Testing that distributions are close

Tuğkan Batu*
Computer Science Department
Cornell University
Ithaca, NY 14853
batu@cs.cornell.edu

Lance Fortnow
NEC Research Institute
4 Independence Way
Princeton, NJ 08540
fortnow@research.nj.nec.com

Ronitt Rubinfeld
NEC Research Institute
4 Independence Way
Princeton, NJ 08540
ronitt@research.nj.nec.com

Warren D. Smith
NEC Research Institute
4 Independence Way
Princeton, NJ 08540
wds@research.nj.nec.com

Patrick White*
Computer Science Department
Cornell University
Ithaca, NY 14853
white@cs.cornell.edu

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.

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 14 in Section 3.3 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-

*This work was partially supported by ONR N00014-97-1-0505, MURI, NSF Career grant CCR-9624552, and an Alfred P. Sloan Research Award.

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 [22, 13, 14, 7, 21]. A related work of Kannan and Yao [17] 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. [11, 2, 8, 9]). Another line of work [3] estimates the distance in frequency count distributions on words between various documents, where again space is limited.

In an interactive setting, Sahai and Vadhan [23] 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 [4, 19, 5, 6, 18]. None of these give sublinear bounds in the domain size for our problem. The specific model of singleton hypothesis classes is studied by Yamanishi [27].

Goldreich and Ron [12] 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 [24], 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 [14] show that testing that a graph is close to an expander requires $\Omega(n^{1/2})$ queries.

The conductance [24] of a graph is known to be closely related to expansion and rapid-mixing properties of the graph [16][24]. Frieze and Kannan [10] 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 [24, 16] is related to the separation between the two largest eigenvalues [1]. 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 [15]. However, for a sparse $n \times n$ *symmetric* matrix with m nonzero entries, $n \leq m$, "Lanczos algorithms" [20] 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.

```
L2-Distance-Test(p, q, m, ε, δ)
Repeat O(log(1/δ)) times
  Let F_p = a set of m samples from p
  Let F_q = a set of m samples from 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 p
  Let Q_q = a set of m samples from q
    Let s_pq be the number of collisions
      between Q_p and Q_q.
  Let r = (2m/(m-1))(r_p + r_q)
  Let s = 2 s_pq
  If r - s > m^2 ε^2/2 then reject
Reject if the majority of iterations reject,
accept otherwise
```

Figure 1. Algorithm L_2-Distance-Test

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 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.

```
L_1-Distance-Test(p, q, ε, δ)
Sample p⃗ and q⃗ for
  M = O(max(ε^{-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 − ε/63) M n^{-2/3} times
If S_p and S_q are empty
    L_2-Distance-Test(p, q, O(n^{2/3}/ε^4), ε/(2√n), δ/2)
else
  ℓ_i^p = # times element i appears in S_p
  ℓ_i^q = # times element i appears in S_q
  Fail if Σ_i |ℓ_i^p − ℓ_i^q| > εM/8.
  Define p⃗' as follows:
    sample an element from p⃗
    if this sample is not in S_p output it,
    otherwise output an x ∈_R [n].
  Define q⃗' similarly.
  L_2-Distance-Test(p', q', O(n^{2/3}/ε^4), ε/(2√n), δ/2)
```

Figure 2. Algorithm L_1-Distance-Test

In Section 3.3 we prove that $\Omega(n^{2/3})$ samples are 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 $\|\vec{p} - \vec{q}\|^2$.

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

To analyze the last expectation, we use two facts. First, it is easy to see, by the definition of covariance, that $\mathrm{E}\left[\bar{C}_{i,j}\bar{C}_{k,l}\right] \leq \mathrm{E}[C_{i,j}C_{k,l}]$. 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

$$
\begin{aligned}
&\mathrm{E}\left[\sum_{\substack{(i,j),(k,l) \in F \times F \\ (i,j) \neq (k,l)}} \bar{C}_{i,j}\bar{C}_{k,l}\right] \\
&= \mathrm{E}\left[\sum_{\substack{(i,j),(i,l) \in F \times F \\ j \neq l}} \bar{C}_{i,j}\bar{C}_{i,l} + \sum_{\substack{(i,j),(k,j) \in F \times F \\ i \neq k}} \bar{C}_{i,j}\bar{C}_{k,j}\right] \\
&\leq \mathrm{E}\left[\sum_{\substack{(i,j),(i,l) \in F \times F \\ j \neq l}} C_{i,j}C_{i,l} + \sum_{\substack{(i,j),(k,j) \in F \times F \\ i \neq k}} C_{i,j}C_{k,j}\right]
\end{aligned}
$$

$$\leq \quad cm^3 \sum_{\ell \in [n]} p_\ell q_\ell^2 + p_\ell^2 q_\ell \leq cm^3 b^2 \sum_{\ell \in [n]} q_\ell \leq cm^3 b^2$$

for some constant c. In order to bound $\mathrm{Var}\,[r-s]$ we use Lemma 4. Since $\mathrm{Var}\,[r] \leq cm^2 b$ and the variance is additive for independent random variables, we can write $\mathrm{Var}\,[r-s] \leq c(m^3 b^2 + m^2 b)$. □

