Taming Big Probability Distributions

Ronitt Rubinfeld *

June 18, 2012

*CSAIL, Massachusetts Institute of Technology, Cambridge MA 02139 and the Blavatnik School of Computer Science, Tel Aviv University. Research supported by NSF grant 1065125 and the Israel Science Foundation grant no. 1147/09.

These days, it seems that we are constantly bombarded by discussions of "big data" and our lack of tools for processing such vast quantities of information. An important class of big data is most naturally viewed as samples from a probability distribution over a very large domain. Such data occurs in almost every setting imaginable – examples include samples from financial transactions, seismic measurements, neurobiological data, sensor nets, and network traffic records. In many cases there is no explicit description of the distribution – just samples. Even so, in order to effectively make use of such data, one must estimate natural parameters and understand basic properties of the underlying probability distribution. Typical questions include: How many distinct elements have non-zero probability in the distribution? Is the distribution uniform, normal or Zipfian? Is a joint distribution independent? What is the entropy of the distribution? All of these questions can be answered fairly well using classical techniques in a relatively straightforward manner. However, unless assumptions are made on the distribution, such as that the distribution is Gaussian or has certain "smoothness" properties, such techniques use a number of samples that scale at least linearly with the size of the domain of the distributions. Unfortunately, the challenge of big data is that the sizes of the domains of the distributions are immense. The good news is that there has been exciting recent progress in the development of *sub-linear* sample algorithmic tools for such problems!

In this article, we will describe two lines of results, the first on testing the similarity of distributions and the second on estimating the entropy of a distribution, which highlight the main new ideas that have led to this progress. We assume that all of our probability distributions are over a *finite* domain D of size n, but (unless otherwise noted) we do not assume anything else about the distribution.

Closeness to another distribution How can we tell whether two distributions are the same? There are many variants of this question that have been considered, but let's begin with a simpler question, motivated by the following: How many years of lottery results would it take for us to believe in its fairness? In our setting – given samples of a single distribution p, how many samples do we need to determine whether p is the uniform distribution?

To properly formalize this problem, we need to allow some form of approximation, since p could be arbitrarily close to uniform, though not exactly uniform, and no algorithm that takes finite samples would have enough information to detect this. We will use the *property testing framework*: What we ask of our testing algorithm is to "pass" distributions that are uniform and to "fail" distributions that are far from uniform. We next need to decide what we mean by "far" – many distance measures are in common use, but for this article we will use the L_1 distance between two probability distributions p and q is defined as:

$$||p, q||_1 = \sum_{x \in D} |p(x) - q(x)|.$$

For $0 < \epsilon < 1$, we say that p and q are ϵ-*close* with respect to the L_1 distance if $||p, q||_1 \leq \epsilon$. Denote by U_D the uniform distribution on D. Then, on input parameter $0 < \epsilon < 1$, the goal of the testing algorithm will be to pass p if it is uniform and fail if $||p, U_D||_1 \geq \epsilon$. If p is in the middle – not uniform, but not far from uniform – then either "pass" or "fail" is an allowable, and not unreasonable, answer.

One natural way to solve this problem, which we will refer to as the "naive algorithm", is to take enough samples of p so that one can get a good estimate of the probability $p(x)$ for each domain element x. It is easy to see that there are distributions for which such a scheme would require at least linear in $|D| = n$ samples. However, there is a much more efficient $O(\sqrt{n}/\epsilon^4)$ sample algorithm, based on an idea of Goldreich and Ron [GR00] (see also [Pan08] for a more recent algorithm which requires only $O(\sqrt{n}/\epsilon^2)$ samples). This algorithm does not attempt to learn any of the probabilities of specific domain elements according to the distribution p. Instead, the algorithm counts "collisions" – the number of times that samples coincidentally

fall on the same domain element. Slightly more specifically, for a set of k samples $x_1, \ldots, x_k$, let $i, j \in [1..k]$ be two indices of samples. Then we say that i and j "collide" if they output the same domain element, i.e., $x_i = x_j$. Note that the probability that i and j collide does not depend on i, j, and is an important parameter of the distribution p which we will refer to as the collision probability C_p. Consider the fraction of pairs i, j in the actual sample set that collide – it is easy to see that its expectation is exactly C_p. A simple calculation shows that C_p is minimized when p is the uniform distribution, in which case $C_{U_D} = 1/n$. One can show that when p is far from the uniform distribution, then C_p is very different than $1/n$. So now it should be clear that the collision probability C_p is a useful statistic to estimate. The especially convenient property of C_p is that it is a statistic that one can estimate with surprisingly few samples, since k samples yield $\binom{k}{2}$ pairs from which to estimate the collision probability. Although these pairs are not independent, [GR00] have shown that they have nice properties, yielding an algorithm whose sample complexity is $O(\sqrt{n} \log n/\epsilon^4)$ which estimates the collision probability and in turn solves our uniform distribution testing problem. In fact, one cannot do much better in terms of n. It is easy to see that generalized collision statistics (including ℓ-way collisions for all ℓ) are the only information that an algorithm can use to determine whether a distribution is uniform. More than that, generalized collision statistics are the only information that an algorithm can use for determining whether p has any of a large class of properties – namely those properties that are independent of the names of the domain elements, the so called *symmetric properties*! Using this observation, one only needs to find a distribution which has no collisions at all until at least $\Omega(\sqrt{n})$ samples are taken, but on the other hand is very far from uniform. Such a distribution can be constructed by taking the uniform distribution over a random subset S of half of the domain [BFR$^+$00].

