Testing that distributions are close*

Tuğkan Batu[†] Lance Fortnow[‡] Ronitt Rubinfeld[§] Warren D. Smith[¶]
Patrick White[‖]

October 12, 2005

Abstract

Given two distributions over an n element set, we wish to check whether these distributions are statistically close by only sampling. We give a sublinear algorithm which uses $O(n^{2/3}\epsilon^{-4}\log n)$ independent samples from each distribution, runs in time linear in the sample size, makes no assumptions about the structure of the distributions, and distinguishes the cases when the distance between the distributions is small (less than $\max(\frac{\epsilon^2}{32\sqrt[3]{n}}, \frac{\epsilon}{4\sqrt{n}})$) or large (more than ϵ) in L_1-distance. We also give an $\Omega(n^{2/3}\epsilon^{-2/3})$ lower bound.

Our algorithm has applications to the problem of checking whether a given Markov process is rapidly mixing. We develop sublinear algorithms for this problem as well.

*A preliminary version of this paper appeared in the 41st Symposium on Foundations of Computer Science, 2000, Redondo Beach, CA.

[†]Department of Computer and Information Science, University of Pennsylvania, PA, 19104. batu@saul.cis.upenn.edu. This work was partially supported by ONR N00014-97-1-0505, MURI, NSF Career grant CCR-9624552, and an Alfred P. Sloan Research Award.

[‡]NEC Research Institute, 4 Independence Way, Princeton, NJ 08540. fortnow@research.nj.nec.com

[§]NEC Research Institute, 4 Independence Way, Princeton, NJ 08540. ronitt@research.nj.nec.com

[¶]NEC Research Institute, 4 Independence Way, Princeton, NJ 08540. wds@research.nj.nec.com

[‖]Department of Computer Science, Cornell University, Ithaca, NY 14853. white@cs.cornell.edu.This work was partially supported by ONR N00014-97-1-0505, MURI, NSF Career grant CCR-9624552, and an Alfred P. Sloan Research Award.

1 Introduction

Suppose we have two distributions over the same n element set, and we want to know whether they are close to each other in L_1-norm. We assume that we know nothing about the structure of the distributions and that the only allowed operation is independent sampling. The naive approach would, for each distribution, sample enough elements to approximate the distribution and then compare these approximations. Theorem 25 in Section 3.4 shows that the naive approach requires at least a linear number of samples.

In this paper, we develop a method of testing that the distance between two distributions is at most ϵ using considerably fewer samples. If the distributions have L_1-distance at most $\max(\frac{\epsilon^2}{32\sqrt[3]{n}}, \frac{\epsilon}{4\sqrt{n}})$ then the algorithm will accept with probability at least $1-\delta$. If the distributions have L_1-distance more than ϵ then the algorithm will accept with probability at most δ. The number of samples used is $O(n^{2/3}\epsilon^{-4}\log n \log \frac{1}{\delta})$. We give an $\Omega(n^{2/3}\epsilon^{-2/3})$ lower bound for testing L_1-distance.

Our test relies on a test for the L_2-distance, which is considerably easier to test: we give an algorithm that uses a number of samples which is independent of n. However, the L_2-distance does not in general give a good measure of the closeness of two distributions. For example, two distributions can have disjoint support and still have small L_2-distance. Still, we can get a very good estimate of the L_2-distance and then we use the fact that the L_1-distance is at most $\sqrt{n}$ times the L_2-distance. Unfortunately, the number of queries required by this approach is too large in general. Because of this, our L_1-test is forced to distinguish two cases.

For distributions with small L_2-norm, we show how to use the L_2-distance to get a good approximation of the L_1-distance. For distributions with larger L_2-norm, we use the fact that such distributions must have elements which occur with relatively high probability. We create a filtering test that estimates the L_1-distance due to these high probability elements, and then approximates the L_1-distance due to the low probability elements using the test for L_2-distance. Optimizing the notion of "high probability" yields our $O(n^{2/3}\epsilon^{-4}\log n \log \frac{1}{\delta})$ algorithm. The L_2-distance test uses $O(\epsilon^{-4}\log(1/\delta))$ samples.

Applying our techniques to Markov chains, we use the above algorithm as a basis for constructing tests for determining whether a Markov chain is rapidly mixing. We show how to test whether iterating a Markov chain for t steps causes it to reach a distribution close to the stationary distribution. Our testing algorithm works by following $\tilde{O}(tn^{5/3})$ edges in the chain. When the Markov chain is represented in a convenient way (such a representation can be computed in linear time and we give an example representation in Section 4), this test remains sublinear in the size of a dense enough Markov chain for small t. We then investigate two notions of being *close* to a rapidly mixing Markov chain that fall within the framework of property testing, and show how to test that a Markov chain is close to a Markov chain that mixes in t steps by following only $\tilde{O}(tn^{2/3})$ edges. In the case of Markov chains that come from directed graphs and pass our test, our theorems show the existence of a directed graph that is close to the original one and rapidly mixing.

Related Work Our results fall within the various frameworks of property testing [26, 16, 17, 9, 25]. A related work of Kannan and Yao [21] outlines a program checking framework for certifying the randomness of a program's output. In their model, one does not assume that samples from the input distribution are independent.

There is much work on the problem estimating the distance between distributions in data streaming models where space is limited rather than time (cf. [14, 2, 10, 12]). Another line of work [5] estimates the distance in frequency count distributions on words between various documents, where again space is limited.

In an interactive setting, Sahai and Vadhan [27] show that given distributions p and q, generated by polynomial-size circuits, the problem of distinguishing whether p and q are close or far in L_1-norm, is complete for statistical zero-knowledge.

There is a vast literature on testing statistical hypotheses. In these works, one is given examples chosen from the same distribution out of two possible choices, say p and q. The goal is to decide which of two distributions the examples are coming from. More generally, the goal can be stated as deciding which of two known classes of distributions contains the distribution generating the examples. This can be seen to be a generalization of our model as follows: Let the first class of distributions be the set of distributions of the form $q \times q$. Let the second class of distributions be the set of distributions of the form $q_1 \times q_2$ where the L_1 difference of q_1 and q_2 is at least ϵ. Then, given examples from two distributions p_1, p_2, create a set of example pairs (x, y) where x is chosen according to p_1 and y according to p_2. Bounds and an optimal algorithm for the general problem for various distance measures are given in [6, 23, 7, 8, 22]. None of these give sublinear bounds in the domain size for our problem. The specific model of singleton hypothesis classes is studied by Yamanishi [31].

Goldreich and Ron [18] give methods allowing testing that the L_2-distance between a given distribution and the uniform distribution is small in time $O(\sqrt{n})$. Their "collision" idea underlies the present paper. Based on this, they give a test which they conjecture can be used for testing whether a regular graph is close to being an expander, where by close they mean that by changing a small fraction of the edges they can turn it into an expander. Their test is based on picking a random node and testing that random walks from this node reach a distribution that is close to uniform. Our tests are based on similar principles, but we do not prove their conjecture. Mixing and expansion are known to be related [28], but our techniques only apply to the mixing properties of random walks on directed graphs, since the notion of closeness we use does not preserve the symmetry of the adjacency matrix. In another work, Goldreich and Ron [17] show that testing that a graph is close to an expander requires $\Omega(n^{1/2})$ queries.

The conductance [28] of a graph is known to be closely related to expansion and rapid-mixing properties of the graph [20][28]. Frieze and Kannan [13] show, given a graph G with n vertices and α, one can approximate the conductance of G to within additive error α in time $O(n2^{\tilde{O}(1/\alpha^2)})$. Their techniques also yield an $O(2^{\mathrm{poly}(1/\epsilon)})$ time test which determines whether an adjacency matrix of a graph can be changed in at most ϵ fraction of the locations to get a graph with high conductance. However, for the purpose of testing whether an n-vertex, m-edge graph is rapid mixing, we would need to approximate its conductance to within $\alpha = O(m/n^2)$; thus only when $m = \Theta(n^2)$ would it run in $O(n)$ time.

It is known that mixing [28, 20] is related to the separation between the two largest eigenvalues [3]. Standard techniques for approximating the eigenvalues of a dense $n \times n$ matrix run in $\Theta(n^3)$ flops and consume $\Theta(n^2)$ words of memory [19]. However, for a sparse $n \times n$ *symmetric* matrix with m nonzero entries, $n \leq m$, "Lanczos algorithms" [24] accomplish the same task in $\Theta(n[m + \log n])$ flops, consuming $\Theta(n+m)$ storage. Furthermore, it is found in practice that these algorithms can be run for far fewer, even a constant number, of iterations while still obtaining highly accurate values for the outer and inner few eigenvalues. Our test for rapid mixing of a Markov chain runs more slowly than the algorithms that are used in practice except on fairly dense graphs ($m \gg tn^{5/3} \log n$). However, our test is more efficient than algorithms whose behavior is mathematically justified at every sparsity level. Our faster, but weaker, tests of various altered definitions of "rapid mixing," are more efficient than the current algorithms used in practice.

2 Preliminaries

We use the following notation. We denote the set $\{1, \ldots, n\}$ as $[n]$. The notation $x \in_R [n]$ denotes that x is chosen uniformly at random from the set $[n]$. The L_1-norm of a vector $\vec{v}$ is denoted by $|\vec{v}|$ and is equal to $\sum_{i=1}^{n} |v_i|$. Similarly the L_2-norm is denoted by $\|\vec{v}\|$ and is equal to $\sqrt{\sum_{i=1}^{n} v_i^2}$, and $\|\vec{v}\|_\infty = \max_i |v_i|$. We assume our distributions are discrete distributions over n elements, and will represent a distribution as a vector $\vec{p} = (p_1, \ldots, p_n)$ where p_i is the probability of outputting element i.

The *collision probability* of two distributions $\vec{p}$ and $\vec{q}$ is the probability that a sample from each of $\vec{p}$ and $\vec{q}$ yields the same element. Note that, for two distributions $\vec{p}, \vec{q}$, the collision probability is $\vec{p} \cdot \vec{q} = \sum_i p_i q_i$. To avoid ambiguity, we refer to the collision probability of $\vec{p}$ and $\vec{p}$ as the *self-collision probability* of $\vec{p}$, note that the self-collision probability of $\vec{p}$ is $\|\vec{p}\|^2$.