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 $\|\vec{p} - \vec{q}\| \leq \epsilon/2$, then L_2-**Distance-Test**$(p, q, m, \epsilon, \delta)$ passes with probability at least $1 - \delta$. If $\|\vec{p} - \vec{q}\| > \epsilon$ then L_2-**Distance-Test**$(p, q, m, \epsilon, \delta)$ 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,

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

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

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

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 $p_i < \epsilon^2 n^{-2/3}$, $\bar{p}_i < (1 - \epsilon/63) n^{-2/3}$.*

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 $\epsilon \leq 1/2$, $3\epsilon^2 \leq (1 - \epsilon/63)$, 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 $\epsilon/9$. 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. $\square$

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.

3.3. Lower Bounds

Theorem 10 *Given any test using only $o(n^{2/3})$ samples, there exist distributions $\vec{a}$ and $\vec{b}$ of L_1-distance 1 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: Fix a testing algorithm that uses $s = o(n^{2/3})$ samples. Without loss of generality we assume that algorithm is symmetric, i.e., given two distributions the algorithm will give the same result for any permutation of the underlying space. Otherwise we could permute the sample space to maximize the error of the testing algorithm; the result (including this pre-permutation) *would* be a symmetric algorithm, and it would have the same failure probability on worst-case input.

Let us assume that n is a multiple of four. We define two distributions $\vec{a}$ and $\vec{b}$ as follows: (1) For $1 \le i \le n^{2/3}$, $a_i = b_i = \frac{1}{2n^{2/3}}$. We call these the heavy elements. (2) For $n/2 < i \le 3n/4$, $a_i = \frac{2}{n}$ and $b_i = 0$. We call these the light elements of $\vec{a}$. (3) For $3n/4 < i \le n$, $b_i = \frac{2}{n}$ and $a_i = 0$. We call these the light elements of $\vec{b}$. (4) For the remaining i, $a_i = b_i = 0$.

The L_1-distance of $\vec{a}$ and $\vec{b}$ is one. We will show that no symmetric algorithm can distinguish the two.

Lemma 11 *(1) With high probability, at most $o(n^{2/3})$ of the heavy elements occur more than twice in the sample space of both distributions combined. (2) With high probability, none of the light elements occur more than twice in the same space of both distributions.*

PROOF: For a fixed heavy element of probability $p = \frac{1}{2n^{2/3}}$ the probability that it appears at least three times is bounded by $s^3p^3 = o(1)$, i.e., that is roughly s^3 possible triples each of which are all equal to our element with probability p^3. By linearity of expectation we have $o(n^{2/3})$ high probability elements occurring three times. For the light elements the same argument gives $o(1)$ low probability elements occurring three times. $\square$

The elements which occur three or more times occur only on the heavy elements which have the same probability in each distribution. So these cannot help the algorithm distinguish the distributions. Let H be the random variable denoting the number of collisions among the heavy elements. Let L be the random variable denoting the number of collisions among the light elements. If the algorithm was given distributions $\vec{a}$ and $\vec{b}$ the number of collisions it would see between them would be H. If the algorithm was given the same distribution $\vec{a}$ twice the number of collisions would be the random variable $H + L$. The only relevant test a symmetric algorithm can make is to determine whether the number of collisions between the distributions comes from H or $H + L$.

The expected value of H is $s^2/2n^{2/3}$. The variance is $\theta(s^2/n^{2/3} + s^3/n^{4/3}) = \theta(s^2/n^{2/3})$ since $s = o(n^{2/3})$. The standard deviation of H is $\sqrt{\theta(s^2/n^{2/3})} = \theta(s/n^{1/3})$. The expected value and variance of L is $\theta(s^2/n) = o(s/n^{1/3})$.

Since the expected value and variance of L are swamped by the standard deviation of H and one would expect it is impossible to distinguish between samples drawn from H versus $H + L$. To see this we need to show that H has reasonable properties, basically that H is approximately Gaussian. Let $f(h)$ be the probability that $H = h$. We will derive an exact formula for $f(h)$.

Consider the experiment of putting s indistinguished balls in $b = n^{2/3}$ distinguished bags without putting three in any bag. If we have h collisions then h bags get 2 balls, $s - 2h$ bags get 1 ball and $b - s + h$ bags get no balls. The number of ways to do this is

$$\frac{b!}{(s-2h)!h!(b-s+h)!} \tag{1}$$

Since the balls are distinguished we need to multiply Equation 1 by $s!/2^h$ which is the number of ways to put the s balls into h bags with 2 balls and $s - 2h$ bags of 1 ball.