What if we want to know whether p is the standard normal distribution? More generally, what if we want to know whether p is the same as another distribution q, where q is known explicitly by the algorithm – that is, $q(i)$ for any domain element i can be determined essentially for free? For example, this would be the case if q is a Gaussian, Zipfian or exponential distribution with known parameters of expectation and variance. Batu et. al. [BFF$^+$01] give an algorithm which solves this problem for *any* q using $O(\sqrt{n} \log n)$ samples from p to perform $O(\log n)$ collision probability estimations over carefully chosen subdomains of D.

Finally, what if both p and q are unknown and the only way we can find out anything about them is to view samples? Up until now, though the analyses are nontrivial, the sample complexities are not terribly surprising to those that have studied "birthday paradox"-type analyses of collisions and hashing. But here the story takes a fascinating turn, since in fact, the complexity of the problem is quite different from $n^{1/2}$! Why? The reason is that there may be elements which are quite "heavy" on which p, q are identical, but whose collision statistics are so big that they hide what is happening on the rest of the domain. Formalizing this lower bound reasoning was quite challenging and eluded researchers for several years, but in 2008, Paul Valiant was able to prove that $\Omega(n^{2/3})$ samples are required for this task [Val08, Val11]. The $O(n^{2/3} \log n \operatorname{poly}(1/\epsilon))$ algorithm, given in 2000 by [BFR$^+$00, BFR$^+$10], distinguishes pairs of distributions p and q that are identical from those pairs p, q which are ϵ-far as follows: First find the "heavy" domain elements, namely, those that have probablity at least $1/n^{2/3}$. Since the domain elements are so heavy, there can be at most $n^{2/3}$ of these, so the naive algorithm which takes $O(n^{2/3} \log n \operatorname{poly}(1/\epsilon))$ samples of p and q and estimates their probabilities on each of the heavy elements is likely to give very good estimates of their probabilities. If p and q seem similar, then check that they are also similar on the rest of the domain by sieving out the heavy elements and using a test that is based on estimating collision probabilities – this time, not just collision probabilities of p and q, but also collisions *between* samples of p and q. Since none of the remaining domain elements are heavy, one can show that $O(n^{2/3} \log n \operatorname{poly}(1/\epsilon))$ samples suffice for this latter task as well.

Such ideas have had further applications: They have been used to design algorithms for testing whether a distribution is monotone increasing or bimodal over the domain [BKR04], or whether a joint distribution is independent [BFF$^+$01]. The sample complexity of many of these problems have been investigated over other distance norms [GR00, GMV09, BNNR11], but in many cases the same collision-based ideas apply. There is much further work on testing similarity, see for example [ADJ$^+$11, GMV09, LRR11].

A *tolerant* test is given two parameters $\epsilon_1 < \epsilon_2$, and is required to pass distributions p that are ϵ_1-close to q and to fail distributions p that are not even ϵ_2-close to q. Unfortunately, even for the problem of testing whether p is the uniform distribution, P. Valiant has shown that for large enough values of ϵ_1, the task becomes much harder, requiring at least $n^{1-o(1)}$ samples [Val08, Val11] (it is known that $n/(\epsilon^2 \log n)$ samples are sufficient [VV11b]). Still, some tiny amount of tolerance can be squeezed out of the more efficient algorithms and it is an interesting direction to see how much more can be achieved.

Estimating the entropy of a distribution. The Shannon entropy of a distribution is an important measure of the randomness of the distribution and the compressibility of the data produced by that distribution – thus, entropy plays a central role in statistics, information theory, data compression and machine learning. The entropy of distribution p over a discrete domain D, is defined as:

$$H(p) \equiv \sum_{x \in D} -p(x) \log p(x).$$

The problems of estimating the entropy of a distribution and the closely related measures of KL-divergence and mutual information have received much attention, because of their usefulness in analyzing data in machine learning and the natural sciences (cf. [Ma81, SKdRvSB98]). How many samples of a distribution are required in order to get a good estimate of the entropy? But first we should ask – what do we mean by a good estimate?

