^B] &= \sum_j E[Z_j^B] \\ &= k[(1-P)\frac{2}{\pi}\int_0^B \frac{x}{1+x^2}dx + PB] \\ &= k[(1-P)/\pi \cdot \ln(B^2+1) + PB] \end{aligned}$$

The inequalities follow from Claim 4. □

Lemma 6. *For any $B > 0$*

$$E[(Z_j^B)^2] \le 4/\pi \cdot B$$

Proof.

$$E[(Z_j^B)^2] = 2/\pi\left[\int_0^B \frac{x^2}{1+x^2}dx + \int_B^\infty \frac{B^2}{1+x^2}dx\right] \le 2/\pi \cdot [B + B^2/B] = 4/\pi \cdot B$$

□

We will first establish T which satisfies the second condition. Let $U = \frac{1+\epsilon}{\gamma}bk$. By the union bound we have $\Pr[\exists_j |Y_j| \ge U] \le kb/U = \frac{\gamma}{1+\epsilon}$. We define $T = E[\sum_j Z_j^U]$. Then

$$\begin{aligned} \Pr[Y \ge (1+\epsilon)T] &\le \Pr[\exists_j |Y_j| > U] + \Pr[Y \ge (1+\epsilon)T : \forall_j |Y_j| \le U] \\ &\le \frac{\gamma}{1+\epsilon} + \frac{E[\sum_j |Y_j| : \forall_j |Y_j| \le U]}{(1+\epsilon)T} \\ &\le \frac{\gamma}{1+\epsilon} + \frac{E[\sum_j Z_j^U]}{(1+\epsilon)T} \\ &= \frac{1+\gamma}{1+\epsilon} \end{aligned}$$

Now we focus on the first condition. Define $\alpha = \frac{1-\epsilon}{1-\gamma}$, $L = U^\alpha$. Observe that $1/2 \le \alpha < 1$. Observe that

$$E[\sum_j Z_j^L] \ge k/\pi \cdot \ln(L^2+1)$$

$$\geq k/\pi \cdot \ln(U^{2\alpha}+1)$$
$$\geq \alpha k/\pi \cdot \ln(U^2+1)$$
$$\geq \alpha(T-bk)$$

where the last inequality follows from Lemma 5 and the definition of T. Thus, $T \leq E[\sum_j Z_j^L]/\alpha + bk$. Set $\gamma' = \gamma/2$, and assume that k is "large enough" with respect to γ'. We have

$$\begin{aligned}
\Pr[\sum_j |Y_j| \leq (1-\epsilon)T] &\leq \Pr[\sum_j Z_j^L \leq (1-\epsilon)T] \\
&\leq \Pr[\sum_j Z_j^L \leq (1-\epsilon)(E[\sum_j Z_j^L]/\alpha + bk)] \\
&= \Pr[\sum_j Z_j^L \leq (1-\gamma)E[\sum_j Z_j^L] + (1-\epsilon)bk] \\
&= \Pr[E[\sum_j Z_j^L] - \sum_j Z_j^L \geq \gamma' E[\sum_j Z_j^L] + (\gamma' E[\sum_j Z_j^L] - (1-\epsilon)bk)]
\end{aligned}$$

Observe that by Lemma 5:

$$\begin{aligned}
\gamma' E[\sum_j Z_j^L] - (1-\epsilon)bk &\geq k[\gamma'/\pi \cdot \ln(L^2+1) - b] \\
&\geq k[\gamma'/\pi \cdot \ln(U^{2\alpha}) - b] \\
&\geq k[\gamma'/\pi \cdot \ln(bk) - b]
\end{aligned}$$

which is positive for k "large enough". Therefore, we have

$$\Pr[\sum_j |Y_j| \leq (1-\epsilon)T] \leq \Pr[E[\sum_j Z_j^L] - \sum_j Z_j^L \geq \gamma' E[\sum_j Z_j^L]]$$

By using the inequality of [Maurer 2003] we get

$$\begin{aligned}
\Pr[E[\sum_j Z_j^L] - \sum_j Z_j^L \geq \gamma' E[\sum_j Z_j^L]] &\leq \exp(-\frac{\gamma'^2 E^2[\sum_j Z_j^L]}{2kE[(Z_j^L)^2]}) \\
&\leq \exp(-\frac{\gamma'^2(\alpha k/\pi \cdot \ln(U^2+1))^2}{2k \cdot 4/\pi \cdot L}) \\
&\leq \exp(-\frac{\gamma'^2(\alpha k/\pi)^2}{2k[(1+\epsilon)bk/\gamma]^\alpha}) \\
&\leq \exp(-c(\gamma)k^{1-\alpha}) \\
&\leq \exp(-c(\gamma)k^{\epsilon-\gamma})
\end{aligned}$$

where $c(\gamma)$ is a constant dependent on γ.