3 Testing closeness of distributions

The main goal of this section is to show how to test that two distributions $\vec{p}$ and $\vec{q}$ are close in L_1-norm in sublinear time in the size of the domain of the distributions. We are given access to these distributions via black boxes which upon a query respond with an element of $[n]$ generated according to the respective distribution. Our main theorem is:

Theorem 1 *Given parameter δ, and distributions $\vec{p}, \vec{q}$ over a set of n elements, there is a test which runs in time $O(\epsilon^{-4} n^{2/3} \log n \log \frac{1}{\delta})$ such that if $|\vec{p} - \vec{q}| \leq \max(\frac{\epsilon^2}{32 \sqrt[3]{n}}, \frac{\epsilon}{4\sqrt{n}})$, then the test outputs* `pass` *with probability at least $1 - \delta$ and and if $|\vec{p} - \vec{q}| > \epsilon$, then the test outputs* `fail` *with probability at least $1 - \delta$.*

In order to prove this theorem, we give a test which determines whether $\vec{p}$ and $\vec{q}$ are close in L_2-norm. The test is based on estimating the self-collision and collision probabilities of $\vec{p}$ and $\vec{q}$. In particular, if $\vec{p}$ and $\vec{q}$ are close, one would expect that the self-collision probabilities of each are close to the collision probability of the pair. Formalizing this intuition, in Section 3.1, we prove:

Theorem 2 *Given parameter δ, and distributions $\vec{p}$ and $\vec{q}$ over a set of n elements, there exists a test such that if $\|\vec{p} - \vec{q}\| \leq \epsilon/2$ then the test passes with probability at least $1 - \delta$. If $\|\vec{p} - \vec{q}\| > \epsilon$ then the test passes with probability less than δ. The running time of the test is $O(\epsilon^{-4} \log \frac{1}{\delta})$.*

The test used to prove Theorem 2 is given in Figure 1. The number of pairwise self-collisions in set F is the count of $i < j$ such that the i^{th} sample in F is same as the j^{th} sample in F. Similarly, the number of collisions between Q_p and Q_q is the count of (i, j) such that the i^{th} sample in Q_p is same as the j^{th} sample in Q_q. We use the parameter m to indicate the number of samples needed by the test to get constant confidence. In order to bound the L_2-distance between $\vec{p}$ and $\vec{q}$ by ϵ, setting $m = O(\frac{1}{\epsilon^4})$ suffices. By maintaining arrays which count the number of times that each element is sampled in F_p, F_q, one can achieve the claimed running time bounds. Thus essentially m^2 estimations of the collision probability can be performed in $O(m)$ time. Using hashing techniques, one can achieve $O(m)$ with an expected running time bound matching Theorem 2.

Since $|v| \leq \sqrt{n}\|v\|$, a simple way to extend the above test to an L_1-distance test is by setting $\epsilon' = \epsilon/\sqrt{n}$. Unfortunately, due to the order of the dependence on ϵ in the L_2-distance test, the resulting running time is prohibitive. It is possible, though, to achieve sublinear running times if the input vectors are known to be reasonably evenly distributed. We make this precise by a closer analysis of the variance of the test in Lemma 5. In particular, we analyze the dependence of the

L_2-Distance-Test$(p, q, m, \epsilon, \delta)$
Repeat $O(\log(\frac{1}{\delta}))$ times
  Let $F_p =$ a set of m samples from $\vec{p}$
  Let $F_q =$ a set of m samples from $\vec{q}$
    Let r_p be the number of pairwise
      self-collisions in F_p.
    Let r_q be the number of pairwise
      self-collisions in F_q.
  Let $Q_p =$ a set of m samples from $\vec{p}$
  Let $Q_q =$ a set of m samples from $\vec{q}$
    Let s_{pq} be the number of collisions
      between Q_p and Q_q.
  Let $r = \frac{2m}{m-1}(r_p + r_q)$
  Let $s = 2s_{pq}$
  If $r - s > m^2\epsilon^2/2$ then reject
Reject if the majority of iterations reject,
accept otherwise

Figure 1: Algorithm L_2-Distance-Test

variance of s on the parameter $b = \max(\|\vec{p}\|_\infty, \|\vec{q}\|_\infty)$. There we show that given $\vec{p}$ and $\vec{q}$ such that $b = O(n^{-\alpha})$, one can call L_2-**Distance-Test** with an error parameter of $\frac{\epsilon}{\sqrt{n}}$ and achieve running time of $O(\epsilon^{-4}(n^{1-\alpha/2} + n^{2-2\alpha}))$.

We use the following definition to identify the elements with large weights.

Definition 3 *An element i is called **big** with respect to a distribution $\vec{p}$ if $p_i > \frac{1}{n^{2/3}}$.*

Our L_1-distance tester calls the L_2-distance testing algorithm as a subroutine. When both input distributions have no big elements, the input is passed to the L_2-distance test unchanged. If the input distributions have a large self-collision probability, the distances induced respectively by the big and non-big elements are measured in two steps. The first step measures the distance corresponding to the big elements via straightforward sampling, and the second step modifies the distributions so that the distance attributed to the non-big elements can be measured using the L_2-distance test. The complete test is given in Figure 2. The proof of Theorem 1 is described in Section 3.2.

In Section 3.4 we prove that $\Omega(n^{2/3})$ samples are required for distinguishing distributions that are far in L_1-distance.

3.1 Closeness in L_2-norm

In this section we analyze the test in Figure 1 and prove Theorem 2. The statistics r_p, r_q and s in Algorithm L_2-**Distance-Test** are estimators for the self-collision probability of $\vec{p}$, of $\vec{q}$, and of the collision probability between $\vec{p}$ and $\vec{q}$, respectively. If $\vec{p}$ and $\vec{q}$ are statistically close, we expect that the self-collision probabilities of each are close to the collision probability of the pair. These probabilities are exactly the inner products of these vectors. In particular if the set F_p of samples from $\vec{p}$ is given by $\{F_p^1, \ldots, F_p^m\}$ then for any pair $i, j \in [m], i \neq j$ we have that $\Pr\left[F_p^i = F_p^j\right] = \vec{p} \cdot \vec{p} = \|\vec{p}\|^2$. By combining these statistics, we show that $r - s$ is an estimator for the desired value $\|\vec{p} - \vec{q}\|^2$.

```
$L_1$-Distance-Test($p, q, \epsilon, \delta$)
Sample $\vec{p}$ and $\vec{q}$ for
  $M = O(\max(\epsilon^{-2}, 4) n^{2/3} \log n)$ times
Let $S_p$ and $S_q$ be the sample sets obtained
  by discarding elements that occur less
  than $(1 - \epsilon/63) M n^{-2/3}$ times
If $S_p$ and $S_q$ are empty
    $L_2$-Distance-Test($p, q, O(n^{2/3}/\epsilon^4), \frac{\epsilon}{2\sqrt{n}}, \delta/2$)
else
  $\ell_i^p = \#$ times element $i$ appears in $S_p$
  $\ell_i^q = \#$ times element $i$ appears in $S_q$
  Fail if $\sum_{i \in S_p \cup S_q} |\ell_i^p - \ell_i^q| > \epsilon M/8$.
  Define $\vec{p}'$ as follows:
    sample an element from $\vec{p}$
    if this sample is not in $S_p \cup S_q$ output it,
    otherwise output an $x \in_R [n]$.
  Define $\vec{q}'$ similarly.
  $L_2$-Distance-Test($p', q', O(n^{2/3}/\epsilon^4), \frac{\epsilon}{2\sqrt{n}}, \delta/2$)
```

Figure 2: Algorithm L_1-Distance-Test

Since our algorithm samples from not one but two distinct distributions, we must also bound the variance of the variable s used in the test. One distinction to make between self-collisions and $\vec{p}, \vec{q}$ collisions is that for the self-collision we only consider samples for which $i \neq j$, but this is not necessary for $\vec{p}, \vec{q}$ collisions. We accommodate this in our algorithm by scaling r_p and r_q appropriately. By this scaling and from the above discussion we see that $\mathrm{E}[s] = 2m^2(\vec{p} \cdot \vec{q})$ and that $\mathrm{E}[r - s] = m^2(\|\vec{p}\|^2 + \|\vec{q}\|^2 - 2(\vec{p} \cdot \vec{q})) = m^2(\|\vec{p} - \vec{q}\|^2)$.

A complication which arises from this scheme, though, is that the pairwise samples are not independent. Thus we use Chebyshev's inequality. That is, for any random variable A, and $\rho > 0$, the probability $\Pr[|A - E[A]| > \rho]$ is bounded above by $\frac{\mathrm{Var}[A]}{\rho^2}$. To use this theorem, we require a bound on the variance, which we give in this section.

Our techniques extend the work of Goldreich and Ron [18], where self-collision probabilities are used to estimate norm of a vector, and the deviation of a distribution from uniform. In particular, their work provides an analysis of the statistics r_p and r_q above through the following lemma.

Lemma 4 ([18]) *Let A be one of r_p or r_q in algorithm L_2-**Distance-Test**. Then* $\mathrm{E}[A] = \binom{m}{2} \cdot \|\vec{p}\|^2$ *and* $\mathrm{Var}[A] \leq 2(\mathrm{E}[A])^{3/2}$

The variance bound is more complicated, and is given in terms of the largest weight in $\vec{p}$ and $\vec{q}$.

Lemma 5 *There is a constant c such that* $\mathrm{Var}[r - s] \leq c(m^3 b^2 + m^2 b)$, *where* $b = \max(\|\vec{p}\|_\infty, \|\vec{q}\|_\infty)$.

Proof: Let F be the set $\{1, \ldots, m\}$. For $(i, j) \in F \times F$, define the indicator variable $C_{i,j} = 1$ if the i^{th} element of Q_p and the j^{th} element of Q_q are the same. Then the variable from the algorithm $s_{pq} = \sum_{i,j} C_{i,j}$. Also define the notation $\bar{C}_{i,j} = C_{i,j} - \mathrm{E}[C_{i,j}]$.

Now $\mathrm{Var}[\sum_{F \times F} C_{i,j}] = \mathrm{E}[(\sum_{F \times F} \bar{C}_{i,j})^2] = \mathrm{E}\left[\sum_{i,j} (\bar{C}_{i,j})^2 + 2\sum_{(i,j) \neq (k,l)} \bar{C}_{i,j} \bar{C}_{k,l}\right] \leq m^2(\vec{p} \cdot \vec{q}) + 2\mathrm{E}\left[\sum_{(i,j) \neq (k,l)} \bar{C}_{i,j} \bar{C}_{k,l}\right]$.

