Streaming and Sublinear Approximation of Entropy and Information Distances

Sudipto Guha* Andrew McGregor* Suresh Venkatasubramanian†

Abstract

In most algorithmic applications which compare two distributions, information theoretic distances are more natural than standard ℓ_p norms. In this paper we design streaming and sublinear time property testing algorithms for entropy and various information theoretic distances.

Batu *et al* posed the problem of property testing with respect to the Jensen-Shannon distance. We present optimal algorithms for estimating bounded, symmetric *f-divergences* (including the Jensen-Shannon divergence and the Hellinger distance) between distributions in various property testing frameworks. Along the way, we close a $(\log n)/H$ gap between the upper and lower bounds for estimating entropy H, yielding an optimal algorithm over all values of the entropy. In a data stream setting (sublinear space), we give the first algorithm for estimating the entropy of a distribution. Our algorithm runs in polylogarithmic space and yields an asymptotic constant factor approximation scheme. An integral part of the algorithm is an interesting use of an F_0 (the number of distinct elements in a set) estimation algorithm; we also provide other results along the space/time/approximation tradeoff curve.

Our results have interesting structural implications that connect sublinear time and space constrained algorithms. The mediating model is the *random order streaming model*, which assumes the input is a random permutation of a multiset and was first considered by Munro and Paterson in 1980. We show that any property testing algorithm in the combined oracle model for calculating a permutation invariant functions can be simulated in the random order model in a single pass. This addresses a question raised by Feigenbaum *et al* regarding the relationship between property testing and stream algorithms. Further, we give a polylog-space PTAS for estimating the entropy of a one pass random order stream. This bound cannot be achieved in the combined oracle (generalized property testing) model.

1 Introduction

There are many settings where the natural unit of data, rather than being a point in a high dimensional vector space, is a distribution defined on n items. Examples include soft clustering [33], where the membership of a point in a cluster is described by a distribution, and anomaly detection [27], where the distance between two empirical distributions is used to detect anomalies. Typically, such settings involve large data sets, and so a natural requirement is that algorithms use small amounts of resources (space or time.)

In this paper, we examine sublinear algorithms for estimating properties of distributions. On the one hand we study the complexity of estimating information theoretic distances and measures on distributions, e.g., entropy, Jensen-Shannon divergence, Hellinger and Triangular distances, to name a few, and on the other, we explore the connections between various models in sublinear algorithms, e.g., property testing models, and data streams. We discuss both of these aspects below. We will not be able to review the extensive literature on either of these topics; however several good surveys exist, including those by Ron [32], Babcock *et al* [4] and Muthukrishnan [31].

1.1 Problems

When dealing with distributions, distances arising from information-theoretic considerations are often more natural than distances based on ℓ_p norms. In the first half of the paper we focus on the Ali-Silvey distances or f-divergences, discovered independently by Csiszár [18], and Ali and Silvey [1]. The class of f-divergences include many commonly used information theoretic distances, e.g., the (asymmetric) Kullback-Liebler (KL) divergence[1] and its symmetrization, the Jensen-Shannon (JS) divergence, Matsusita's Divergence or the squared Hellinger distance, the (asymmetric) χ^2 distance and its symmetrization, the Triangle distance. Every convex function f gives rise to an f-divergence $D_f(q,p) = \sum_{x\in\Omega} p(x) f\left(q(x)/p(x)\right)$ if $f(1) = 0$ and f is strictly convex at 1.[2]

Results of Csiszár [18], Liese and Vajda [28], Amari

*Department of Computer Information Sciences, University of Pennsylvania, 3330 Walnut St, Philadelphia, 19104. **Email:** {sudipto,andrewm}@cis.upenn.edu. This research was supported in part by the Alfred P. Sloan Research Fellowship, NSF Award CCF-0430376 and NSF ITR 0205456.

†AT&T Research Labs, 180 Park Ave, Florham Park, NJ 07928, **Email:** suresh@research.att.com

[1] Many of the measures we consider in this paper are not metrics – and several authors use constant multiples of the definitions in this paper. Traditionally, the term 'divergence' has been used to distinguish such measures from distances and metrics. We will use the terms 'distance' and 'divergence' interchangeably; a distance is not a metric unless explicitly mentioned.

[2] We can easily verify that $f(u) = u\ln u$ gives us the KL divergence; $f(u) = (\ln(2/(1+u)) + u\ln(2u/(1+u)))$ gives us the Jensen-Shannon (JS) divergence. The asymmetric Pearson's χ^2 distance is realized with $f(u) = (u-1)^2$ and is symmetrized to the Triangle distance with $f(u) = (u-1)^2/(u+1)$. Matsusita's Divergence or the (squared) Hellinger distance has $f(u) = (\sqrt{u}-1)^2$. The ℓ_1 or variational distance is realized with $f(u) = |u-1|$.

[3] and many others show that f-divergences are the unique class of distances on distributions that arise from a fairly simple set of axioms, e.g., permutation invariance, non-decreasing projections, certain direct sum theorems etc., in much the same way that ℓ_2 is a natural measure for points in R^n. Moreover, all of these distances are related to each other (via the Fisher information matrix) [13] in a way that other plausible measures (most notably ℓ_2) are not. In addition, the log-likelihood ratio $\ln \frac{q(x)}{p(x)}$ is a crucial parameter in Neyman-Pearson style hypothesis testing [17], and distances based on this (like the KL-distance and the JS-distance) appear as exponents of error probabilities for optimal classifiers. Recently, these distance measures have been used in more algorithmic contexts, as natural distances for clustering distributional data [33, 20, 5]. Batu *et al* [12] gave algorithms for testing closeness of distributions for the ℓ_1 and ℓ_2 distances, and raised the question of testing closeness of distributions under the JS-divergence. They state that they suspect that this is "the most powerful" notion of closeness.

In this paper we provide optimal (up to constants) algorithms for testing f-divergences of distributions. We consider the problem of estimating the entropy H of a distribution, providing optimal (up to constants) upper bounds for testing entropy. This improves the previous result of Batu *et al* [11] by a factor $\frac{\log n}{H(p)}$. Entropy is naturally related to the JS-divergence since $JS(p,q) = \ln 2(2H((p+q)/2) - H(p) - H(q))$ where $(p+q)/2$ is the average of the two distributions.

Switching from sublinear time to sublinear space, we then focus on the streaming model and derive several (regular) streaming algorithms that give a three way tradeoff between space, approximation, and number of passes. We note that these algorithms naturally imply (weaker) tradeoffs in JS distance and omit further discussion. We then develop a polylogarithmic space PTAS for estimating entropy in the random order stream model, which assumes that the input is a random permutation of some fixed multiset.