Thus, setting $k = \ln(1/\delta)^{1/(\epsilon-\gamma)}/c(\gamma)$ for a proper function $c(\gamma)$ ensures that the first condition of the theorem holds as well.

5. 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 explicit embedding of l_2^d into l_1 with large 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 $q = (q_1, \dots, q_n) \in l_2^n$ we have

$$E[|\sum_i q_i X_i|] = C\|q\|_2$$

(this easily follows from 2-stability of Gaussian distribution and properties of a norm). This is approximately true even if the Gaussian variables are discretized to be representable using $b = O(\log n)$ bits; the details are as in Section 3. Thus if we create a matrix A with n columns and $(2^b)^n$ rows, one for each configuration of $(X_1, \dots, X_n)$, then $\|Aq\|_1/(2^b)^n \approx C\|q\|_2$, 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 $E[|\sum_i X_i q_i|]$ and $E[|\sum_i X'_i q_i|]$ 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

$$\begin{aligned} E &= E[|\sum_i X_i q_i|] \\ &= (1-P)E[|\sum_i X_i q_i| : \forall i |X_i| \le t] + PE[|\sum_i X_i q_i| : \exists i |X_i| > t] \\ &= (1-P)E_1 + PE_2 \end{aligned}$$

and

$$\begin{aligned} E' &= E[|\sum_i X_i q_i|] \\ &= (1-P)E[|\sum_i X'_i q_i| : \forall i X'_i \ne 0] + PE[|\sum_i X'_i q_i| : \exists i X'_i = 0] \\ &= (1-P)E'_1 + PE'_2 \end{aligned}$$

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.

THEOREM 6. *For any $n > 0$ there exists an explicitly constructible embedding of l_2^n into $l_1^{n^{O(\log n)}}$ with distortion $(1 + 1/n^{O(1)})$.*

6. EXTENSIONS, DISCUSSION AND OPEN PROBLEMS

6.1 Approximate result checking

The technique of using a random linear mapping to estimate the norm of a vector has its computer science roots in (approximate) checking of computation. Consider the following problem: for a fixed linear mapping $A : \Re^n \rightarrow \Re^m$, construct a "checker", that given $x \in \Re^n$ and $y \in \Re^m$, checks if $Ax = y$. The check should be preferably done in time $O(n)$, so that the overhead of checking is low compared to the computation time. The latter is typically $\omega(n)$, e.g., it is $\Theta(n \log n)$ for Fourier Transform.

A solution to this problem [Freivalds 1979; Wasserman and Blum 1997] can be obtained as follows. Firstly, observe that for any $r \in \Re^m$ we have $r^T(Ax) = (r^T A)x = s^T x$. Moreover, if $Ax \neq y$, then for r chosen uniformly at random from $\{0, 1\}^m$ we have $\Pr[s^T x = r^T y] = \Pr[r^T(Ax - y) = 0] \leq \frac{1}{2}$. These two observations give us a probabilistic checker for $Ax = y$ that runs in $O(n)$ time, provided we generate the pair (r, s) in advance.

A more refined *approximate* checker was proposed in [Ar et al. 1993; Wasserman and Blum 1997]. It not only verifies if $Ax = y$, but also enables to estimate the *norm* of the difference vector $Ax - y$. In particular, [Wasserman and Blum 1997] observes that $(s^T x - r^T y)^2$ provides an unbiased estimator of $\|Ax - y\|_2^2$.

In the context of the aforementioned research, one can easily see that our sketching algorithms can be directly translated into approximate checkers that work for any l_p norm, $p \in (0, 2]$.

6.2 Further developments

Since the earlier version of this paper has been presented at FOCS'00, the techniques introduced in this paper have been used in several other papers. In this section we briefly discuss those results.