To analyze the last expectation, we use two facts. First, it is easy to see, by the definition of covariance, that $\mathrm{E}\left[\bar{C}_{i,j}\bar{C}_{k,l}\right] \le \mathrm{E}\left[C_{i,j}C_{k,l}\right]$. Secondly, we note that $C_{i,j}$ and $C_{k,l}$ are not independent only when $i = k$ or $j = l$. Expanding the sum we get

$$\begin{aligned}
&\mathrm{E}\left[\sum_{\substack{(i,j),(k,l)\in F\times F\\(i,j)\neq(k,l)}} \bar{C}_{i,j}\bar{C}_{k,l}\right] \\
&= \mathrm{E}\left[\sum_{\substack{(i,j),(i,l)\in F\times F\\ j\neq l}} \bar{C}_{i,j}\bar{C}_{i,l} + \sum_{\substack{(i,j),(k,j)\in F\times F\\ i\neq k}} \bar{C}_{i,j}\bar{C}_{k,j}\right] \\
&\le \mathrm{E}\left[\sum_{\substack{(i,j),(i,l)\in F\times F\\ j\neq l}} C_{i,j}C_{i,l} + \sum_{\substack{(i,j),(k,j)\in F\times F\\ i\neq k}} C_{i,j}C_{k,j}\right] \\
&\le cm^3 \sum_{\ell\in[n]} p_\ell q_\ell^2 + p_\ell^2 q_\ell \le cm^3 b^2 \sum_{\ell\in[n]} q_\ell \le cm^3 b^2
\end{aligned}$$

for some constant c. Next, we bound $\mathrm{Var}\left[r\right]$ similarly to $\mathrm{Var}\left[s\right]$ using the argument in the proof of Lemma 4 from [18]. Consider an analogous calculation to the preceding inequality for $\mathrm{Var}\left[r_p\right]$ (similarly, for $\mathrm{Var}\left[r_q\right]$) where $X_{ij} = 1$ for $1 \le i < j \le m$ if the ith and jth samples in F_p are the same. Similarly to above, define $\bar{X}_{ij} = X_{ij} - \mathrm{E}\left[X_{ij}\right]$. Then, we get

$$\begin{aligned}
\mathrm{Var}\left[r\right] &= \mathrm{E}\left[\left(\sum_{1\le i<j\le m} \bar{X}_{ij}\right)^2\right] \\
&= \sum_{1\le i<j\le m} \mathrm{E}\left[\bar{X}_{i,j}^2\right] + 4\sum_{1\le i<j<k\le m} \mathrm{E}\left[\bar{X}_{i,j}\bar{X}_{i,k}\right] \\
&\le \binom{m}{2}\cdot\sum_{t\in[n]} p_t^2 + 4\cdot\binom{m}{3}\sum_{t\in[n]} p_t^3 \\
&\le O(m^2)\cdot b + O(m^3)\cdot b^2.
\end{aligned}$$

Since variance is additive for independent random variables, we can write $\mathrm{Var}\left[r-s\right] \le c(m^3b^2 + m^2b)$. □

Now using Chebyshev's inequality, it follows that if we choose $m = O(\epsilon^{-4})$, we can achieve an error probability less than 1/3. It follows from standard techniques that with $O(\log\frac{1}{\delta})$ iterations we can achieve an error probability at most δ.

Lemma 6 *For two distributions $\vec{p}$ and $\vec{q}$ such that $b = \max(\|\vec{p}\|_\infty, \|\vec{q}\|_\infty)$ and $m = O((b^2 + \epsilon^2\sqrt{b})/\epsilon^4)$, if $\|\vec{p}-\vec{q}\| \le \epsilon/2$, then L_2-**Distance-Test**(p,q,m,ϵ,δ) passes with probability at least $1-\delta$. If $\|\vec{p}-\vec{q}\| > \epsilon$ then L_2-**Distance-Test**(p,q,m,ϵ,δ) passes with probability less than δ. The running time is $O(m\log(\frac{1}{\delta}))$.*

PROOF: For our statistic $A = (r-s)$ we can say, using Chebyshev's inequality, that for some constant k,

$$\Pr\left[|A - \mathrm{E}\left[A\right]| > \rho\right] \le \frac{k(m^3b^2 + m^2b)}{\rho^2}$$

Then when $\|\vec{p}-\vec{q}\| \leq \epsilon/2$, for one iteration,

$$\begin{aligned}
\Pr[\text{pass}] &= \Pr\left[(r-s) < m^2\epsilon^2/2\right] \\
&\geq \Pr\left[|(r-s) - \mathrm{E}[r-s]| < m^2\epsilon^2/4\right] \\
&\geq 1 - \tfrac{4k(m^3b^2+m^2b)}{m^4\epsilon^4}
\end{aligned}$$

It can be shown that this probability will be at least $2/3$ whenever $m > c(b^2 + \epsilon^2\sqrt{b})/\epsilon^4$ for some constant c. A similar analysis can be used to show the other direction. □

3.2 Closeness in L_1-norm

The L_1-closeness test proceeds in two stages. The first phase of the algorithm filters out big elements (as defined in Definition 3) while estimating their contribution to the distance $|\vec{p}-\vec{q}|$. The second phase invokes the L_2-test on the filtered distribution, with closeness parameter $\frac{\epsilon}{2\sqrt{n}}$. The correctness of this subroutine call is given by Lemma 6 with $b = n^{-2/3}$. With these substitutions, the number of samples m is $O(\epsilon^{-4}n^{2/3})$. The choice of threshold $n^{-2/3}$ for the weight of the big elements arises from optimizing the running-time trade-off between the two phases of the algorithm.

We need to show that by using a sample of size $O(\epsilon^{-2}n^{2/3}\log n)$, we can estimate the weights of the big elements to within a multiplicative factor of $O(\epsilon)$.

Lemma 7 *Let $\epsilon \leq 1/2$. In L_1**-Distance-Test**, after performing $M = O(\frac{n^{2/3}\log n}{\epsilon^2})$ samples from a distribution $\vec{p}$, we define $\bar{p}_i = \ell_i^p/M$. Then, with probability at least $1-\frac{1}{n}$, the following hold for all i: (1) if $p_i \geq \epsilon^2 n^{-2/3}$ then $|\bar{p}_i - p_i| < \frac{\epsilon}{63}\max(p_i, n^{-2/3})$, (2) if $p_i < \epsilon^2 n^{-2/3}$, $\bar{p}_i < (1-\epsilon/63)n^{-2/3}$ and $|\bar{p}_i - p_i| < p_i/32$.*

PROOF: We analyze three cases; we use Chernoff bounds to show that for each i, with probability at least $1-\frac{1}{n^2}$, the following holds: (1a) If $p_i > n^{-2/3}$ then $|\bar{p}_i - p_i| < \epsilon p_i/63$. (1b) If $\epsilon^2 n^{-2/3} < p_i \leq n^{-2/3}$ then $|\bar{p}_i - p_i| < \epsilon n^{-2/3}/63$. (2) If $p_i < \epsilon^2 n^{-2/3}$ then $\bar{p}_i < 3\epsilon^2 n^{-2/3}$. Since, for $\epsilon \leq 1/2$, $3\epsilon^2 \leq (1-\epsilon/63)$. Another application of Chernoff bounds gives us $\Pr[|\bar{p}_i - p_i| > p_i/32] \leq 1/n^{-2}$. The lemma follows. □

Once the big elements are identified, we use the following fact to prove the gap in the distances of accepted and rejected pairs of distributions.

Fact 8 *For any vector v, $\|v\|^2 \leq |v| \cdot \|v\|_\infty$.*

Theorem 9 *L_1**-Distance-Test** passes distributions $\vec{p}, \vec{q}$ such that $|\vec{p}-\vec{q}| \leq \max(\frac{\epsilon^2}{32\sqrt[3]{n}}, \frac{\epsilon}{4\sqrt{n}})$, and fails when $|\vec{p}-\vec{q}| > \epsilon$. The error probability is δ. The running time of the whole test is $O(\epsilon^{-4}n^{2/3}\log n \log(\frac{1}{\delta}))$.*

PROOF: Suppose items (1) and (2) from Lemma 7 hold for all i, and for both $\vec{p}$ and $\vec{q}$. By Lemma 7, this event happens with probability at least $1-\frac{2}{n}$.

Let $S = S_p \cup S_q$. By our assumption, all the big elements of both $\vec{p}$ and $\vec{q}$ are in S, and no element with weight less than $\epsilon^2 n^{-2/3}$ (in either distribution) is in S.

Let Δ_1 be the L_1-distance attributed to the elements in S. Let $\Delta_2 = |\vec{p'} - \vec{q'}|$ (in the case that S is empty, $\Delta_1 = 0$, $\vec{p} = \vec{p'}$ and $\vec{q} = \vec{q'}$).

Notice that $\Delta_1 \leq |\vec{p}-\vec{q}|$. We can show that $\Delta_2 \leq |\vec{p}-\vec{q}|$, and $|\vec{p}-\vec{q}| \leq 2\Delta_1 + \Delta_2$.

The algorithm estimates Δ_1 in a brute-force manner to within an additive error of $\epsilon/9$. By Lemma 7, the error on the i^{th} term of the sum is bounded by $\frac{\epsilon}{63}(\max(p_i, n^{-2/3}) + \max(q_i, n^{-2/3})) \leq$

$\frac{\epsilon}{63}(p_i + q_i + 2n^{-2/3})$. Consider the sum over i of these error terms. Notice that this sum is over at most $2n^{2/3}/(1-\epsilon/63)$ elements in S. Hence, the total additive error is bounded by

$$\sum_{i \in S} \frac{\epsilon}{63}(p_i + q_i + 2n^{-2/3}) \leq \frac{\epsilon}{63}(2 + 4/(1-\epsilon/63)) \leq \epsilon/9.$$

Note that $\max(\|\vec{p}'\|_\infty, \|\vec{q}'\|_\infty) < n^{-2/3} + n^{-1}$. So, we can use the L_2**-Distance-Test** on $\vec{p}'$ and $\vec{q}'$ with $m = O(\epsilon^{-4} n^{2/3})$ as shown by Lemma 6.

If $|\vec{p} - \vec{q}| < \frac{\epsilon^2}{32\sqrt[3]{n}}$ then so are Δ_1 and Δ_2. The first phase of the algorithm clearly passes. By Fact 8, $\|\vec{p}' - \vec{q}'\| \leq \frac{\epsilon}{4\sqrt{n}}$. Therefore, the L_2**-Distance-Test** passes. Similarly, if $|\vec{p} - \vec{q}| > \epsilon$ then either $\Delta_1 > \epsilon/4$ or $\Delta_2 > \epsilon/2$. Either the first phase of the algorithm or the L_2**-Distance-Test** will fail.

To get the running time, note that the time for the first phase is $O(\epsilon^{-2} n^{2/3} \log n)$ and that the time for L_2**-Distance-Test** is $O(n^{2/3}\epsilon^{-4} \log \frac{1}{\delta})$. It is easy to see that our algorithm makes an error either when it makes a bad estimation of Δ_1 or when L_2**-Distance-Test** makes an error. So, the probability of error is bounded by δ. □

We believe we can eliminate the $\log n$ term in Theorem 1 (and Theorem 9). Instead of requiring that we correctly identify the big and small elements, we allow some misclassifications. The filtering test should not misclassify very many very big and very small elements and a good analysis should show that our remaining tests will not have significantly different behavior.

The next theorem improves this result by looking at the dependence of the variance calculation in Section 3.1 on L_∞ norms of the distributions separately.

Theorem 10 *Given two black-box distributions* $\mathbf{p}, \mathbf{q}$ *over* $[n]$, *with* $\|\mathbf{p}\|_\infty \leq \|\mathbf{q}\|_\infty$, *there is a test requiring* $O((n^2\|\mathbf{p}\|_\infty\|\mathbf{q}\|_\infty\epsilon^{-4} + \sqrt{n}\|\mathbf{p}\|_\infty\epsilon^{-2})\log(1/\delta))$ *samples that (1) if* $|\mathbf{p} - \mathbf{q}| \leq \frac{\epsilon^2}{\sqrt[3]{n}}$, *it outputs PASS with probability at least* $1-\delta$ *and (2) if* $|\mathbf{p} - \mathbf{q}| > \epsilon$, *it outputs FAIL with probability at least* $1-\delta$.

Finally, by similar methods to the proof of Theorem 10 (in conjunction with those of [18]), we can show the following (proof omitted):

Theorem 11 *Given a black-box distribution* $\mathbf{p}$ *over* $[n]$, *there is a test that takes* $O(\epsilon^{-4}\sqrt{n}\log(n)\log(1/\delta))$ *samples, outputs PASS with probability at least* $1-\delta$ *if* $\mathbf{p} = U_{[n]}$, *and outputs FAIL with probability at least* $1-\delta$ *if* $|\mathbf{p} - U_{[n]}| > \epsilon$.