1.2 Models As it turns out, sublinear algorithms for testing distributions reveal interesting structure about the relationship between property testing and stream algorithms. Feigenbaum *et al* [21] considered the problem of property testing in a data stream model. They showed that there exist functions (e.g., SORTED-SUPERSET, a variant of permutation, [21]) that are easy in the property testing model but hard to test in streams. This was surprising since many sampling based techniques can be extended to data streams. For example, Bar-Yossef *et al* [10] showed that non-adaptive sampling can be easily simulated in an aggregate (all occurrences of item i are grouped together) streaming model with a small blowup in space. The aggregation assumption can be removed with an extra pass.

We show that in fact these (variants of permutations) are the only hard functions. Specifically, we show that any property testing algorithm for a permutation invariant (also known as symmetric) function in the *combined* oracle model can be simulated by a single pass data stream algorithm that assumes a random permutation of the input. The random permutation assumption can be removed using an extra pass to give a two-pass simulation in the regular streaming model. The simulation builds upon the reductions used by Bar-Yossef *et al* [8, 6, 7] in deriving strong lower bounds for sampling. However we use the reductions for upper bounds.

In a natural sense, if we exclude permutation dependent functions, stream testing in the random permutation model subsumes combined oracle property testing, and it is the testing of entropy that reveals this difference between the models.

2 Definitions

DEFINITION 2.1. ([18]) *Let p and q be two discrete probability distributions defined on base $[n]$. The* f-divergence *between p and q is defined as $D_f(p,q) = \sum p_i f(q_i/p_i)$ for some function f (convex, $f(1) = 0$.) Many commonly used distance measures are f-divergences, including the ℓ_1 distance, the* Hellinger distance[3] *$Hellinger(p,q) = \sum_i (\sqrt{p_i} - \sqrt{q_i})^2$, the* Jensen-Shannon distance *$JS(p,q) = \sum_i p_i \ln \frac{2p_i}{p_i+q_i} + q_i \ln \frac{2q_i}{p_i+q_i}$ and the* Triangle distance *$\Delta(p,q) = \sum_i \frac{(p_i-q_i)^2}{p_i+q_i}$.*

DEFINITION 2.2. *The entropy of a distribution is defined as $H(p) = \sum_i p_i \log \frac{1}{p_i}$. (All logs are base 2.)*

Throughout we will make use of the following form of the Chernoff/Hoeffding bound.

LEMMA 2.1. (HOEFFDING '63) *Let $\{X_t\}_{1 \le t \le m}$ be independently distributed random variables with (continuous) range $[0,u]$. Let $X = \sum_{1 \le t \le m} X_t$. Then for $\gamma > 0$, $\mathbb{P}(|X - \mathbb{E}(X)| \ge \gamma \mathbb{E}(X)) \le 2\exp(\frac{-\gamma^2 \mathbb{E}(X)}{3u})$.*

3 Property Testing

3.1 Oracle Models for Property Testing of Distributions Two main oracle models have been used in the property testing literature for testing properties of distributions. These are the *generative*

[3] Note that the Hellinger distance is sometimes defined as the square-root of the above quantity.

and *evaluative* models introduced by Kearns *et al* [26]. The black-box or generative model of a distribution permits only one operation: taking a sample from the distribution. In other words, given a distribution $p = \{p_1, \ldots p_n\}$, `sample`(p) returns i with probability p_i. In the evaluative model, a `probe` operation is permitted. `probe`(p, i) returns p_i. A natural third model, the *combined* model was introduced by Batu *et al* [11]. In this model both the `sample` and `probe` operations are permissible. In all three models, the complexity of an algorithm is measured by the number of operations.

3.2 Testing Jensen-Shannon, Hellinger and Triangle Divergences (Generative Oracle) In this section we consider property testing in the generative model for various information theoretic distances. We will present the results for the Triangle Divergence Δ. However, the Jensen-Shannon and Hellinger divergences are constant factor related to the Triangle divergence as follows:

$$\begin{aligned}(3.1)\quad \text{Hellinger}(p,q)/2 \leq \Delta(p,q)/2 \leq JS(p,q) \\ \leq \ln(2)\ \Delta(p,q) \leq 2\ln(2)\ \text{Hellinger}(p,q)\end{aligned}$$

(Parts of Eqn. 3.1 are proved in [34].) Therefore the results presented here naturally imply analogous results for them as well. Our algorithm is similar to that in [12], and is presented in Figure 1. It relies on an ℓ_2 tester given in [12]. Central to the analysis are the following inequalities.

$$(3.2)\quad \frac{\ell_2^2(p,q)}{\ell_\infty(p)+\ell_\infty(q)} \leq \Delta(p,q) \leq \ell_1(p,q) \leq \sqrt{n}\ell_2(p,q)$$

LEMMA 3.1. (ℓ_2 TESTING [12]) *There exists an algorithm that given distributions p, q draws $s = O(\log \frac{1}{\delta}(b^2+\epsilon^2\sqrt{b})/\epsilon^4)$ samples (where $b = \max_i(p_i, q_i)$) and if $\ell_2(p,q) \leq \epsilon/2$, the algorithms passes with probability at least $1-\delta$, but if $\ell_2(p,q) \geq \epsilon$ the algorithm passes with probability less than δ.*

The proofs of the following lemmas can be found in the full version [24].

LEMMA 3.2. *We say an estimate is* heavy *if it is greater than $1/n^\alpha$. Then, with $m = O(\log \frac{1}{\delta}\frac{n^\alpha \log n}{\gamma^2})$ samples, with probability $1-\delta/2$, for any heavy estimate $\tilde{p}_i$, $\tilde{p}_i$ is at most $p_i\gamma/100$ from p_i. Furthermore, if at least one of $\tilde{p}_i$ or $\tilde{q}_i$ is heavy then we can estimate $\frac{(p_i-q_i)^2}{p_i+q_i}$ up to $\pm\gamma\frac{\max\{p_i,q_i\}}{10}$.*

Algorithm Δ-*Test*

1. Draw m samples from p and from q
2. $n_i^p = \#$ times element i appears and $\tilde{p}_i = n_i^p/m$
3. $n_i^q = \#$ times element i appears and $\tilde{q}_i = n_i^q/m$
4. Let $S = \{i : \max\{n_i^p, n_i^q\} \geq mn^{-\alpha}\}$
5. **return** "Fail" if $\sum_{i\in S}\frac{(\tilde{p}_i-\tilde{q}_i)^2}{\tilde{p}_i+\tilde{q}_i} > \epsilon/10$
6. Define p' (and q' analogously) as follows:
7. $i \leftarrow$ `sample`(p)
8. if $i \notin S$, output i else output j uniformly chosen from $[n]$
9. **return** ℓ_2-Tester ran on p', q' and $\frac{\epsilon}{\sqrt{n}}$

Figure 1: Δ-Testing in the Generative Model

THEOREM 3.1. (Δ TESTING) *For two distributions p and q, there exists an algorithm drawing*

$$s = O(\log\left(\frac{1}{\delta}\right)\max\{\frac{n^\alpha \log n}{\epsilon^2}, (n^{-2\alpha+2}+\epsilon^2 n^{1-\alpha/2})/\epsilon^4\})$$

samples such that if $\Delta(p,q) \leq \epsilon^2/n^{1-\alpha}$, the algorithm passes with probability at least $1-\delta$, but if $\Delta(p,q) \geq \epsilon$ the algorithm passes with probability less than δ.