Let us begin with the setting in which one would like an *additive estimate* of the entropy – that is, the algorithm should output a number y such that $H(p) - \epsilon < y < H(p) + \epsilon$ for a given input parameter ϵ. In such a case, one very common estimator for the entropy, sometimes referred to as the "plug-in" estimate, is based on a "best guess" of what the entire distribution p looks like. That is, if $\hat{p}(x)$ is the fraction of times that a domain element x is seen in a sample, then the estimate of the entropy is given by the entropy of $\hat{p}$, namely, $\hat{H} = \sum_{x \in D} -\hat{p}(x) \log \hat{p}(x)$. It is easy to see that for this estimate to have good quality, one must take enough samples to get a good estimate the value of $p(x)$ for most x, which in general is only guaranteed to happen when the number of samples is at least linear in n. Other commonly used estimators such as the *Miller-Madow corrected estimator* and the *jacknifed naive estimator* are similar in that they require linear samples because they do not adequately deal with the contribution to the entropy from the *unseen* samples. A major barrier was broken when in [Pan04], Paninski gave a non-constructive proof of the existence of an algorithm for estimating the entropy with a number of samples that is sublinear in the domain size. Recently, very exciting results of G. Valiant and P. Valiant in [VV11a, VV10a, VV10b, VV11b] settled the longstanding open question of the complexity of this problem. On one hand, they give an $O(n/(\epsilon^2 \log n))$-sample algorithm for estimating the entropy of a distribution over a domain of size n to within an additive error of ϵ (improving on the results of [Val11, RRSS09]). On the other hand, they show that this task is not doable with fewer than $\Omega(n/(\epsilon \log n))$ samples. In order to show the first result, they show that linear programming can be used to find a distribution that is expected to have a similar set of collision statistics to p. Then, this new distribution, though likely to be very different from p in L_1-distance, is similar to p in at least one important way – it is likely to have similar entropy! To show the second result, their lower bound constructs two families of distributions that have very similary collision behaviors, yet are very different in terms of support size or entropy.

Let us now turn to the setting in which a much weaker *multiplicative estimate* of the entropy is sufficient. That is, on input parameter $\gamma > 1$, the algorithm should output a number y such that $H(p)/\gamma < y < \gamma H(p)$. In this case, it suffices to have a number of samples that is dramatically smaller that the domain size – algorithms which use only $O(n^{(1+o(1))/\gamma^2})$ samples can be achieved [BDKR05].[1] Furthermore, it is known that $\Omega(n^{1/\gamma^2})$ samples are necessary for this task [Val08, Val11]. Just to be concrete — this means that one can approximate the entropy to within a multiplicative factor of two using only slightly worse than $O(n^{1/4})$ samples, which is significantly less than what is required for the additive error case! The description of the algorithm is very simple – it uses the plug-in estimate for any domain elements that have high probability, and assumes that the distribution is uniform over the rest of the domain.

Similar results can be achieved for another closely related and well studied task – that of estimating the support size of a distribution. The question of estimating the support size of a distribution has been considered at least since the early 1940's by Fisher and Corbet to estimating the number of butterflies in a region (see [Bun] for a large number of other reasons to consider this problem). In [VV11b], $\theta(n/\log n)$ samples are shown to be both necessary and sufficient to achieve an additive estimate of the support size.

Summary and some final words. The study of big data has led the Computer Science community to make very exciting progress on classical statistical problems. Still, in some settings, the lower bounds can be daunting. One approach to overcoming the lower bounds is to find specialized algorithms for distributions that have convenient properties, such as that of being continuous, monotone, normal... Such assumptions often lead to dramatically better query complexities.

A second approach to overcoming the lower bounds is to note that in some settings it is natural to assume that the algorithm has access to other information in addition to random samples, such as the ability to quickly determine $p(x)$ for any domain element x. For example, when determining distributional properties of data that is stored in sorted order, one can still take a random sample of the data, but it is also easy to determine the number of times that a given element appears in the data set. In another example, Google [FB06] has published their N-gram models, which describe their distribution model on 5-word sequences (from which the $p(x)$ values can be found), along with the complete set of texts on which their model was constructed (from which random samples according to p can be obtained). Algorithms for such models have received less attention, though they have been studied in [BDKR05, RS09]. The efficiency of algorithms in these more powerful models can be exponentially faster, and thus when more powerful queries are available, it is crucial to take advantage of them.

Acknowledgments The author would like to thank Reut Levi and Ning Xie for their very helpful comments on this manuscript.