3.3 Characterization of canonical algorithms for testing properties of distributions

In this section, we characterize canonical algorithms for testing properties of distributions defined by permutation-invariant functions. The argument hinges on the irrelevance of the labels of the domain elements for such a function. We obtain this canonical form in two steps, corresponding to the two lemmas below. The first step makes explicit the intuition that such an algorithm should be symmetric, that is, the algorithm would not benefit from discriminating among the labels. In the second step, we remove the use of labels altogether, and show that we can present the sample to the algorithm in an aggregate fashion.

Characterizations of property testing algorithms have been studied in other settings. For example, using similar techniques, Alon et al. [1] show a canonical form for algorithms for testing graph properties. Later, Goldreich and Trevisan [15] formally prove the result by Alon et al. In a different setting, Bar-Yossef et al. [4] show a canonical form for sampling algorithms that approximate symmetric functions of the form $f : A^n \to B$ where A and B are arbitrary sets. In the latter setting, the algorithm is given oracle access to the input vector and takes samples from the coordinate values of this vector.

Definition 12 (Permutation of a distribution) *For a distribution $\mathbf{p}$ over $[n]$ and a permutation π on $[n]$, define $\pi(\mathbf{p})$ to be the distribution such that for all i, $\pi(\mathbf{p})_{\pi(i)} = p_i$.*

Definition 13 (Symmetric Algorithm) *Let $\mathcal{A}$ be an algorithm that takes samples from k discrete black-box distributions over $[n]$ as input. We say that $\mathcal{A}$ is* symmetric *if, once the distributions are fixed, the output distribution of $\mathcal{A}$ is identical for any permutation of the distributions.*

Definition 14 (Permutation-invariant function) *A k-ary function f on distributions over $[n]$ is* permutation-invariant *if for any permutation π on $[n]$, and all distributions $(\mathbf{p}^{(1)}, \ldots, \mathbf{p}^{(k)})$,*

$$f(\mathbf{p}^{(1)}, \ldots, \mathbf{p}^{(k)}) = f(\pi(\mathbf{p}^{(1)}), \ldots, \pi(\mathbf{p}^{(k)})).$$

Lemma 15 *Let $\mathcal{A}$ be an arbitrary testing algorithm for a k-ary property $\mathcal{P}$ defined by a permutation-invariant function. Suppose $\mathcal{A}$ has sample complexity $s(n)$, where n is the domain size of the distributions. Then, there exists a symmetric algorithm that tests the same property of distributions with sample complexity $s(n)$.*

Proof: Given the algorithm $\mathcal{A}$, construct a symmetric algorithm $\mathcal{A}'$ as follows: Choose a random permutation of the domain elements. Upon taking $s(n)$ samples, apply this permutation to each sample. Pass this (renamed) sample set to $\mathcal{A}$ and output according to $\mathcal{A}$.

It is clear that the sample complexity of the algorithm does not change. We need to show that the new algorithm also maintains the testing features of $\mathcal{A}$. Suppose that the input distributions $(\mathbf{p}^{(1)}, \ldots, \mathbf{p}^{(k)})$ have the property $\mathcal{P}$. Since the property is defined by a permutation-invariant function, any permutation of the distributions maintains this property. Therefore, the permutation of the distributions should be accepted as well. Then,

$$\Pr\left[\mathcal{A}' \text{ accepts } (\mathbf{p}^{(1)}, \ldots, \mathbf{p}^{(k)})\right] = \sum_{\text{perm. } \pi} \frac{1}{n!} \Pr\left[\mathcal{A} \text{ accepts } (\pi(\mathbf{p}^{(1)}), \ldots, \pi(\mathbf{p}^{(k)}))\right],$$

which is at least $2/3$ by the accepting probability of $\mathcal{A}$.

An analogous argument on the failure probability for the case of the distributions $(\mathbf{p}^{(1)}, \ldots, \mathbf{p}^{(k)})$ that should be rejected completes the proof. □

In order to avoid introducing additional randomness in $\mathcal{A}'$, we can try $\mathcal{A}$ on all possible permutations and output the majority vote. This change would not affect the sample complexity, and it can be shown that it maintains correctness.

Definition 16 (Fingerprint of a sample) *Let S_1 and S_2 be multisets of at most s samples taken from two black-box distributions over $[n]$, $\mathbf{p}$ and $\mathbf{q}$, respectively. Let the random variable C_{ij}, for $0 \le i, j \le s$, denote the number of elements that appear exactly i times in S_1 and exactly j times in S_2. The collection of values that the random variables $\{C_{ij}\}_{0 \le i,j \le s}$ take is called the* fingerprint *of the sample.*

For example, let sample sets be $S_1 = \{5, 7, 3, 3, 4\}$ and $S_2 = \{2, 4, 3, 2, 6\}$. Then, $C_{10} = 2$ (elements 5 and 7), $C_{01} = 1$ (element 6), $C_{11} = 1$ (element 4), $C_{02} = 1$ (element 2), $C_{21} = 1$ (element 3), and for remaining i, j's, $C_{ij} = 0$.

Lemma 17 *If there exists a symmetric algorithm $\mathcal{A}$ for testing a binary property of distributions defined by a permutation-invariant function, then there exist an algorithm for the same task that gets as input only the fingerprint of the sample that $\mathcal{A}$ takes.*

PROOF: Fix a canonical order for C_{ij}'s in the fingerprint of a sample. Let us define the following transformation on the sample: Relabel the elements such that the elements that appear exactly the same number of times from each distribution (i.e., the ones that contribute to a single C_{ij} in the fingerprint) have consecutive labels and the labels are grouped to conform to the canonical order of C_{ij}'s. Let us call this transformed sample the standard form of the sample. Since the algorithm $\mathcal{A}$ is symmetric and the property is defined by a permutation-invariant function, such a transformation does not affect the output of $\mathcal{A}$. So, we can further assume that we always present the sample to the algorithm in the standard form.

It is clear that given a sample, we can easily write down the fingerprint of the sample. Moreover, given the fingerprint of a sample, we can always construct a sample (S_1, S_2) in the standard form using the following algorithm: (1) Initialize S_1 and S_2 to be empty, and $e = 1$, (2) for every C_{ij} in the canonical order, and for $C_{ij} = k_{ij}$ times, include i and j copies of the element e in S_1 and S_2, respectively, then increment e. This algorithm shows a one-to-one and onto correspondence between all possible sample sets in the standard form and all possible $\{C_{ij}\}_{0\leq i,j\leq s}$ values.

Consider the algorithm $\mathcal{A}'$ that takes the fingerprint of a sample as input. Next, by using algorithm from above, algorithm $\mathcal{A}'$ constructs the sample in the standard form. Finally, $\mathcal{A}'$ outputs what $\mathcal{A}$ outputs on this sample. □

Remark 18 *Note that the definition of the fingerprint from Definition 16 can be generalized for a collection of k sample sets from k distributions for any k. An analogous lemma to Lemma 17 can be proven for testing algorithms for k-ary properties of distributions defined by a permutation-invariant function. We fixed $k = 2$ for ease of notation and because we will use this specific case later.*

3.4 A lower bound on sample complexity of testing closeness

In this section, we give a proof of a lower bound on the sample complexity of testing closeness in L_1 distance as a function of the size, denoted by n, of the domain of the distributions.

Theorem 19 *Given any algorithm using only $o(n^{2/3})$ samples from two discrete black-box distributions over $[n]$ for all sufficiently large n, there exist distributions $\mathbf{p}$ and $\mathbf{q}$ with L_1 distance 1 such that the algorithm will be unable to distinguish the case where one distribution is $\mathbf{p}$ and the other is $\mathbf{q}$ from the case where both distributions are $\mathbf{p}$.*

PROOF: By Lemma 15, we restrict our attention to symmetric algorithms. Fix a testing algorithm $\mathcal{A}$ that uses $o(n^{2/3})$ samples from each of the input distributions. Next, we define the distributions $\mathbf{p}$ and $\mathbf{q}$ from the theorem statement. Note that these distributions do not depend on $\mathcal{A}$.

Let us assume, without loss of generality, that n is a multiple of four and $n^{2/3}$ is an integer. We define the distributions $\mathbf{p}$ and $\mathbf{q}$ as follows: (1) For $1 \leq i \leq n^{2/3}$, $p_i = q_i = \frac{1}{2n^{2/3}}$. We call these elements the *heavy* elements. (2) For $n/2 < i \leq 3n/4$, $p_i = \frac{2}{n}$ and $q_i = 0$. We call these element the *light* elements of $\mathbf{p}$. (3) For $3n/4 < i \leq n$, $q_i = \frac{2}{n}$ and $p_i = 0$. We call these elements the *light* elements of $\mathbf{q}$. (4) For the remaining i's, $p_i = q_i = 0$.

The L_1 distance of $\mathbf{p}$ and $\mathbf{q}$ is 1. Now, consider the following two cases:

Case 1: The algorithm is given access to two black-box distributions: both of which output samples according to the distribution $\mathbf{p}$.

Case 2: The algorithm is given access to two black-box distributions: the first one outputs samples according to the distribution $\mathbf{p}$ and the second one outputs samples according to the distribution $\mathbf{q}$.

We show that a symmetric algorithm with sample complexity $o(n^{2/3})$ can not distinguish between these two cases. By Lemma 15, the theorem follows.

When restricted to the heavy elements, both distributions are identical. The only difference between $\mathbf{p}$ and $\mathbf{q}$ comes from the light elements, and the crux of the proof will be to show that this difference will not change the relevant statistics in a statistically significant way. We do this by showing that the only really relevant statistic is the number of elements that occur exactly once from each distribution. We then show that this statistic has a very similar distribution when generated by Case 1 and Case 2, because the expected number of such elements that are light is much less than the standard deviation of the number of such elements that are heavy.

We would like to have the frequency of each element be independent of the frequencies of the other elements. To achieve this, we assume that algorithm $\mathcal{A}$ first chooses two integers s_1 and s_2 independently from a Poisson distribution with the parameter $\lambda = s = o(n^{2/3})$. The Poisson distribution with the positive parameter λ has the probability mass function $p(k) = \exp(-\lambda)\lambda^k/k!$. Then, after taking s_1 samples from the first distribution and s_2 samples from the second distribution, $\mathcal{A}$ decides whether to accept or reject the distributions. In the following, we show that $\mathcal{A}$ cannot distinguish between Case 1 and Case 2 with success probability at least 2/3. Since both s_1 and s_2 will have values larger than $s/2$ with probability at least $1 - o(1)$ and we will show an upper bound on the statistical distance of the distributions of two random variables (i.e., the distributions on the samples), it will follow that no symmetric algorithm with sample complexity $s/2$ can.

Let F_i be the random variable corresponding to the number of times the element i appears in the sample from the first distribution. Define G_i analogously for the second distribution. It is well known that F_i is distributed identically to the Poisson distribution with parameter $\lambda = sr$, where r is the probability of element i (cf., Feller ([11], p. 216). Furthermore, it can also be shown that all F_i's are mutually independent. Thus, the total number of samples from the heavy elements and the total number of samples from the light elements are independent.