Observe that setting $\alpha = 2/3$ yields an algorithm with sample complexity $\tilde{O}(n^{2/3}/\epsilon^4)$. For distributions such that either $p_i = q_i$ or one of p_i, q_i is 0, $\Delta(p,q) = \ell_1(p,q)$. The same holds true for the Jensen-Shannon and Hellinger divergences. Batu *et al* [12] discuss lower bounds for ℓ_1 distance property testing.

3.3 Testing all Symmetric Bounded f-Divergences (Combined Oracle) In this section we consider property testing in the combined oracle model for all symmetric bounded f-divergences. Recall that a convex function f defines a divergence $D_f(p,q) = \sum_i p_i f(\frac{q_i}{p_i})$; this encodes the permutation invariance. We are interested in symmetric (over p, q) divergences, i.e., $D_f(p,q) = D_f(q,p)$. We define a divergence to be *bounded*[4] if $\max\{f(u), uf(\frac{1}{u})\} = \tau < \infty$. JS, Hellinger, Δ and ℓ_1 all are bounded and satisfy $\tau \leq 2$. We show an interesting decomposition property of symmetric f divergences. Define a *conjugate* $f^*(u) = uf(\frac{1}{u})$. It can be verified that JS, Hellinger, Δ and ℓ_1 are self conjugates, i.e., $f(u) = f^*(u) = uf(\frac{1}{u})$. One useful characterization of symmetric f-divergences is the following:

[4]Since f is typically monotone, i.e., if the likelihood ratio is closer to 1, then the distance decreases, and so boundedness reduces to $\lim_{u\to 0} f(u)$ and $\lim_{u\to\infty}\frac{f(u)}{u}$ exist and is at most $\tau < \infty$. Note that from continuity $\lim_{u\to 1} f(u) = f(1) = 0$.

LEMMA 3.3. ([28]) *$D_f(p,q) = D_f(q,p)$ iff $f^*(u) = f(u) - c(u-1)$ for some constant c.*

Therefore using f is the same as using $(f(u) + f^*(u) - 2c(u-1))/2$, but since $\sum_i p_i c(\frac{q_i}{p_i} - 1) = c(\sum_i q_i - \sum_i p_i) = c(1-1) = 0$, we may as well use $g(u) = (f(u) + f^*(u))/2$. We now claim the following:

LEMMA 3.4. *For any symmetric divergence f,*

$$D_f(p,q) = \sum_{i:p_i>q_i} p_i g(x) + \sum_{i:q_i>p_i} q_i g(1/x)$$

where $g(x) = \frac{1}{2}(f(x) + xf(\frac{1}{x}))$ and $x = \frac{q_i}{p_i}$. Further, if f is bounded then $g(x) \leq \tau$ if $x \in [0,1]$.

Although the above appears simple, it actually allows us to break the divergence into small, positive components. This allows us to use sharp concentration bounds.

Algorithm *Combined Oracle Distance Testing*
1. $E \leftarrow 0$
2. **for** $t = 1$ to m:
3. **do** $i \leftarrow$ `sample`(p) and $x = \frac{\texttt{probe}(q,i)}{\texttt{probe}(p,i)}$
4. If $x > 1$ then $a \leftarrow g(x)$ else $a \leftarrow 0$
5. $j \leftarrow$ `sample`(q) and $x = \frac{\texttt{probe}(q,j)}{\texttt{probe}(p,j)}$
6. If $x < 1$ then $b \leftarrow g(1/x)$ else $b \leftarrow 0$
7. $E \leftarrow (a+b)/2\tau + E$
8. **return** $2\tau E/m$

Figure 2: Combined Oracle Distance Testing

THEOREM 3.2. *For a bounded symmetric f-divergence D_f in the combined oracle model we can estimate $D_f(p,q)$ up to a factor $(1+\epsilon)$ in $O(\tau/(\epsilon^2 D_f(p,q)))$ time. ($\tau = O(1)$ for ℓ_1, Hellinger, Δ, and JS.)*

Proof. Consider the value $\frac{a+b}{2u}$ added to E in each iteration. This is a random variable with range $[0,1]$ and mean $\frac{D_f(p,q)}{2\tau}$. Hence by Lemma 2.1,

$$\mathbb{P}\left(|E - m\frac{D_f(p,q)}{2\tau}| < \epsilon m \frac{D_f(p,q)}{2\tau}\right) \leq 2e^{-\epsilon^2 D_f(p,q) m/6\tau}$$

Therefore with $O(\tau/(\epsilon^2 D_f(p,q)))$ samples/probes the probability that we do not estimate $D_f(p,q)$ as required can be made arbitrarily small.

Note that although ℓ_2 is not an f-divergence, by setting $a = p_i(1 - q_i/p_i)^2$ and $b = q_i(p_i/q_i - 1)^2$ in the above we can estimate $\ell_2^2(p,q)$ in $O(1/(\epsilon^2 \ell_2^2(p,q)))$ time. It is worth mentioning that the above results can be rephrased as an $O(1/\epsilon)$ algorithm if we are interested in distinguishing between the cases where the distance is greater than ϵ or less than $\epsilon/2$.

We now prove a corresponding lower bound that shows that our algorithm is tight. Note that while it is relatively simple to see that there exists two distributions that are indistinguishable with less than $o(1/\ell_1)$ oracle calls, it requires some work to also show a lower bound with a dependence on ϵ. Further note that the proof below also gives analogous results for JS, Hellinger and Δ. (This follows from the remarks made at the end of the previous section.)