References

[ADJ+11] J. Acharya, H. Das, A. Jafarpour, A. Orlitsky, and S. Pan. Competitive closeness testing. *Journal of Machine Learning Research*, pages 47–68, 2011. Proceedings Track 19.

[BDKR05] T. Batu, S. Dasgupta, R. Kumar, and R. Rubinfeld. The complexity of approximating the entropy. *SIAM Journal on Computing*, 35(1):132–150, 2005.

[1] This statement is a minor simplifcation, in fact, it only holds for distributions that have at least constant minimum entropy.

[BFF+01] T. Batu, L. Fortnow, E. Fischer, R. Kumar, R. Rubinfeld, and P. White. Testing random variables for independence and identity. In *Proceedings of FOCS*, pages 442–451, 2001.

[BFR+00] T. Batu, L. Fortnow, R. Rubinfeld, W.D. Smith, and P. White. Testing that distributions are close. In *Proceedings of FOCS*, pages 259–269, Los Alamitos, CA, USA, 2000. IEEE Computer Society.

[BFR+10] T. Batu, L. Fortnow, R. Rubinfeld, W. D. Smith, and P. White. Testing closeness of discrete distributions. *CoRR*, abs/1009.5397, 2010. This is a long version of [BFR+00].

[BKR04] T. Batu, R. Kumar, and R. Rubinfeld. Sublinear algorithms for testing monotone and unimodal distributions. In *Proceedings of STOC*, pages 381–390, 2004.

[BNNR11] K. Do Ba, H. L. Nguyen, H. N. Nguyen, and R. Rubinfeld. Sublinear time algorithms for earth mover's distance. *Theory Comput. Syst.*, 48(2):428–442, 2011.

[Bun] J. Bunge. Bibliography of references on the problem of estimating support size. Available at http://www.stat.cornell.edu/ bunge/bibliography.html.

[FB06] A. Franz and T. Brants. All our n-gram are belong to you. `http://googleresearch.blogspot.com/2006/08/all-our-n-gram-are-belong-t` , 2006.

[GMV09] S. Guha, A. McGregor, and S. Venkatasubramanian. Sub-linear estimation of entropy and information distances. *ACM Transactions on Algorithms*, 5, 2009.

[GR00] O. Goldreich and D. Ron. On testing expansion in bounded-degree graphs. *Electronic Colloqium on Computational Complexity*, 7(20), 2000.

[LRR11] R. Levi, D. Ron, and R. Rubinfeld. Testing properties of collections of distributions. In *Proceedings of ICS*, pages 179–194, 2011. See also ECCC TR10-157.

[Ma81] S-K. Ma. Calculation of entropy from data of motion. *J. of Statistical Physics*, 26:221–240, 1981.

[Pan04] L. Paninski. Estimating entropy on m bins given fewer than m samples. *IEEE Transactions on Information Theory*, 50(9):2200–2203, 2004.

[Pan08] L. Paninski. Testing for uniformity given very sparsely-sampled discrete data. *IEEE Transactions on Information Theory*, 54(10):4750–4755, 2008.

[RRSS09] S. Raskhodnikova, D. Ron, A. Shpilka, and A. Smith. Strong lower bonds for approximating distributions support size and the distinct elements problem. *SIAM Journal on Computing*, 39(3):813–842, 2009.

[RS09] R. Rubinfeld and R. Servedio. Testing monotone high-dimensional distributions. *Random Struct. Algorithms*, 34(1):24–44, 2009.

[SKdRvSB98] S. P. Strong, R. Koberle, R. R. de Ruyter van Steveninck, and W. Bialek. Entropy and information in neural spike trains. *Phys. Rev. Lett.*, 80:197–200, 1998.

[Val08] P. Valiant. Testing symmetric properties of distributions. In *Proceedings of STOC*, pages 383–392, 2008.

[Val11] Paul Valiant. Testing symmetric properties of distributions. *SIAM Journal on Computing*, 40(6):1927–1968, 2011.

[VV10a] G. Valiant and P. Valiant. A CLT and tight lower bounds for estimating entropy. Technical Report TR10-179, Electronic Colloquium on Computational Complexity (ECCC), 2010.

[VV10b] G. Valiant and P. Valiant. Estimating the unseen: A sublinear-sample cannonical estimator of distributions. Technical Report TR10-180, Electronic Colloquium on Computational Complexity (ECCC), 2010.

[VV11a] G. Valiant and P. Valiant. Estimating the unseen: an $n/\log(n)$-sample estimator for entropy and support size, shown optimal via new CLTs. In *Proceedings of STOC*, pages 685–694, 2011.

[VV11b] Gregory Valiant and Paul Valiant. The power of linear estimators. In *Proceedings of the 52nd Annual Symposium on Foundations of Computer Science (FOCS 2011)*, pages 403–412, Palm Springs, CA, 2011.