Our algorithms for estimating the l_p norms, as well as the use of Nisan's generator to reduce the storage needed for the random bits, become a standard tool in the area of streaming algorithms (cf. [Gilbert et al. 2002; Datar et al. 2002; Cormode et al. 2002; Thaper et al. 2002; Cormode and Muthukrishnan 2003; Indyk 2004; Indyk and Woodruff 2005]; see also the surveys [Cormode 2003; Muthukrishnan 2003]). In [Cormode et al. 2002] the authors use the algorithms in a non-streaming setting to reduce the dimensionality of the data and the running time needed to compute distances between the vectors.

Stable distributions found use in other algorithmic settings as well. In [Datar et al. 2004] they are used to construct a Locality-Sensitive Hashing scheme that works directly in l_p norms; the earlier scheme of [Indyk and Motwani 1998] works only for Hamming space. In [Feigenbaum et al. 2001a; 2001b] (Appendix B.2), it is showed how to augment our l_2 estimation algorithm to construct sketches that are *cryptographically secure*. Specifically, the authors use the "memoryless" property of p-stable distributions: a dot-product of any vector $x \in \Re^n$ with a vector of n independent p-stable random variables is a random variable that depends only on $\|x\|_p$ and not on any other properties of x.

The observation that Nisan's generator can be used to reduce the randomness needed for dot product computation has been used in [Engebretsen et al. 2002] to give an efficient derandomization of an approximation algorithm based on semidefinite programming. Their algorithm was fairly complex and involved the method of conditional probabilities in addition to the use of pseudorandom generator. Independently, Sivakumar [Sivakumar 2002] showed that a similar result (as well as many others) can be obtained directly, by using a different version of Nisan's generator [Nisan 1992].

6.3 Open problems

There are several interesting problems left open by this paper. In particular, we do not know if the use of Nisan's generator is really necessary for our purpose. It is plausible that one could use $O(1)$-wise independent families of random variables (as in [Alon et al. 1996]) to generate random variables which have "sufficient" stable law properties. If so, then one would be able to reduce the space used by our algorithm by a logarithmic factor. Even better, this might give an explicit construction of an embedding of l_2^d into $l_1^{d^{O(1)}}$ with distortion arbitrarily close to 1.

In general, closing the gap between probabilistic and explicit constructions of such embeddings remains an important open problem (cf. [Matoušek 2004], Problem 2.2).

ACKNOWLEDGMENTS

The author would like to thank Martin Strauss, Joan Feigenbaum, Graham Cormode, Anastasios Sidiropoulos and the anonymous referees for helpful comments and discussions.

REFERENCES

Alon, N., Matias, Y., and Szegedy, M. 1996. The space complexity of approximating the frequency moments. *Proceedings of the Symposium on Theory of Computing*, 20–29.

Ar, S., Blum, M., Codenotti, B., and Gemmell, P. 1993. Checking approximate computation over the reals. *Annual ACM Symposium on Theory of Computing*.

Berger, B. 1997. The fourth moment method. *SIAM Journal on Computing 26*.

Brinkman, B. and Charikar, M. 2003. On the impossibility of dimension reduction in l1. *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*.

Broder, A. 1998. Filtering near-duplicate documents. *Proceedings of FUN*.

Broder, A., Glassman, S., Manasse, M., and Zweig, G. 1997. Syntactic clustering of the web. *Proceedings of the Sixth International World Wide Web Conference*, 391–404.

Chambers, J. M., Mallows, C. L., and Stuck, B. W. 1976. A method for simulating stable random variables. *J. Amer. Statist. Assoc. 71*, 340–344.

Cohen, E., Datar, M., Fujiwara, S., Gionis, A., Indyk, P., Motwani, R., Ullman, J., and Yang, C. 2000. Finding interesting associations without support prunning. *Proceedings of the 16th International Conference on Data Engineering (ICDE).*

Cormode, G. 2003. Stable distributions for stream computations: it's as easy as 0,1,2. *Proceedings of the Workshop on Management and Processing of Data Streams*.

Cormode, G., Datar, M., Indyk, P., and Muthukrishnan, S. 2002. Comparing data streams using hamming norms. *Proceedings of the International Conference on Very Large Databases (VLDB).*

Cormode, G., Indyk, P., Koudas, N., and Muthukrishnan, S. 2002. Fast mining of massive tabular data via approximate distance computations. *Proc. 18th International Conference on Data Engineering (ICDE).*

Cormode, G. and Muthukrishnan, S. 2003. Estimating dominance norms of multiple data streams. *European Symposium on Algorithms*.

Datar, M., Gionis, A., Indyk, P., and Motwani, R. 2002. Maintaining stream statistics over sliding windows. *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*.