THEOREM 3.3. (ℓ_1 LOWER BOUND) *Any approximation up to a $1+\epsilon$ factor of the ℓ_1 distance between two distributions in the combined oracle model requires $\Omega(\frac{1}{\epsilon^2 \ell_1})$ time.*

Proof. Let p and q^r be the distributions on $[n]$ described by the following two probability vectors:

$$p = (1 - 3a/2, \overbrace{3a\epsilon/2k, \ldots, 3a\epsilon/2k}^{k/\epsilon}, 0, \ldots, 0)$$

$$q^r = (1 - 3a/2, \overbrace{0, \ldots 0}^{r}, \overbrace{3a\epsilon/2k, \ldots, 3a\epsilon/2k}^{k/\epsilon}, 0, \ldots, 0)$$

Then $\ell_1(p, q^{k/3\epsilon}) = a$ and $\ell_1(p, q^{k/3\epsilon+k}) = a(1+3\epsilon)$. Hence to $1+\epsilon$ approximate the distance between p and q^r we need to distinguish between the cases when $r = k/3\epsilon (=: r_1)$ and $r = k/3\epsilon + k (=: r_2)$. Consider the distributions p' and $q^{r\prime}$ formed by arbitrarily permuting the base sets of the p and q^r. Note that the ℓ_1 distance remains the same. We will show that, without knowledge of the permutation, it is impossible to estimate this distance with $o(1/(\epsilon^2 a))$ oracle calls. We reason this by first disregarding the value of any "blind probes", ie. a probe `probe`(p', i) or `probe`(q', i) for any i that has not been returned as a sample. This is the case because, by choosing $n \gg k/(a\epsilon^2)$ we ensure that, with arbitrarily high probability, for any $o(1/(\epsilon^2 a))$ set of i's chosen from any $n - o(1/(a\epsilon^2))$ sized subset of $[n]$, ${p_i}' = {q_i}^{r\prime} = 0$. This is the case for both r_1 and r_2. Let $I = \{i : p_i \text{ or } q_i^r = 3a\epsilon/2k\}$ and $I_1 = \{i \in I : p_i \neq q_i^r\}$. Therefore determining whether $r = r_1$ or r_2 is equivalent to determining whether $|I_1|/|I| = 1/2$ or $1/2 + \frac{9\epsilon}{8+6\epsilon}$. We may assume that every time an algorithm sees i returned by `sample`(p) or `sample`(q), it learns the exact values of p_i and q_i for free. Furthermore, by making k large ($k = \omega(1/\epsilon^3)$ suffices) we can ensure that no two `sample` oracle calls will ever return the same $i \in I$ (with high probability.) Hence distinguishing between $|I_1|/|I| = 1/2$ and $1/2 + \frac{9\epsilon}{8+6\epsilon}$ is analogous to distinguishing between a fair coin and a $\frac{9\epsilon}{8+6\epsilon} = \Theta(\epsilon)$ biased coin. It is well known that the latter requires $\Omega(1/\epsilon^2)$

samples. Unfortunately only $O(1/a)$ samples return an $i \in I$ since with probability $1-3a/2$ we output an $i \notin I$ when sampling either p or q. The bound follows.

3.4 Testing Entropy (Combined Oracle) In this subsection we show that a simple algorithm achieves the optimal bounds for estimating the entropy in the combined oracle model of property testing. Note that this algorithm improves upon the previous upper bound of Batu *et al* [11] by a factor of $\log n/H$ where H is the entropy of the distribution. The authors of [11] showed that their algorithms were tight for $H = \Omega(\log n)$; we show that the upper and lower bounds match for arbitrary H. The algorithm is presented in Figure 3. It is structurally similar to the algorithm given in [11] but uses a cutoff that will allow for a much tighter analysis via Chernoff bounds.

Algorithm *Combined Oracle Entropy Testing*
1. $E \leftarrow 0$
2. **for** $t = 1$ to m:
3. **do** $i \leftarrow \texttt{sample}(p)$
4. $p_i \leftarrow \texttt{probe}(p, i)$
5. if $p_i \geq \frac{1}{n^3}$ then $a \leftarrow \frac{\log 1/p_i}{3 \log n}$ else $a \leftarrow 0$
6. $E \leftarrow a + E$
7. **return** $3E \log n/m$

Figure 3: Combined Oracle Entropy Testing

The next lemma estimates the contribution of the unseen elements and that leads to the main theorem about estimating entropy in the combined oracle model.

LEMMA 3.5. *Consider any set subset S of $[n]$, then $\sum_{i \in S} p_i \log 1/p_i \leq \sum_{i \in S} p_i \log \frac{|S|}{\sum_{i \in S} p_i}$. In particular, if $\sum_{i \in S} p_i \leq \frac{\epsilon H(p)}{\log n - \log \epsilon H(p)}$ then $\sum_{i \in S} p_i \log 1/p_i \leq \epsilon H(p)$.*

THEOREM 3.4. *In the combined oracle model we can $1+\epsilon$ estimate the entropy $H(p)$ of a distribution using $O(\frac{\log n}{\epsilon^2 H(p)})$ where H is the entropy of the distribution.*

Proof. We restrict our attention to the case when $H(p) > 1/n$ and $\epsilon > 1/\sqrt{n}$ since otherwise we can trivially find the entropy exactly in $O(1/\epsilon^2 H(p))$ time by simply probing each of the n p_i's. Consider the value a added to E in each iteration. This is a random variable with range $[0, 1]$ since $p_i \geq 1/n^3$ guarantees that $\frac{\log 1/p_i}{3 \log n} \leq 1$. Now, the combined mass of all p_i such that $p_i < 1/n^3$ is at most $1/n^2$. Hence since $1/n^2 \leq \frac{\epsilon/2 H(p)}{\log n - \log \epsilon/2 H(p)}$ by Lemma 3.5 the maximum contribution to the entropy from such i is at most $\epsilon H(p)$. Hence the expected value of a is between $(1 - \epsilon/2)H(p)/(3 \log n)$ and $H(p)/(3 \log n)$ and therefore, if we can $1+\epsilon/2$ approximate $\mathbb{E}(a)$ then we are done. We use the probability concentration Lemma 2.1 to get that $\mathbb{P}(|E - m\mathbb{E}(a)| < (\epsilon/2) m\mathbb{E}(a)) \leq 2e^{-(\epsilon/2)^2 m\mathbb{E}(a)/3}$. Therefore with $O(1/(\epsilon^2 \mathbb{E}(a))) = O(\log n/\epsilon^2 H(p))$ samples/probes the probability that we do not $1+\epsilon/2$ approximate $\mathbb{E}(a)$ can be made arbitrarily small.

4 Data Streams

4.1 The Data Streams Model The data stream model characterizes small space algorithms that can access the read-only input in order. The algorithm makes passes over the input; any item not explicitly stored is inaccessible to the algorithm in the same pass. In many cases the number of passes is limited to one. The crucial aspect of a (regular) data stream algorithm is that the algorithm is required to produce a correct output for an arbitrary permutation of the input stream.

As mentioned in the introduction, *a random order stream algorithm* is a data stream algorithm that reads a *randomly permuted* input from its read-only input tape. Alternate definitions are possible, but this definition dates back to Munro and Paterson [30] and we will restrict ourselves to this definition. (It also appeared in [19].) All other features are the same as a general stream algorithm. As usual, the complexity of the algorithm is measured primarily in terms of the amount of space used on the work tape (for which the algorithm has random read-write access.)