Recall the definition of the fingerprint of a sample from Section 3.3. The random variable C_{ij}, denotes the number of elements that appear exactly i times in the sample from the first distribution and exactly j times in the sample from the second distribution. For the rest of the proof, we shall assume that the algorithm is only given the fingerprint of the sample. The theorem follows by Lemma 17.

The proof will proceed by showing that the distributions on the fingerprint when the samples come from Case 1 or Case 2 are indistinguishable. The following lemma shows that with high probability, it is only the heavy elements that contribute to the random variables C_{ij} for $i + j \geq 3$.

Lemma 20 *(1) With probability $1 - o(1)$, at most $o(s)$ of the heavy elements appear at least three times in the combined sample from both distributions. (2) With probability $1 - o(1)$, none of the light elements appear at least three times in the combined sample from both distributions.*

Proof: Fix a heavy element i of probability $\frac{1}{2n^{2/3}}$. Recall that F_i and G_i denote the number of times this element appears from each distribution. The sum of the probabilities of the samples in which element i appears at most twice is

$$\rho = \exp(-s/n^{2/3})(1 + \frac{s}{n^{2/3}} + \frac{s^2}{2n^{4/3}}).$$

By using the approximation $e^{-x} = 1 - x + x^2/2$, we can show that $1 - \rho = O(s^3/n^2)$. By linearity of expectation, we expect to have $o(s)$ heavy elements that appear at least three times. For the light elements, an analogous argument shows that $o(1)$ light elements appear at least three times. The lemma follows by Markov's inequality. □

Let D_1 and D_2 be the distributions on all possible fingerprints when samples come from Case 1 and Case 2, respectively. The rest of the proof proceeds as follows. We first construct two processes T_1 and T_2 that generate distributions on fingerprints such that T_1 is statistically close to D_1 and T_2 is statistically close to D_2. Then, we prove that the distributions T_1 and T_2 are statistically close. Hence, the theorem follows by the indistinguishability of D_1 and D_2.

Each process has two phases. The first phase is the same in both processes. They randomly generate the frequency counts for each heavy element i using the random variables F_i and G_i defined above. The processes know which elements are heavy and which elements are light, although any distinguishing algorithm does not. This concludes the first phase of the processes.

In the second phase, process T_i determines the frequency counts of each light element according to Case i. If any light element is given a total frequency count of at least three during this step, the second phase of the process is restarted from scratch.

Since the frequency counts for all elements are determined at this point, both process output the fingerprint of the sample they have generated.

Lemma 21 *The output of T_1, viewed as a distribution, has L_1 distance $o(1)$ to D_1. The output of T_2, viewed as a distribution, has L_1 distance $o(1)$ to D_2.*

PROOF: The distribution that T_i generates is the distribution D_i conditioned on the event that all light elements appear at most twice in the combined sample. Since this conditioning holds true with probability at least $1 - o(1)$ by Lemma 20, $|T_i - D_i| \leq o(1)$. □

Lemma 22 $|T_1 - T_2| \leq 1/6$.

PROOF: By the generation process, the L_1 distance between T_1 and T_2 can only arise from the second phase. We show that the second phases of the processes do not generate an L_1 distance larger than 1/6.

For any variable C_{ij} of the fingerprint, the number of heavy elements that contribute to C_{ij} is independent of the number of light elements that contribute to C_{ij}. Let H be the random variable denoting the number of heavy elements that appear exactly once from each distribution. Let L be the random variable denoting the number of light elements that appear exactly once from each distribution. In Case 1, C_{11} is distributed identically to $H + L$, whereas, in Case 2, C_{11} is distributed identically to H.

Let $\mathcal{C} \stackrel{def}{=} \{C_{ij}\}_{i,j}$ and $\mathcal{C}^+ \stackrel{def}{=} \mathcal{C} \setminus \{C_{10}, C_{11}, C_{01}, C_{00}\}$. Since $\sum_{i,j} C_{ij} = n$, without loss of generality, we omit C_{00} in the rest of the discussion. Define $C_{1*} = \sum_j C_{1j}$ and $C_{*1} = \sum_i C_{i1}$. We use the notation $\Pr_{T_i}[\mathcal{C}']$ to denote the probability that T_i generates the random variable $\mathcal{C}'$ (defined on the fingerprint). We will use the fact that for any $\mathcal{C}^+, C_{1*}, C_{*1}$, $\Pr_{T_1}[\mathcal{C}^+, C_{1*}, C_{*1}] = \Pr_{T_2}[\mathcal{C}^+, C_{1*}, C_{*1}]$ in the following calculation. This fact follows from the conditioning that T_i generates on the respective D_i, namely, the condition that it is only the heavy elements that appear at least three times. Thus, only the heavy elements contribute to the variables C_{ij}, for $i + j \geq 3$, so the distribution on this part of the fingerprint is identical in both cases. The probability that a light element contributes to the random variable C_{20} conditioned on the event that it does not appear more than twice is exactly the probability that it appears twice from the first distribution. Therefore, C_{20} is also identically distributed (conditioned on C_{ij}'s for $i+j \geq 3$) in both cases by the fact that the contribution of the light elements to C_{20} is independent of that of the heavy elements. An analogous argument applies to C_{02}, C_{1*} and C_{*1}. So, we get

$$\begin{aligned}
|T_1 - T_2| &= \sum_{\mathcal{C}} |\Pr_{T_1}[\mathcal{C}] - \Pr_{T_2}[\mathcal{C}]| \\
&= \sum_{\mathcal{C}^+, C_{1*}, C_{*1}} \Pr_{T_1}[\mathcal{C}^+, C_{1*}, C_{*1}] \\
&\qquad \sum_{h,k,l \geq 0} |\Pr_{T_1}[(C_{11}, C_{10}, C_{01}) = (h,k,l) | \mathcal{C}^+, C_{1*}, C_{*1}] \\
&\qquad\quad - \Pr_{T_2}[(C_{11}, C_{10}, C_{01}) = (h,k,l) | \mathcal{C}^+, C_{1*}, C_{*1}]| \\
&= \sum_{\mathcal{C}^+, C_{1*}, C_{*1}} \Pr_{T_1}[\mathcal{C}^+, C_{1*}, C_{*1}] \\
&\qquad \sum_{h \geq 0} |\Pr_{T_1}[C_{11} = h | \mathcal{C}^+, C_{1*}, C_{*1}] - \Pr_{T_2}[C_{11} = h | \mathcal{C}^+, C_{1*}, C_{*1}]| \\
&= \sum_{h \geq 0} |\Pr[H = h] - \Pr[H + L = h]|
\end{aligned}$$

The third line follows since C_{10} and C_{01} are determined once $\mathcal{C}^+, C_{1*}, C_{*1}, C_{11}$ are determined. In the rest of the proof, we show that the fluctuations in H dominate the magnitude of L.

Let ξ_i be the indicator random variable that takes value 1 when element i appears exactly once from each distribution. Then, $H = \sum_{\text{heavy } i} \xi_i$. By the assumption about the way samples are generated, the ξ_i's are independent. Therefore, H is distributed identically to the binomial distribution on the sum of $n^{2/3}$ Bernoulli trials with success probability $\Pr[\xi_i = 1] = \exp(-s/n^{2/3})(s^2/4n^{4/3})$. An analogous argument shows that L is distributed identically to the binomial distribution with parameters $n/4$ and $\exp(-4s/n)(4s^2/n^2)$.

As n grows large enough, both H and L can be approximated well by normal distributions. That is,

$$\Pr[H = h] \to \frac{1}{\sqrt{2\pi}\mathrm{StDev}[H]} \exp(-(h - \mathrm{E}[H])^2/2\mathrm{Var}[H])$$

as $n \to \infty$. Therefore, by the independence of H and L, $H + L$ is also approximated well by a normal distribution.

Thus, $\Pr[H = h] = \Omega(1/\mathrm{StDev}[H])$ over an interval I_1 of length $\Omega(\mathrm{StDev}[H]) = \Omega(s/n^{1/3})$ centered at $\mathrm{E}[H]$. Similarly, $\Pr[H + L = h] = \Omega(1/\mathrm{StDev}[H + L])$ over an interval I_2 of length $\Omega(\mathrm{StDev}[H + L])$ centered at $\mathrm{E}[H + L]$. Since $\mathrm{E}[H + L] - \mathrm{E}[H] = \mathrm{E}[L] = O(s^2/n) = o(s/n^{1/3})$, $I_1 \cap I_2$ is an interval of length $\Omega(\mathrm{StDev}[H])$. Therefore,

$$\sum_{h \in I_1 \cap I_2} |\Pr[H = h] - \Pr[H + L = h]| \leq o(1)$$

because for $h \in I_1 \cap I_2$, $|\Pr[H = h] - \Pr[H + L = h]| = o(1/\mathrm{StDev}[H])$. We can conclude that $\sum_h |\Pr[H = h] - \Pr[H + L = h]|$ is less than $1/6$ after accounting for the probability mass of H and $H + L$ outside $I_1 \cap I_2$. □

The theorem follows by Lemma 21 and Lemma 22. □

By appropriately modifying the distributions $\vec{a}$ and $\vec{b}$ we can give a stronger version of Theorem 19 with a dependence on ϵ.

Corollary 23 *Given any test using only $o(n^{2/3}/\epsilon^{2/3})$ samples, there exist distributions $\vec{a}$ and $\vec{b}$ of L_1-distance ϵ such that the test will be unable to distinguish the case where one distribution is $\vec{a}$ and the other is $\vec{b}$ from the case where both distributions are $\vec{a}$.*

We can get a lower bound of $\Omega(\epsilon^{-2})$ for testing the L_2-Distance with a rather simple proof.

Theorem 24 *Given any test using only $o(\epsilon^{-2})$ samples, there exist distributions $\vec{a}$ and $\vec{b}$ of L_2-distance ϵ such that the test will be unable to distinguish the case where one distribution is $\vec{a}$ and the other is $\vec{b}$ from the case where both distributions are $\vec{a}$.*

PROOF: Let $n = 2$, $a_1 = a_2 = 1/2$ and $b_1 = 1/2 - \epsilon/\sqrt{2}$ and $b_2 = 1/2 + \epsilon/\sqrt{2}$. Distinguishing these distributions is exactly the question of distinguishing a fair coin from a coin of bias $\theta(\epsilon)$ which is well known to require $\theta(\epsilon^2)$ coin flips. □

The next theorem shows that learning a distribution using sublinear number of samples is not possible.

Theorem 25 *Suppose we have an algorithm that draws $o(n)$ samples from some unknown distribution $\vec{b}$ and outputs a distribution $\vec{c}$. There is some distribution $\vec{b}$ for which the output $\vec{c}$ is such that $\vec{b}$ and $\vec{c}$ have L_1-distance close to one.*

PROOF: (Sketch) Let A_S be the distribution that is uniform over $S \subseteq \{1, \ldots, n\}$. Pick S at random among sets of size $n/2$ and run the algorithm on A_S. The algorithm only learns $o(n)$ elements from S. So with high probability the L_1-distance of whatever distribution the algorithm output will have L_1-distance from A_S of nearly one. □

4 Application to Markov Chains

Random walks on Markov chains generate probability distributions over the states of the chain which are endpoints of a random walk. We employ L_1-Distance-Test , described in Section 3, to test mixing properties of Markov Chains.