Datar, M., Immorlica, N., Indyk, P., and Mirrokni, V. 2004. Locality-sensitive hashing scheme based on p-stable distributions. *Proceedings of the ACM Symposium on Computational Geometry*.

Engebretsen, L., Indyk, P., and O'Donnell, R. 2002. Deterministic dimensionality reduction with applications. *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*.

Feigenbaum, J., Ishai, Y., Malkin, T., Nissim, K., Strauss, M. J., and Wright, R. N. 2001a. Secure multiparty computation of approximations. *Lecture Notes in Computer Science 2076*, 927.

Feigenbaum, J., Ishai, Y., Malkin, T., Nissim, K., Strauss, M. J., and Wright, R. N. 2001b. Secure multiparty computation of approximations. `http://eprint.iacr.org/2001/024/`.

Feigenbaum, J., Kannan, S., Strauss, M., and Viswanathan, M. 1999. An approximate l1-difference algorithm for massive data streams. *Proceedings of the Symposium on Foundations of Computer Science*.

Figiel, T., Lindenstrauss, J., and Milman, V. D. 1977. The dimension of almost spherical sections of convex bodies. *Acta Mathematica 139*, 53–94.

Freivalds, R. 1979. Fast probabilistic algorithms. *Mathematical Foundations of Computer Science LNCS 74*.

Gilbert, A., Guha, S., Kotidis, Y., Indyk, P., Muthukrishnan, M., and Strauss, M. 2002. Fast, small-space algorithms for approximate histogram maintenance. *Annual ACM Symposium on Theory of Computing*.

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

Indyk, P. 2000. Dimensionality reduction techniques for proximity problems. *Proceedings of the Ninth ACM-SIAM Symposium on Discrete Algorithms*.

Indyk, P. 2001. Tutorial: Algorithmic applications of low-distortion geometric embeddings. *Annual Symposium on Foundations of Computer Science*.

Indyk, P. 2004. Algorithms for dynamic geometric problems over data streams. *Annual ACM Symposium on Theory of Computing*.

Indyk, P., Koudas, N., and Muthukrishnan, S. 2000. Identifying representative trends in massive time series datasets using sketches. *Proceedings of the 26th International Conference on Very Large Databases (VLDB).*

Indyk, P. and Motwani, R. 1998. Approximate nearest neighbor: towards removing the curse of dimensionality. *Proceedings of the Symposium on Theory of Computing*.

Indyk, P. and Woodruff, D. 2005. Optimal approximations of the frequency moments of data streams. *Annual ACM Symposium on Theory of Computing*.

Johnson, W. and Lindenstrauss, J. 1984. Extensions of lipshitz mapping into hilbert space. *Contemporary Mathematics 26*, 189–206.

Johnson, W. and Schechtman, G. 1982. Embedding l_p^m into l_1^n. *Acta Mathematica 149*, 71–85.

Lindenstrauss, J. and Milman, V. Local theory of normed spaces and convexity. *Handbook of convex geometry B*, 1149–1220.

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

Matoušek, J. 2004. Collection of open problems on low-distortion embeddings of finite metric spaces. *Available at* `http://kam.mff.cuni.cz/~matousek/metrop.ps.gz`.

Maurer, A. 2003. A bound on the deviation probability for sums of non-negative random variables. *Journal of Inequalities in Pure and Applied Mathematics 4(1)*, Art. 15.

Muthukrishnan, S. 2003. Data streams: Algorithms and applications (invited talk at soda'03). *Available at http://athos.rutgers.edu/∼muthu/stream-1-1.ps.*

Nisan, N. 1990. Pseudorandom generators for space-bounded computation. *Annual ACM Symposium on Theory of Computing*, 204–212.

NISAN, N. 1992. RL $\subset$ SC. *Annual ACM Symposium on Theory of Computing*, 619–623.

SIVAKUMAR, D. 2002. Algorithmic derandomization via complexity theory. *Annual ACM Symposium on Theory of Computing*, 619–626.

THAPER, N., GUHA, S., INDYK, P., AND KOUDAS, N. 2002. Dynamic multidimensional histograms. *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*.

WASSERMAN, H. AND BLUM, M. 1997. Software reliability via run-time result checking. *Journal of the ACM*.

ZOLOTAREV, V. 1986. *One-Dimensional Stable Distributions*. Vol. 65 of Translations of Mathematical Monographs, American Mathematical Society.