Modeling stream distributions: There are two ways in which a data stream can be considered to define a probability distribution p. These are the *update data stream model* and the *aggregate data stream model*. Firstly we discuss the *update data stream model*. We are given a base domain $[1, \ldots, n]$ over integers and a function $f_p()$ is specified as $\langle p, i, + \rangle$ which corresponds to $f_p(i) \leftarrow f_p(i) + 1$. This is the model used by Alon *et al* in [2]. The model naturally captures $f_p(i) \leftarrow f_p(i) + \Delta_i$, however we do not consider $f_p(i) \leftarrow f_p(i) - 1$ (deletions) since the negative term does not correspond to any operation over distributions.

An alternate model is the *aggregate model* where the input is $\langle p, i, f_p(i) \rangle$. This is the model used by Feigenbaum *et al* [22] for ℓ_1 differences. In this model, computing the entropy is trivial and the Hellinger distance reduces to computing the ℓ_2^2 norm. Note that this implies that we can compute the JS Divergence and the Δ divergence up to a constant factor as well. Obtaining a PTAS for them remains an interesting open question even in this simpler model. However, the aggregate model is restrictive for distributions, since the aggregation loses the "distribution" aspect.

We will focus on the update model, and, as we argued above, will only consider insertions. We will assume that the length of the stream, m, is polynomial in n. In particular we will assume we know an upper bound on the length of the stream, $m^{\max}(n)$.

Random Streams and two distributions: When we are computing a function of two distributions p and q we also have a function $f_q()$ specified by data items $\langle q, i, + \rangle$. Note that, for a random stream algorithm, we consider the random permutation to be over $\langle q, i, + \rangle$ and $\langle p, i, + \rangle$ together. We will assume that $\sum_i f_p(i)$ and $\sum_i f_q(i)$ are within a constant factor of each other.

4.2 Simulating the Combined Oracle Models In the next section we will discuss how algorithms that make (combined) oracle calls may be simulated in the various streaming models. In particular this leads to the following theorem.

THEOREM 4.1. *In two passes of a regular stream, there exist algorithms that,*

1. $(1+\epsilon)-$*approximate the entropy H using $\tilde{O}(\frac{1}{\epsilon^2 H})$ space.*
2. $(1+\epsilon)-$*approximate a τ bounded symmetric f-Divergence D_f using $\tilde{O}(\frac{\tau}{\epsilon^2 D_f})$ space.*

If the stream is randomly ordered then one pass suffices in each case.

Unfortunately, it is sometimes unrealistic to assume more than a single pass over the data. Hence we now concentrate on single pass algorithms. In what follows we present a single pass, polylog space, asymptotic constant factor approximation in a single pass. We then briefly discuss a single pass algorithm that uses significantly more space but achieves a multiplicative approximation even when the entropy is very small. Finally, we present one pass algorithm for the random streaming model whose memory needs are not in terms of $1/H$.

4.3 An Asymptotic Approximation Scheme for Estimating Entropy in Regular Streams To construct our algorithm we will use algorithms for approximating the F_0. There has been a long history of papers for computing the frequency moments of streams. We focus our attention to the best known (ϵ, δ) approximation algorithm of Bar-Yossef *et al* [9], where F_0 is approximated up to a factor $(1+\epsilon)$ with probability $1-\delta$. Their result shows that the (ϵ, δ)-approximation can be performed in $O((\frac{1}{\epsilon^2}\log\log n + \log n)\log\frac{1}{\delta})$ space. We will only focus on the fact that the space bounds are polylogarithmic. The basic intuition of our algorithm is similar to the sublinear time minimum spanning tree algorithm of Chazelle *et al* [15] and the streaming geometry algorithms by Indyk [25]. The idea is to count objects at various resolutions.

Our algorithm works by randomly generating conceptual sub-streams from the data stream. Each sub-stream has a associated level j and we will perform the random generation of a sub-stream of level j in such a way that we only expect elements i with $p_i \geq 2^{-j}$ to appear in the stream. There will be a family of sub-streams corresponding to each guess of the stream length, ie. the substreams $S_{\tilde{m},j}$ are indexed by our guess of the stream length, $\tilde{m}$, and the level of the stream j. We will feed each sub-stream into an algorithm for estimating the number of distinct elements in the sub-stream. Summing up the these estimates (appropriately scaled) will give our estimate. The net result will be an asymptotic approximation scheme of factor $\frac{e}{e-1}$, i.e., for H sufficiently large (but constant)[5]. The algorithm is presented in Figure 4.

Algorithm *Entropy-Estimation*
1. Guess length $\tilde{m} = 1, (1+\epsilon), \ldots (1+\epsilon)^{\log_{1+\epsilon} m^{\max}}$
2. $m \leftarrow 0$
3. **for** each stream item $\langle i, + \rangle$
4. **do** $m \leftarrow m+1$
5. **for** each guess $\tilde{m}$ of m
6. **do for** $j = 1, \ldots \log\frac{n}{\epsilon}$
7. **do** w.p. $\frac{2^j}{\tilde{m}}$, $S_{\tilde{m},j} \leftarrow i$
8. update $F_0(S_{\tilde{m},j})$
9. For guess $\tilde{m}$ such that $(1+\epsilon)\tilde{m} > m \geq \tilde{m}$, for each j let $f_j = F_0(S_{\tilde{m},j})$
10. **return** $\sum_j f_j/2^j$

Figure 4: The F_0 Algorithm for Estimating Entropy in a Data Stream

The centerpiece of the algorithm is the following lemma. Let χ_{ij} be the event that item i showed up in level j. Note that χ_{ij} are independent.

LEMMA 4.1. *For entropy H,*

$$(1-\frac{1}{e})(H-1) \leq \mathbf{E}\left[\sum_j \frac{f_j}{2^j}\right] \leq (1+\epsilon)H + 2 \ .$$

Proof. First we bound $Pr[\chi_{ij}]$ as follows:

$$Pr[\chi_{ij}] \leq \frac{(1+\epsilon)\cdot 2^j}{m} m p_i \leq (1+\epsilon) p_i 2^j$$

[5] This is in the same spirit as the approximation algorithms for bin-packing, checking for packing when 2 bins is NP-HARD, but we have a PTAS as the number of bins is large.

and

$$Pr[\chi_{ij}] \geq 1-\left(1-\frac{2^j}{m}\right)^{mp_i} \geq 1-e^{-2^j p_i} \geq (1-\frac{1}{e})2^j p_i \ .$$