Preliminaries/Notation Let $\mathbf{M}$ be a Markov chain represented by the transition probability matrix $\mathbf{M}$. The uth state of $\mathbf{M}$ corresponds to an n-vector $\vec{e}_u = (0, \ldots, 1, \ldots, 0)$, with a one in only the uth location and zeroes elsewhere. The distribution generated by t-step random walks starting at state u is denoted as a vector-matrix product $\vec{e}_u\mathbf{M}^t$.

Instead of computing such products in our algorithms, we assume that our L_1-Distance-Test has access to an oracle, `next_node` which on input of the state u responds with the state v with probability $\mathbf{M}(u, v)$. Given such an oracle, the distribution $\vec{e}_u^T\mathbf{M}^t$ can be generated in $O(t)$ steps. Furthermore, the oracle itself can be realized in $O(\log n)$ time per query, given linear preprocessing time to compute the cumulative sums $\mathbf{M}_c(j, k) = \sum_{i=1}^{k} \mathbf{M}(j, i)$. The oracle can be simulated on input u by producing a random number α in $[0, 1]$ and performing binary search over the uth row of $\mathbf{M}_c$ to find v such that $\mathbf{M}_c(u, v) \leq \alpha \leq \mathbf{M}_c(u, v+1)$. It then outputs state v. Note that when $\mathbf{M}$ is such that every row has at most d nonzero terms, slight modifications of this yield an $O(\log d)$ implementation consuming $O(n + m)$ words of memory if $\mathbf{M}$ is $n \times n$ and has m nonzero entries. Improvements of the work given in [30] can be used to prove that in fact constant query time is achievable with space consumption $O(n+m)$ for implementing `next_node` given linear preprocessing time.

We say that two states u and v are (ϵ, t)-*close* if the distribution generated by t-step random walks starting at u and v are within ϵ in the L_1 norm, *i.e.* $|\vec{e}_u\mathbf{M}^t - \vec{e}_v\mathbf{M}^t| < \epsilon$. Similarly we say that a state u and a distribution $\vec{s}$ are (ϵ, t)-close if $|\vec{e}_u\mathbf{M}^t - \vec{s}| < \epsilon$. We say $\mathbf{M}$ is (ϵ, t)-*mixing* if all states are (ϵ, t)-close to the same distribution:

Definition 26 *A Markov chain $\mathbf{M}$ is (ϵ, t)-*mixing *if a distribution $\vec{s}$ exists such that for all states u, $|\vec{e}_u\mathbf{M}^t - \vec{s}| \leq \epsilon$.*

For example, if $\mathbf{M}$ is $(\epsilon, O(\log n \log 1/\epsilon))$-mixing, then $\mathbf{M}$ is *rapidly-mixing* [28]. It can be easily seen that if $\mathbf{M}$ is (ϵ, t_0)-mixing then it is (ϵ, t) mixing for all $t > t_0$.

We now make the following definition:

Definition 27 *The* average t-step distribution, *$\vec{s}_{\mathbf{M},t}$ of a Markov chain $\mathbf{M}$ with n states is the distribution*

$$\vec{s}_{\mathbf{M},t} = \frac{1}{n}\sum_u \vec{e}_u\mathbf{M}^t.$$

This distribution can be easily generated by picking u uniformly from $[n]$ and walking t steps from state u. In an (ϵ, t)-mixing Markov chain, the average t-step distribution is ϵ-close to the stationary distribution. In a Markov chain that is not (ϵ, t)-mixing, this is not necessarily the case.

Each test given below assumes access to an L_1 distance tester L_1-Distance-Test(u, v, ϵ, δ) which given oracle access to distributions $\vec{e}_u, \vec{e}_v$ over the same n element set decides whether $|\vec{e}_u - \vec{e}_v| \leq f(\epsilon)$ or if $|\vec{e}_u - \vec{e}_v| > \epsilon$ with confidence $1 - \delta$. The time complexity of L_1`_test` is $T(n, \epsilon, \delta)$, and f is the *gap* of the tester. The implementation of L_1-Distance-Test given earlier in Section 3 has gap $f(\epsilon) = \epsilon/(4\sqrt{n})$, and time complexity $T = \tilde{O}(\frac{1}{\epsilon^4}n^{2/3}\log\frac{1}{\delta})$.

4.1 A test for mixing and a test for almost-mixing

We show how to decide if a Markov chain is (ϵ, t)-mixing; then we define and solve a natural relaxation of that problem.

In order to test that $\mathbf{M}$ is (ϵ, t)-mixing, one can use L_1-Distance-Test to compare each distribution $\vec{e}_u\mathbf{M}^t$ with $\vec{s}_{\mathbf{M},t}$, with error parameter ϵ and confidence δ/n. The running time is $O(nt \cdot T(n, \epsilon, \delta/n))$. If every state is $(f(\epsilon)/2, t)$-close to some distribution $\vec{s}$, then $\vec{s}_{\mathbf{M},t}$ is $f(\epsilon)/2$-close to $\vec{s}$. Therefore every state is (ϵ, t)-close to $\vec{s}_{\mathbf{M},t}$. On the other hand, if there is no distribution that is (ϵ, t)-close to all states, then, in particular, $\vec{s}_{\mathbf{M},t}$ is not (ϵ, t)-close to at least one state. We have shown

Theorem 28 *Let $\mathbf{M}$ be a Markov chain. Given L_1-Distance-Test with time complexity $T(n, \epsilon, \delta)$ and gap f and an oracle for* `next_node`*, there exists a test with time complexity $O(nt \cdot T(n, \epsilon, \delta/n))$ with the following behavior: If $\mathbf{M}$ is $(f(\epsilon)/2, t)$-mixing then* $\Pr[\mathbf{M}$ *passes*$] > 1 - \delta$*; if $\mathbf{M}$ is not (ϵ, t)-mixing then* $\Pr[\mathbf{M}$ *passes*$] < \delta$.

For the implementation of L_1-Distance-Test given in Section 3 the running time is $O(\frac{1}{\epsilon^4}n^{5/3}t\log n\log\frac{1}{\delta})$. It distinguishes between chains which are $\epsilon/(4\sqrt{n})$ mixing and those which are not ϵ-mixing. The running time is sublinear in the size of $\mathbf{M}$ if $t \in o(n^{1/3}/\log(n))$.

A relaxation of this procedure is testing that *most* starting states reach the same distribution after t steps. If $(1 - \rho)$ fraction of the states u of a given $\mathbf{M}$ satisfy $|\vec{s} - \vec{e}_u\mathbf{M}^t| \leq \epsilon$, then we say that $\mathbf{M}$ is (ρ, ϵ, t)-*almost mixing*. By picking $O(1/\rho \cdot \ln(/\delta))$ starting states uniformly at random, and testing their closeness to $\vec{s}_{\mathbf{M},t}$ we have:

Theorem 29 *Let* $\mathbf{M}$ *be a Markov chain. Given* L_1*-Distance-Test with time complexity* $T(n,\epsilon,\delta)$ *and gap* f *and an oracle for* `next_node`*, there exists a test with time complexity* $O(\frac{t}{\rho}T(n,\epsilon,\delta\rho)\log\frac{1}{\delta})$ *with the following behavior: If* $\mathbf{M}$ *is* $(\rho, f(\epsilon)/2, t)$*-almost mixing then* $\Pr[\mathbf{M} \text{ passes}] > 1-\delta$*; If* $\mathbf{M}$ *is not* (ρ,ϵ,t)*-almost mixing then* $\Pr[\mathbf{M} \text{ passes}] < \delta$.

4.2 A Property Tester for Mixing

The main result of this section is a test that determines if a Markov chain's matrix representation can be changed in an ϵ fraction of the non-zero entries to turn it into a $(4\epsilon, 2t)$-mixing Markov chain. This notion falls within the scope of property testing [26, 16, 17, 9, 25], which in general takes a set S with distance function Δ and a subset $P \subseteq S$ and decides if an elements $x \in S$ is in P or if it is far from every element in P, according to Δ. For the Markov chain problem, we take as our set S all matrices $\mathbf{M}$ of size $n \times n$ with at most d non-zero entries in each row. The distance function is given by the fraction of non-zero entries in which two matrices differ, and the difference in their average t-step distributions.

Definition 30 *Let* $\mathbf{M}_1$ *and* $\mathbf{M}_2$ *be* n*-state Markov chains with at most* d *non-zero entries in each row. Define distance function* $\Delta(\mathbf{M}_1,\mathbf{M}_2) = (\epsilon_1,\epsilon_2)$ *iff* $\mathbf{M}_1$ *and* $\mathbf{M}_2$ *differ on* $\epsilon_1 dn$ *entries and* $|\vec{s}_{\mathbf{M}_1,t} - \vec{s}_{\mathbf{M}_2,t}| = \epsilon_2$*. We say that* $\mathbf{M}_1$ *and* $\mathbf{M}_2$ *are* (ϵ_1,ϵ_2)*-close if* $\Delta(\mathbf{M}_1,\mathbf{M}_2) \le (\epsilon_1,\epsilon_2)$.[1]

A natural question is whether all Markov chains are ϵ-close to an (ϵ,t)-mixing Markov chain, for certain parameters of ϵ. For constant ϵ and $t = O(\log n)$, one can show that every strongly-connected Markov chain is $(\epsilon,1)$-close to another Markov chain which (ϵ,t)-mixes. However, the situation changes when asking whether there is an (ϵ,t)-mixing Markov chain that is close both in the matrix representation and in the average t-step distribution: specifically, it can be shown that there exist constants $\epsilon,\epsilon_1,\epsilon_2 < 1$ and Markov chain $\mathbf{M}$ for which no Markov chain is both (ϵ_1,ϵ_2)-close to $\mathbf{M}$ and $(\epsilon,\log n)$-mixing. In fact, when ϵ_1 is small enough, the problem becomes nontrivial even for $\epsilon_2 = 1$. The Markov chain corresponding to random walks on the n-cycle provides an example which is not $(t^{-1/2},1)$-close to any (ϵ,t)-mixing Markov chain.

Motivation As before, our algorithm proceeds by taking random walks on the Markov chain and comparing final distributions by using the L_1 distance tester. We define three types of states. First a *normal* state is one from which a random walk arrives at nearly the average t-step distribution. In the discussion which follows, t and ϵ denote constant parameters fixed as input to the algorithm `TestMixing`.

Definition 31 *Given a Markov Chain* $\mathbf{M}$*, a state* u *of the chain is* normal *if it is* (ϵ,t)*-close to* $\vec{s}_{\mathbf{M},t}$*. That is if* $|\vec{e}_u\mathbf{M}^t - \vec{s}_{\mathbf{M},t}| \le \epsilon$*. A state is* bad *if it is not normal.*

Testing normality requires time $O(t \cdot T(n,\epsilon,\delta))$. Using this definition the first two algorithms given in this section can be described as testing whether all (*resp.* most) states in $\mathbf{M}$ are *normal*. Additionally, we need to distinguish states which not only produce random walks which arrive near $\vec{s}_{\mathbf{M},t}$ but which have low probability of visiting a bad state. We call such states *smooth* states:

Definition 32 *A state* $\vec{e}_u$ *in a Markov chain* $\mathbf{M}$ *is* smooth *if* (a) u *is* (ϵ,τ)*-close to* $\vec{s}_{\mathbf{M},t}$ *for* $\tau = t,\ldots,2t$ *and* (b) *the probability that a* $2t$*-step random walk starting at* $\vec{e}_u$ *visits a bad state is at most* ϵ.