Consider $\mathbf{E}\left[\sum_j \frac{f_j}{2^j}\right]$

$$\mathbf{E}\left[\sum_j \frac{f_j}{2^j}\right] = \sum_j \mathbf{E}\left[\frac{f_j}{2^j}\right] = \sum_j \frac{1}{2^j}\sum_i Pr[\chi_{ij}]$$

$$\leq \sum_i \left\{ \sum_{j=1}^{\lfloor \log \frac{1}{p_i} \rfloor} (1+\epsilon)p_i 2^j \frac{1}{2^j} + \sum_{j=\lfloor \log \frac{1}{p_i} \rfloor + 1}^{\infty} \frac{1}{2^j} \right\}$$

$$\leq \sum_i \left\{ (1+\epsilon)p_i \log \frac{1}{p_i} + 2p_i \right\}$$

$$\leq (1+\epsilon)H + 2 \ .$$

Similarly,

$$\mathbf{E}\left[\sum_j \frac{f_j}{2^j}\right] = \sum_{i,j} \frac{1}{2^j} Pr[\chi_{ij}]$$

$$\geq \sum_i \sum_{j=1}^{\infty} \left[1-\left(1-\frac{2^j}{m}\right)^{mp_i}\right] \frac{1}{2^j}$$

$$\geq \sum_i \sum_{j=1}^{\lfloor \log \frac{1}{p_i} \rfloor} \left[1-\left(1-\frac{2^j}{m}\right)^{mp_i}\right] \frac{1}{2^j}$$

$$\geq \sum_i \sum_{j=1}^{\lfloor \log \frac{1}{p_i} \rfloor} (1-\frac{1}{e})\frac{2^j p_i}{2^j}$$

$$\geq (1-\frac{1}{e})\sum_i p_i(\log \frac{1}{p_i} - 1) = (1-\frac{1}{e})(H-1)$$

which proves the lemma.

We assume that the length of the stream is $m \gg n/\epsilon$ but that m is polynomial in n. Now observe that $\sum_j f_j/2^j$ is a sum of many (bounded) Bernoulli variables. We can apply Chernoff and show with probability $O(\exp(-\epsilon_c^2 H))$ the sum $\sum_j f_j/2^j \in \left[(1-\frac{1}{e})(H-1)(1-\epsilon_c), (\frac{1}{t}(1+\epsilon)^2 H + 2)(1+\epsilon_c)\right]$. We need to repeat the above for $O(\log n/\epsilon_c H)$ times for high probability. Now, we lose a further $1\pm\epsilon_0$ factor in the F_0 estimation. Overall, we maintain $O(\log n/\epsilon)$ different F_0 estimation structure, each of which takes space $O((1/\epsilon_0^2)\log\log n + \log n)\log n$ and succeed with high probability. Finally there are $\log n/\epsilon$ guesses for $\tilde{m}$. The overall space bound is $O((\log^3 n/(\epsilon\epsilon_c H))(\frac{1}{\epsilon_0^2}\log\log n + \log n))$.

THEOREM 4.2. *We have a polylogarithmic space asymptotic $\frac{e}{e-1}+\epsilon$ approximation for the entropy.*

A natural question is if the above analysis is tight. It is possible to slightly improve the bounds in Lemma 4.1 but the problem is that there will always be constant shift between the entropy and our estimate. There is a natural bias in the estimation entropy and this particular method alone is unlikely to yield better results.

4.4 A True PTAS (at a Price) for Estimating Entropy in Regular Streams Since the algorithm in the previous section only succeeds in finding a good multiplicative approximation if entropy is large, it is natural to ask if we can find a true PTAS? The answer is yes, but it comes with a price. The space bound for $\approx \frac{1}{\alpha}$ approximation increases by a factor approximately n^α/H. The algorithm divides the elements into "large" and "small" classes. It proceeds in two steps (i) Finds the elements with large $f(i)$ and estimate them and (ii) Uses a worst case bound for small $f(i)$. The first step is achieved by a careful sampling technique reminiscent of the online facility location algorithm of Meyerson [29] in the context of stream clustering. There are also some similarities with the count/count-min sketches of [14, 16]. Our algorithm will "track" a few items, i.e., maintain explicit counters for them. Let H be the true entropy and p_i the true probability of i. Assume $m > n \geq 3$. The algorithm is presented in Figure 5. The following theorem states the performance and precise resource requirements of the algorithm. The proof appears in the full version [24].

THEOREM 4.3. *We can estimate the entropy H up to a factor $\frac{1}{\alpha}(1+\epsilon')$ using space $O(\frac{n^\alpha \log n}{f(\epsilon',H)})$ where $f(\epsilon',H)$ is the value of ϵ satisfying, $\epsilon' = \frac{\alpha(\epsilon\log(n/\epsilon)+n^{-\alpha})}{H} + \frac{\log 1/\epsilon}{\log n}$.*

4.5 A PTAS in the Random Streams Model Emulating the combined oracle algorithm is only inefficient when the entropy is very small. But when the entropy is small, it is easy to see that there must be one element with probability mass ≈ 1. The high level idea of our algorithm is to keep track of the exact counts of the elements of a certain set A; and establish that the projection of the distribution (rescaled to 1) to the complement of A has large entropy. On this projection we run the simulation of the combined oracle algorithm. However, we have several problems; (a) when we discover that the entropy is large we have already seen a few elements from the projection – we have a dependence, (b) the projected distribution may again have one element with mass ≈ 1, and (c) we will not know

Algorithm *Entropy-Estimation*

1. Guess length $\tilde{m} = 1, 2, \ldots 2^{\log_2 m^{\max}}$
2. $m \leftarrow 0$
3. **for** each stream item $\langle i, + \rangle$
4. **do** $m \leftarrow m + 1$
5. **for** each guess $\tilde{m}$, of m
6. **do if** i is being "tracked"
7. **then** $c_{\tilde{m}}(i) \leftarrow c_{\tilde{m}}(i) + 1$
8. **else** w.p. $\frac{\log n}{\epsilon n^{\alpha} m}$ start tracking i
9. Keep checking entropy is non-zero (at least two elements are seen)
10. Let $\tilde{m}$ be such that $2\tilde{m} \geq m \geq \tilde{m}$
11. **return** $\hat{H} = \hat{H}_{\hat{S}} + \sum_{i \in \bar{S}} h_i$ where $\bar{S}$ is the set of tracked elements with $c_{\tilde{m}}(i) \geq n^{\alpha}$, $\hat{S}$ be the complement of $\bar{S}$,

$$\hat{H}_{\hat{S}} = \hat{w} \log \frac{n}{\hat{w}}, \hat{w} = 1 - \sum_{i \in \bar{S}} \frac{c_{\tilde{m}}(i)}{m},$$

$$\text{and } h_i = \frac{c_{\tilde{m}}(i)}{m} \log \frac{m}{c_{\tilde{m}}(i)}$$

Figure 5: The Large-Small Algorithm for Estimating Entropy in a Data Stream

the exact probability mass of any set until the end of input.