Testing smoothness of a state requires $O(t^2 \cdot T(n,\epsilon,\delta))$ time. Our property test merely verifies by random sampling that most states are smooth.

[1] We say $(x,y) \le (a,b)$ iff $x \le a$ and $y \le b$

The test Figure 3 gives an algorithm which on input Markov chain $\mathbf{M}$ and parameter ϵ determines whether at least $(1-\epsilon)$ fraction of the states of $\mathbf{M}$ are smooth according to two distributions: uniform and the average t-step distribution. Assuming access to L_1-Distance-Test with complexity $T(n,\epsilon,\delta)$, this test runs in time $O(\epsilon^{-2}t^2T(n,\epsilon,\frac{1}{6t}))$.

```
TestMixing(M, t, ε)
Let k = Θ(1/ε)
Select k states u_1, ..., u_k uniformly
Select k states u_{k+1}, ..., u_{2k} according to s_{M,t}
For i = 1 to 2k
  u = e_{u_i}
  For w = 1 to O(1/ε)
    For j = 1 to 2t
      u = next_node(M, u)
      L_1-Distance-Test(e_u M^t, s_{M,t}, ε, 1/(6t))
    End
  End
  For τ = t to 2t
    L_1-Distance-Test(e_{u_i} M^τ, s_{M,t}, ε, 1/(3t))
  End
Pass if all tests pass
```

Figure 3: Algorithm TestMixing

The main lemma of this section says that any Markov chain which passes our test is $(2\epsilon, 1.01\epsilon)$-close to a $(4\epsilon, 2t)$-mixing Markov chain. First we give the modification

Definition 33 *F is a function from $n \times n$ matrices to $n \times n$ matrices such that $F(\mathbf{M})$ returns $\widetilde{\mathbf{M}}$ by modifying the rows corresponding to bad states of $\mathbf{M}$ to $\vec{e}_u$ where u is a smooth state.*

An important feature of the transformation F is that it does not affect the distribution of random walks originating from smooth states very much.

Lemma 34 *Given a Markov chain $\mathbf{M}$ and any state $u \in M$ which is smooth. If $\widetilde{\mathbf{M}} = F(\mathbf{M})$ then for any time $t \le \tau \le 2t$, $|\vec{e}_u\mathbf{M}^\tau - \vec{e}_u\widetilde{\mathbf{M}}^\tau| \le \epsilon$ and $|\vec{s}_{\mathbf{M},t} - \vec{e}_u\widetilde{\mathbf{M}}^\tau| \le 2\epsilon$.*

Proof: Define Γ as the set of all walks of length τ from u in $\mathbf{M}$. Partition Γ into Γ_B and $\bar{\Gamma}_B$ where Γ_B is the subset of walks which visit a bad state. Let $\chi_{w,i}$ be an indicator function which equals 1 if walk w ends at state i, and 0 otherwise. Let weight function $W(w)$ be defined as the probability that walk w occurs. Finally define the primed counterparts Γ', etc. for the Markov chain $\widetilde{\mathbf{M}}$. Now the ith element of $\vec{e}_u\mathbf{M}^\tau$ is $\sum_{w\in\Gamma_B}\chi_{w,i}\cdot W(w) + \sum_{w\in\bar{\Gamma}_B}\chi_{w,i}\cdot W(w)$. A similar expression can be written for each element of $\vec{e}_u\widetilde{\mathbf{M}}^\tau$. Since $W(w) = W'(w)$ whenever $w \in \bar{\Gamma}_B$ it follows that $|\vec{e}_u\mathbf{M}^\tau - \vec{e}_u\widetilde{\mathbf{M}}^\tau| \le \sum_i\sum_{w\in\Gamma_B}\chi_{w,i}|W(w)-W'(w)| \le \sum_i\sum_{w\in\Gamma_B}\chi_{w,i}W(w) \le \epsilon$.

Additionally, since $|\vec{s}_{\mathbf{M},t} - \vec{e}_u\mathbf{M}^\tau| \le \epsilon$ by the definition of smooth, it follows that $|\vec{s}_{\mathbf{M},t} - \vec{e}_u\widetilde{\mathbf{M}}^\tau| \le |\vec{s}_{\mathbf{M},t} - \vec{e}_u\mathbf{M}^\tau| + |\vec{e}_u\mathbf{M}^\tau - \vec{e}_u\widetilde{\mathbf{M}}^\tau| \le 2\epsilon$. □

We can now prove the main lemma:

Lemma 35 *If according to both the uniform distribution and the distribution $\vec{s}_{\mathbf{M},t}$, $(1-\epsilon)$ fraction of the states of a Markov chain $\mathbf{M}$ are smooth, then the matrix $\mathbf{M}$ is $(2\epsilon, 1.01\epsilon)$-close to a matrix $\widetilde{\mathbf{M}}$ which is $(4\epsilon, 2t)$-mixing.*

Proof: Let $\widetilde{\mathbf{M}} = F(\mathbf{M})$. $\widetilde{\mathbf{M}}$ and $\mathbf{M}$ differ on at most $\epsilon n(d+1)$ entries. This gives the first part of our distance bound. For the second we analyze $|\vec{s}_{\mathbf{M},t} - \vec{s}_{\widetilde{\mathbf{M}},t}| = \frac{1}{n}\sum_u |\vec{e}_u \mathbf{M}^t - \vec{e}_u \widetilde{\mathbf{M}}^t|$ as follows. The sum is split into two parts, over the nodes which are smooth and those nodes which are not. For each of the smooth nodes u, Lemma 34 says that $|\vec{e}_u \mathbf{M}^t - \vec{e}_u \widetilde{\mathbf{M}}^t| \leq \epsilon$. Nodes which are not smooth account for at most ϵ fraction of the nodes in the sum, and thus can contribute no more than ϵ absolute weight to the distribution $\vec{s}_{\widetilde{\mathbf{M}},t}$. The sum can be bounded now by $|\vec{s}_{\mathbf{M},t} - \vec{s}_{\widetilde{\mathbf{M}},t}| \leq \frac{1}{n}((1-\epsilon)n\epsilon + \epsilon n) \leq 2\epsilon$.

In order to show that $\widetilde{\mathbf{M}}$ is $(4\epsilon, 2t)$-mixing, we prove that for every state u, $|\vec{s}_{\mathbf{M},t} - \vec{e}_u \mathbf{M}^{2t}| \leq 4\epsilon$. The proof considers three cases: u smooth, u bad, and u normal. The last case is the most involved.

If u is smooth in the Markov chain $\mathbf{M}$, then Lemma 34 immediately tells us that $|\vec{s}_{\mathbf{M},t} - \vec{e}_u \widetilde{\mathbf{M}}^{2t}| \leq 2\epsilon$. Similarly if u is bad in the Markov chain $\mathbf{M}$, then in the chain $\widetilde{\mathbf{M}}$ any path starting at u transitions to a smooth state v in one step. Since $|\vec{s}_{\mathbf{M},t} - \vec{e}_v \widetilde{\mathbf{M}}^{2t-1}| \leq 2\epsilon$ by Lemma 34, the desired bound follows.

If $\vec{e}_u$ is a normal state which is not smooth we need a more involved analysis of the distribution $|\vec{e}_u \widetilde{\mathbf{M}}^{2t}|$. We divide Γ, the set of all $2t$-step walks in $\mathbf{M}$ starting at u, into three sets, which we consider separately.

For the first set take $\Gamma_B \subseteq \Gamma$ to be the set of walks which visit a bad node before time t. Let $\vec{d_b}$ be the distribution over endpoints of these walks, that is, let $\vec{d_b}$ assign to state i the probability that any walk $w \in \Gamma_B$ ends at state i. Let $w \in \Gamma_B$ be any such walk. If w visits a bad state at time $\tau < t$, then in the new Markov chain $\widetilde{\mathbf{M}}$, w visits a smooth state v at time $\tau + 1$. Another application of Lemma 34 implies that $|\vec{e}_v \widetilde{\mathbf{M}}^{2t-\tau-1} - \vec{s}_{\mathbf{M},t}| \leq 2\epsilon$. Since this is true for all walks $w \in \Gamma_B$, we find $|\vec{d_b} - \vec{s}_{\mathbf{M},t}| \leq 2\epsilon$.

For the second set, let $\Gamma_S \subseteq \Gamma \setminus \Gamma_B$ be the set of walks not in Γ_B which visit a smooth state at time t. Let $\vec{d_s}$ be the distribution over endpoints of these walks. Any walk $w \in \Gamma_S$ is identical in the chains $\mathbf{M}$ and $\widetilde{\mathbf{M}}$ up to time t, and then in the chain $\widetilde{\mathbf{M}}$ visits a smooth state v at time t. Thus since $|\vec{e}_v \widetilde{\mathbf{M}}^t - \vec{s}_{\mathbf{M},t}| \leq 2\epsilon$, we have $|\vec{d_s} - \vec{s}_{\mathbf{M},t}| \leq 2\epsilon$.

Finally let $\Gamma_N = \Gamma \setminus (\Gamma_B \cup \Gamma_S)$, and let $\vec{d_n}$ be the distribution over endpoints of walks in Γ_N. Γ_N consists of a subset of the walks from a normal node u which do not visit a smooth node at time t. By the definition of normal, u is (ϵ, t)-close to $\vec{s}_{\mathbf{M},t}$ in the Markov chain $\mathbf{M}$. By assumption at most ϵ weight of $\vec{s}_{\mathbf{M},t}$ is assigned to nodes which are not smooth. Therefore $|\Gamma_N|/|\Gamma|$ is at most $\epsilon + \epsilon = 2\epsilon$.

Now define the weights of these distributions as ω_b, ω_s and ω_n. That is ω_b is the probability that a walk from u in $\mathbf{M}$ visits a bad state before time t. Similarly ω_s is the probability that a walk does not visit a bad state before time t, but visits a smooth state at time t, and ω_n is the probability that a walk does not visit a bad state but visits a normal, non-smooth state at time t. Then $\omega_b + \omega_s + \omega_n = 1$. Finally $|\vec{e}_u \widetilde{\mathbf{M}}^{2t} - \vec{s}_{\mathbf{M},t}| = |\omega_b \vec{d_b} + \omega_s \vec{d_s} + \omega_n \vec{d_n} - \vec{s}_{\mathbf{M},t}| \leq \omega_b |\vec{d_b} - \vec{s}_{\mathbf{M},t}| + \omega_s |\vec{d_s} - \vec{s}_{\mathbf{M},t}| + \omega_n |\vec{d_n} - \vec{s}_{\mathbf{M},t}| \leq (\omega_b + \omega_s)\max\{|\vec{d_b} - \vec{s}_{\mathbf{M},t}|, |\vec{d_s} - \vec{s}_{\mathbf{M},t}|\} + \omega_n |\vec{d_n} - \vec{s}_{\mathbf{M},t}| \leq 4\epsilon$. □

Given this, we finally can show our main theorem:

Theorem 36 *Let* $\mathbf{M}$ *be a Markov chain. Given* L_1*-Distance-Test with time complexity* $T(n, \epsilon, \delta)$ *and gap* f *and an oracle for* `next_node`*, there exists a test such that if* $\mathbf{M}$ *is* $(f(\epsilon), t)$*-mixing then the test passes with probability at least* $2/3$*. If* $\mathbf{M}$ *is not* $(2\epsilon, 1.01\epsilon)$*-close to any* $\widetilde{\mathbf{M}}$ *which is* $(4\epsilon, 2t)$*-mixing then the test fails with probability at least* $2/3$*. The runtime of the test is* $O(\frac{1}{\epsilon^2} \cdot t^2 \cdot T(n, \epsilon, \frac{1}{6t}))$*.*

Proof: Since in any Markov chain $\mathbf{M}$ which is (ϵ, t)-mixing all states are smooth, $\mathbf{M}$ passes this test with probability at least $(1-\delta)$. Furthermore, any Markov chain with at least $(1-\epsilon)$ fraction of smooth states is $(2\epsilon, 1.01\epsilon)$-close to a Markov chain which is $(4\epsilon, 2t)$-mixing, by Lemma 35. □

4.3 Extension to sparse graphs and uniform distributions

The property test can also be made to work for general sparse Markov chains by a simple modification to the testing algorithms. Consider Markov chains with at most $m << n^2$ nonzero entries, but with no nontrivial bound on the number of nonzero entries per row. Then the definition of the distance should be modified to $\Delta(M_1, M_2) = (\epsilon_1, \epsilon_2)$ if M_1 and M_2 differ on $\epsilon_1 \cdot m$ entries and the $\vec{s}_{\mathbf{M}_1,t} - \vec{s}_{\mathbf{M}_2,t} = \epsilon_2$. The above test does not suffice for testing that $\mathbf{M}$ is (ϵ_1, ϵ_2)-close to an (ϵ, t)-mixing Markov chain $\widetilde{M}$, since in our proof, the rows corresponding to bad states may have many nonzero entries and thus $\mathbf{M}$ and $\widetilde{M}$ may differ in a large fraction of the nonzero entries. However, let D be a distribution on states in which the probability of each state is proportional to cardinality of the support set of its row. Natural ways of encoding this Markov chain allow constant time generation of states according to D. By modifying the test in Figure 3 to also test that most states according to D are smooth, one can show that $\mathbf{M}$ is close to an (ϵ, t)-mixing Markov chain $\widetilde{M}$.

Because of our ability to test ϵ-closeness to the *uniform* distribution in $O(n^{1/2}\epsilon^{-2})$ steps [18], it is possible to speed up our test for mixing for those Markov chains known to have uniform stationary distribution, such as Markov chains corresponding to random walks on regular graphs. An ergodic random walk on the vertices of an undirected graph instead may be regarded (by looking at it "at times $t + 1/2$") as a random walk on the *edge-midpoints* of that graph. The stationary distribution on edge-midpoints always exists and is uniform. So, for undirected graphs we can speed up mixing testing by using a tester for closeness to uniform distribution.

5 Further Research

It would be interesting to study these questions for other difference measures. For example, the Kullback-Leibler asymmetric "distance" from Information Theory defined as

$$\mathrm{KLdist}(\vec{p}, \vec{q}) = \sum_i p_i \ln \frac{p_i}{q_i}$$

measures the relative entropy between two distributions. However, small changes to the distribution can cause great changes in the Kullback-Leibler distance making distinguishing the cases impossible.

Perhaps some variation of Kullback-Leibler distance might lead to more interesting results. For example, consider the following distance formula

$$\mathrm{NPdist}(\vec{p}, \vec{q}) = \mathrm{KLdist}(\vec{p}, \frac{\vec{p} + \vec{q}}{2}) + \mathrm{KLdist}(\vec{q}, \frac{\vec{p} + \vec{q}}{2}).$$

Although it is not a true metric (it does not obey trangle inequailty), it has constant value if $\vec{p}$ and $\vec{q}$ have disjoint support and cannot increase if we use the same Markov chain transition of $\vec{p}$ and $\vec{q}$. We suspect NPdist is in some sense "most powerful" for the purpose of deciding whether $\vec{p} \neq \vec{q}$.

Russell Impagliazzo also suggests considering weighted differences, i.e., estimating $\|\vec{p} - \vec{q}\| / \max(\|\vec{p}\|, \|\vec{q}\|)$ for various norms like the L_2-norm.

Suppose instead of having two unknown distributions, we have only one distribution to sample and we want to know whether it is close to some known fixed distribution D. If D is the uniform distribution, Goldreich and Ron [18] give a tight bound of $\theta(\sqrt{n})$. For other D the question remains open. Our $\Omega(n^{2/3})$ lower bound proof does not apply.

What if our samples are not fully independent? Our upper bound works even if the samples are only four-way independent. How do our bounds increase if we lack even that much independence?

Finally our lower and upper bounds do not precisely match. Can we get tighter bounds with better analysis or do we need new variations on our tests and/or counterexamples?

Smith [29] has some improved bounds and additional applications of the results in this paper.

Acknowledgments We are very grateful to Oded Goldreich and Dana Ron for sharing an early draft of their work with us and for several helpful discussions. We would also like to thank Naoke Abe, Richard Beigel, Yoav Freund, Russell Impagliazzo, Alexis Maciel, Sofya Raskhodnikova, and Tassos Viglas for helpful discussions. Finally, we thank Ning Xie for pointing out two errors in the proofs in an earlier version.

References

[1] N. Alon, M. Krivelevich, E. Fischer, and M. Szegedy. Efficient testing of large graphs. In IEEE, editor, *40th Annual Symposium on Foundations of Computer Science: October 17–19, 1999, New York City, New York,*, pages 656–666, 1109 Spring Street, Suite 300, Silver Spring, MD 20910, USA, 1999. IEEE Computer Society Press.

[2] N. Alon, Y. Matias, and M. Szegedy. The space complexity of approximating the frequency moments. *JCSS*, 58, 1999.

[3] Noga Alon. Eigenvalues and expanders. *Combinatorica*, 6(2):83–96, 1986.

[4] Ziv Bar-Yossef, Ravi Kumar, and D. Sivakumar. Sampling algorithms: Lower bounds and applications. In *Proceedings of 33th Symposium on Theory of Computing*, Crete, Greece, 6–8 July 2001. ACM.

[5] A. Broder, M. Charikar, A. Frieze, and M. Mitzenmacher. Min-wise independent permutations. *JCSS*, 60, 2000.

[6] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory.* Wiley Series in Telecommunications. John Wiley & Sons, 1991.

[7] N. Cressie and P.B. Morgan. Design considerations for Neyman Pearson and Wald hypothesis testing. *Metrika*, 36(6):317–325, 1989.

[8] I. Csiszár. Information-type measures of difference of probability distributions and indirect observations. *Studia Scientiarum Mathematicarum Hungarica*, 1967.

[9] Funda Ergün, Sampath Kannan, S. Ravi Kumar, Ronitt Rubinfeld, and Mahesh Viswanathan. Spot-checkers. In *STOC 30*, pages 259–268, 1998.

[10] J. Feigenbaum, S. Kannan, M. Strauss, and M. Viswanathan. An approximate L^1-difference algorithm for massive data streams (extended abstract). In *FOCS 40*, 1999.

[11] William Feller. *An Introduction to Probability Theory and Applications*, volume 1. John Wiley & Sons Publishers, New York, NY, 3rd ed., 1968.

[12] J. Fong and M. Strauss. An approximate L^p-difference algorithm for massive data streams. In *Annual Symposium on Theoretical Aspects of Computer Science*, 2000.

[13] Alan Frieze and Ravi Kannan. Quick approximation to matrices and applications. *COMBINAT: Combinatorica*, 19, 1999.

[14] Phillip B. Gibbons and Yossi Matias. Synopsis data structures for massive data sets. In *SODA 10*, pages 909–910. ACM-SIAM, 1999.

[15] O. Goldreich and L. Trevisan. Three theorems regarding testing graph properties. Technical Report ECCC-10, Electronic Colloquium on Computational Complexity, January 2001.

[16] Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. In *FOCS 37*, pages 339–348. IEEE, 14–16 October 1996.

[17] Oded Goldreich and Dana Ron. Property testing in bounded degree graphs. In *STOC 29*, pages 406–415, 1997.

[18] Oded Goldreich and Dana Ron. On testing expansion in bounded-degree graphs. Technical Report TR00-020, Electronic Colloquium on Computational Complexity, 2000.

[19] G. H. Golub and C. F. van Loan. *Matrix Computations*. The John Hopkins University Press.

[20] R. Kannan. Markov chains and polynomial time algorithms. In Shafi Goldwasser, editor, *Proceedings: 35th Annual Symposium on Foundations of Computer Science, November 20–22, 1994, Santa Fe, New Mexico*, pages 656–671, 1109 Spring Street, Suite 300, Silver Spring, MD 20910, USA, 1994. IEEE Computer Society Press.

[21] Sampath Kannan and Andrew Chi-Chih Yao. Program checkers for probability generation. In Javier Leach Albert, Burkhard Monien, and Mario Rodríguez-Artalejo, editors, *ICALP 18*, volume 510 of *Lecture Notes in Computer Science*, pages 163–173, Madrid, Spain, 8–12 July 1991. Springer-Verlag.

[22] E. L. Lehmann. *Testing Statistical Hypotheses*. Wadsworth and Brooks/Cole, Pacific Grove, CA, second edition, 1986. [Formerly New York: Wiley].

[23] J. Neyman and E.S. Pearson. On the problem of the most efficient test of statistical hypotheses. *Philos. Trans. Royal Soc. A*, 231:289–337, 1933.

[24] Beresford N. Parlett. *The Symmetric Eigenvalue Problem*, volume 20 of *Classics in applied mathematics*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1998.

[25] Michal Parnas and Dana Ron. Testing the diameter of graphs. In Dorit Hochbaum, Klaus Jensen, José D.P. Rolim, and Alistair Sinclair, editors, *Randomization, Approximation, and Combinatorial Optimization*, volume 1671 of *Lecture Notes in Computer Science*, pages 85–96. Springer-Verlag, 8–11 August 1999.

[26] Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2):252–271, April 1996.

[27] Amit Sahai and Salil Vadhan. A complete promise problem for statistical zero-knowledge. In *Proceedings of the 38th Annual Symposium on the Foundations of Computer Science*, pages 448–457. IEEE, 20–22 October 1997.

[28] Alistair Sinclair and Mark Jerrum. Approximate counting, uniform generation and rapidly mixing Markov chains. *Information and Computation*, 82(1):93–133, July 1989.

[29] Warren D. Smith. Testing if distributions are close via sampling. Technical Report Available as Report #56, NECI, 2000. http://www.neci.nj.nec.com/homepages/wds/works.html.

[30] A. J. Walker. An efficient method for generating discrete random variables with general distributions. *ACM trans. math. software*, 3:253–256, 1977.

[31] Kenji Yamanishi. Probably almost discriminative learning. *Machine Learning*, 18(1):23–50, 1995.