PROPOSITION 4.1. *In the random stream model if we consider the first s elements which do not belong to A, we have a random sample (without replacement) from the distribution projected to $[n] - A$.*

Using Proposition 4.1, we can show that either estimates are sufficient for (b) and (c), or that the stream has few distinct elements, which we count exactly.[6] At the end of input we will know the exact probability mass of A and rescale/shift to get the contribution of the projection to the original distribution.

LEMMA 4.2. *We can eliminate the dependence between MS and the simulated property testing algorithm.*

Proof. We want to find $t = O(\epsilon^{-2} \log n)$ elements from the projection at random with replacement (H^{-1} is constant.) Let MS be the multi-set of items which allowed us to conclude that the entropy is large. We keep the subsequent counts of all the elements in MS. We also look at the first t elements of the projection (irrespective of membership in MS), denoted by `Prefix` and keep track of the new elements seen. At the end of the stream, we know the length m, and can now simulate the combined oracle algorithm using MS and `Prefix`.

The algorithm: At any point of time we are maintaining a set of items A. For each $i \in A$ we are counting the number of occurrences of i. We keep seeing elements in the stream until we accumulate a multi-set MS of size $c_1 \log n$ of elements that do not belong to A and appear earliest in the stream. (c_1 is some large constant.) We check if any $i' \in MS$ occurs at least $\frac{c_1}{2} \log n$ times in MS.

- If there is no such i' (projection has large entropy), we proceed to simulate the t queries with the correction due to M as described above (maintaining counts of the elements in A as well.)
- If there is any such i', let $A \leftarrow A \cup \text{distinct}(MS)$. We operate on the rest of the stream with this new A. Observe that the invariant of maintaining the count of elements in A can be maintained.

We also store the last $O(\epsilon^{-2} \log n)$ elements in the stream (also a multi-set) seen at all time.

Note that we can assume that the remaining stream is much larger than MS – we maintain the last $O(\epsilon^{-2} \log n)$ elements and would discover that the remaining stream is small within our alloted space. But in that event we would have an exact description of the distribution of the items in the stream. The next lemma follows from the assumption, Proposition 4.1 and the Chernoff bound.

LEMMA 4.3. *Assuming that the remaining stream is much larger than MS, if there is an item in the projection with mass $\geq \frac{1}{2}$, then it occurs at least $\frac{1}{4}$ fraction in MS with high probability polynomially close to 1.*

The theorem follows from the fact that either we take out probability mass quickly, or shift to the simulation phase.

THEOREM 4.4. *We can compute a $(1 + \epsilon)$-approximation for the entropy in a random stream model using $O((\log n + \epsilon^{-2}) \log n)$ space.*

Proof. Reconsider the algorithm above. If there is no i' that occurs frequently in MS, we are guaranteed (with high probability) that the entropy H of the projection satisfies $H \geq 1$ and the simulation is successful whp as well. If there is such an i' then whp we decrease the mass of the projection by at least a factor 1/4. This can repeat for at most $8 \log m = O(\log n)$ steps since after that the probability mass of the residual

[6] We can view the setting as a "robust distribution" as in Gilbert *et al* [23]

distribution is smaller than $1/m^2$. The contribution to $H(p)$ of the elements in the residual distribution is at most $\frac{2\log m+\log n}{m^2}$. But if there were more than one element in the original distribution then the minimum entropy of that distribution is $O((1/m)\log m)$, since the worst case occurs when $m-1$ elements are the same and 1 element is different (from concavity.) So we can ignore the contribution of these elements in the residual distribution (note that after the first projection we are guaranteed that there are two distinct elements in the stream.)

We can tighten the above by noticing that if the number of distinct elements in MS is $r+1$ then the entropy is $\Omega(r/\log n)$ and we can shift to the second phase. We omit the discussion.

5 Connecting Oracle and Streaming Models

We direct the reader to [6] for a detailed treatment of the relative computational power of the data stream, sketch (see [6] for a definition) and generative sampling models. Here we restrict ourselves to comparing the combined oracle model with the streaming model.

We say that a function $f(p) = f(p_1, p_2, \ldots p_n)$ is *symmetric* (or *permutation-invariant*), if f remains unchanged when its arguments are permuted. (Symmetry is a desirable and often-assumed property of functions on distributions, and is a special case of general invariance under coordinate reparametrizations [13].) We will show that we can always express an algorithm for the combined oracle model in a canonical form where the algorithm first samples and then probes the samples along with a few other elements. The idea would be to view the original algorithm, after the sampling stages and probing of the samples, as a randomized decision tree that we rewrite as an oblivious decision tree along the lines of Bar-Yossef *et al* [10, 6]. Then we could simulate this new decision tree in the random stream model. We start with the necessary definitions.

Definition 5.1. *A randomized decision tree that computes a function $f(x)$ is defined (as usual) as a decision tree having three types of nodes; a* query *node that asks for the value of an input parameter and maps the resulting value (and the history of all queries up to that point) to a choice of child node to visit, a random choice node, where the child node is chosen at random, and output nodes, where an answer expressed as a function of all queries thus far is returned.*

Definition 5.2. *An* oblivious decision tree *is one where the queries are made independent of the input, or the random choices in the algorithm. Formally, suppose we have a tree T with worst-case query complexity u. Then an* I-relabeling *of T by $I = \{i_1, \ldots i_u\}$ relabels all query nodes of depth j by the query to i_j, yielding the tree T^I. An oblivious decision tree is then a pair T, Δ_u, where T is a decision tree with worst-case complexity q and Δ_u is a distribution on $[n]^u$. A computation on an oblivious decision tree consists of two steps: (1) sample u elements I from Δ_q, (2) Relabel T to T^I and run it on input x.*

The first lemma shows how any combined oracle tester can be transformed (with only a slight blow-up in complexity) to one of a canonical form. This proof, and subsequent proofs, can be found in the full version [24].

Lemma 5.1. (Canonical Form Algorithm) *Let $\mathcal{A}$ be a combined oracle property testing algorithm that $1+\gamma$ estimates a symmetric function $f(p)$ using (worst-case) t oracle calls and probability of error δ. Then there exists a canonical algorithm $\mathcal{A}'$ that uses (worst case) $O(t)$ oracle calls with equal performance.*

We are now ready to prove the main structural result of this section. The central idea for simulating in two pass regular stream model is to sample in the first pass and then do exact counting in the second pass. For the random order stream result we are able to do both the sampling and exact counting in the same pass by using, roughly speaking, the prefix of the random order stream as a source for sample oracle queries.

Theorem 5.1. *Let p be the probability distribution described by an update data stream. Let $\mathcal{A}$ be a combined oracle property testing algorithm that makes at most t oracle calls to $1+\gamma$ estimate a symmetric function $f(p)$ with probability of error at most δ. Then there exist a single pass random stream algorithm and a two pass regular stream algorithm that use $O(t \log m \log n)$ space with equal performance.*

The proof can be generalized to the case when we are computing a function of two distributions $f(p, q)$, e.g., a distance between two distributions. In this case we consider f as a function over n tuples, ie. $f(p, q) = f((p_1, q_1), (p_2, q_2), \ldots (p_n, q_n))$. f is symmetric if it is invariant of permutations of the n-tuples. The only important caveat is that we need $\sum_i f_p(i) = \Theta(\sum_i f_q(i))$ such that, with high probability, there are t elements of the form $\langle p, i, + \rangle$ (for some i) and t elements of the form $\langle q, i, + \rangle$ (for some i) in the first $O(t)$ data items.

References

[1] S. M. Ali and S. D. Silvey, *A general class of coefficients of divergence of one distribution from another*, J. of Royal Statistical Society, Series B **28** (1966), 131–142.

[2] N. Alon, Y. Matias, and M. Szegedy, *The space complexity of approximating the frequency moments*, J. Comput. Syst. Sci. **58** (1999), no. 1, 137–147.

[3] S. Amari, *Differential-geometrical methods in statistics*, Springer-Verlag, New York, 1985.

[4] Brian Babcock, Shivnath Babu, Mayur Datar, Rajeev Motwani, and Jennifer Widom, *Models and issues in data stream systems*, Proc. of PODS (2002), 1–16.

[5] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh, *Clustering with bregman divergences*, JMLR (2005).

[6] Ziv Bar-Yossef, *The complexity of massive data set computations.*, Ph.D. thesis, University of California at Berkeley, 2002.

[7] ———, *Sampling lower bounds via information theory*, Electronic Colloquium on Computational Complexity (ECCC) (See also STOC 2003) **10** (2003), no. 037.

[8] Ziv Bar-Yossef, T. S. Jayram, Ravi Kumar, and D. Sivakumar, *An information statistics approach to data stream and communication complexity.*, FOCS, IEEE Computer Society, 2002, pp. 209–218.

[9] Ziv Bar-Yossef, T. S. Jayram, Ravi Kumar, D. Sivakumar, and Luca Trevisan, *Counting distinct elements in a data stream*, Proc. of RANDOM (2002), 1–10.

[10] Ziv Bar-Yossef, Ravi Kumar, and D. Sivakumar, *Sampling algorithms: lower bounds and applications*, Proc. of STOC (2001), 266–275.

[11] Tugkan Batu, Sanjoy Dasgupta, Ravi Kumar, and Ronitt Rubinfeld, *The complexity of approximating entropy.*, STOC, 2002, pp. 678–687.

[12] Tugkan Batu, Lance Fortnow, Ronitt Rubinfeld, Warren D. Smith, and Patrick White, *Testing that distributions are close.*, FOCS, 2000, pp. 259–269.

[13] N. N. Cencov, *Statistical decision rules and optimal inference*, Transl. Math. Monographs, Am. Math. Soc. (Providence) (1981).

[14] Moses Charikar, Kevin Chen, and Martin Farach-Colton, *Finding frequent items in data streams*, Theor. Comput. Sci. **312** (2004), no. 1, 3–15.

[15] Bernard Chazelle, Ronitt Rubinfeld, and Luca Trevisan, *Approximating the minimum spanning tree weight in sublinear time*, Proc. of ICALP (2001), 190–200.

[16] Graham Cormode and S. Muthukrishnan, *An improved data stream summary: The count-min sketch and its applications*, Proc. of LATIN (2004), 29–38.

[17] Thomas M. Cover and Joy A. Thomas, *Elements of information theory*, Wiley Series in Telecommunications, John Wiley & Sons, New York, NY, USA, 1991.

[18] I. Csiszar, *Why least squares and maximum entropy? an axiomatic approach to inference for linear inverse problems*, Ann. Statist. (1991), 2032–2056.

[19] Erik D. Demaine, Alejandro López-Ortiz, and J. Ian Munro, *Frequency estimation of internet packet streams with limited space.*, ESA (Rolf H. Möhring and Rajeev Raman, eds.), Lecture Notes in Computer Science, vol. 2461, Springer, 2002, pp. 348–360.

[20] I. S. Dhillon, S. Mallela, and R. Kumar, *A divisive information-theoretic feature clustering algorithm for text classification*, JMLR **3** (2003), 1265–1287.

[21] Joan Feigenbaum, Sampath Kannan, Martin Strauss, and Mahesh Viswanathan, *Testing and spot-checking of data streams (extended abstract).*, SODA, 2000, pp. 165–174.

[22] ———, *An approximate L_1-difference algorithm for massive data streams.*, SIAM J. Comput. **32** (2002), no. 1, 131–151.

[23] A. Gilbert, S. Guha, P. Indyk, Y. Kotadis, S. Muthukrishnan, and M. Strauss, *Fast, small-space algorithms for approximate histogram maintanance*, Proc. of STOC (2002).

[24] Sudipto Guha, Andrew McGregor, and Suresh Venkatasubramanian, *Streaming and sublinear approximation of entropy and information distances*, CoRR **cs/0508122** (2005).

[25] Piotr Indyk, *Algorithms for dynamic geometric problems over data streams*, Proc. of STOC (2004), 373–380.

[26] Michael J. Kearns, Yishay Mansour, Dana Ron, Ronitt Rubinfeld, Robert E. Schapire, and Linda Sellie, *On the learnability of discrete distributions.*, STOC, 1994, pp. 273–282.

[27] Balachander Krishnamurthy, Harsha V. Madhyastha, and Suresh Venkatasubramanian, *On stationarity in internet measurements through an information-theoretic lens*, Ist IEEE International Workshop on Networking Meets Databases (NetDB), 2005.

[28] F. Liese and F. Vajda, *Convex statistical distances*, Teubner-Texte zur Mathematik, Band 95, Leipzig (1987).

[29] Adam Meyerson, *Online facility location*, Proc. of FOCS (2001), 426–431.

[30] J. Munro and M. Paterson, *Selection and Sorting with Limited Storage*, Theoretical Computer Science (1980), 315–323.

[31] S. Muthukrishnan, *Data streams: Algorithms and applications*, Survey available on request at `muthu@research.att.com` (2003).

[32] Dana Ron, *Property testing (a tutorial)*, Handbook on Randomization (2000).

[33] N. Tishby, F. Pereira, and W. Bialek, *The information bottleneck method*, Proceedings of the 37-th Annual Allerton Conference on Communication, Control and Computing, 1999, pp. 368–377.

[34] Flemming Topsøe, *Some inequalities for information divergence and related measures of discrimination.*, IEEE Transactions on Information Theory **46** (2000), no. 4, 1602–1609.