We then divide by the b^s ways of placing s distinguishable balls into $\vec{b}$ bags to get

$$f(h) = \frac{b!s!}{2^h(s-2h)!(b-s+h)!h!b^s}$$

It is useful to consider the ratio of $f(h)$ and $f(h-1)$.

$$g(h) = \frac{f(h)}{f(h-1)} = \frac{(s-2h+1)(s-2h+2)}{2h(h+b-s)}$$

By Chebyshev's inequality, we only need to consider the case that h is within a constant number of standard deviations around the expected value of H. In this case we have $s = o(n^{2/3}) = o(b)$ and $h = O(s^2/b) = O(s(s/b)) =$

$o(s)$. We then have $g(h)$ approximately $s^2/2bh$. Note that f achieves its maximum about where $g(h) = 1$, i.e., $h = s^2/2b$ which is the expected value of H.

There is a constant r such that if for some k, $s^2/2b - ks/\sqrt{b} \leq h_1 \leq h_2 \leq s^2/2b + ks/\sqrt{b}$ then $f(h_1)$ and $f(h_2)$ are within a factor of $1+rk$. This follows by approximating the product of the $g(h)$'s in this range.

Now we want to show that H and $H + L$ do not differ much as distributions. Let $u(\ell)$ be the probability that $L = \ell$ and $v(x)$ be the probability that $H + L = x$. We have $v(x) = \sum_\ell f(x - \ell)u(\ell)$.

Since the expected value of L is $O(s^2/n)$, by Markov's inequality we can get a good approximation to $v(x)$ by only considering ℓ with $|\ell| = O(s^2/n)$. In this range $f(x)$ and $f(x-\ell)$ differ by at most a factor of $1+O(s^2/n)n^{1/3}/s = 1 + O(s/n^{2/3}) = 1 + o(1)$. We have $v(x) = \sum_\ell (1 + o(1))f(x)u(\ell) = (1 + o(1))f(x)\sum_\ell u(\ell) = f(x) + o(f(x))$ since $\sum_\ell u(\ell) = 1$.

The L_1-norm of the distance of H and $H + L$ is $\sum_x o(f(x)) = o(1)$ since f is a probability distribution. Thus no statistical test can distinguish H and $H + L$ with nontrivial probability. $\square$

By appropriately modifying the distributions $\vec{a}$ and $\vec{b}$ we can give a stronger version of Theorem 10 with a dependence on ϵ.

Corollary 12 *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 13 *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. $\square$

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

Theorem 14 *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. $\square$

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 [26] 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 15 *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* [24]. 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 16 *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 17 *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 18 *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* $\Pr[\mathbf{M}$ *passes*$] > 1-\delta$*; If* $\mathbf{M}$ *is not* (ρ, ϵ, t)*-almost mixing then* $\Pr[\mathbf{M}$ *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 [22, 13, 14, 7, 21], 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 19 *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* $\Delta(\mathbf{M}_1, \mathbf{M}_2) \leq (\epsilon_1, \epsilon_2)$.[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`.

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

Definition 20 *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* $|\vec{e}_u \mathbf{M}^t - \vec{s}_{\mathbf{M},t}| \leq \epsilon$*. 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 21 *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.

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 $O(\epsilon^{-2} t^2 T(n, \epsilon, \frac{1}{6t}))$.

```
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 22 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 23 *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* $t \leq \tau \leq 2t$*,* $|\vec{e}_u \mathbf{M}^\tau - \vec{e}_u \widetilde{\mathbf{M}}^\tau| \leq \epsilon$ *and* $|\vec{s}_{\mathbf{M},t} - \vec{e}_u \widetilde{\mathbf{M}}^\tau| \leq 2\epsilon$*.*

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 $|\vec{e}_u \mathbf{M}^\tau - \vec{e}_u \widetilde{\mathbf{M}}^\tau| \leq \sum_i \sum_{w \in \Gamma_B} \chi_{w,i} |W(w) - W'(w)| \leq \sum_i \sum_{w \in \Gamma_B} \chi_{w,i} W(w) \leq \epsilon$.

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

We can now prove the main lemma:

Lemma 24 *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 23 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 23 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 23, 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 23 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 25 *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 24. □

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 [12], 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

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

We can show it is a true metric, 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 [12] 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 [25] 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.

References

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

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

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

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

[5] N. Cressie and P. Morgan. Design considerations for neyman pearson and wald hypothesis testing. *Metrika*, 36(6):317–325, 1989.

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

[7] F. Ergün, S. Kannan, S. R. Kumar, R. Rubinfeld, and M. Viswanathan. Spot-checkers. In *STOC 30*, pages 259–268, 1998.

[8] 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.

[9] 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.

[10] A. Frieze and R. Kannan. Quick approximation to matrices and applications. *Combinatorica*, 19, 1999.

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

[12] O. Goldeich and D. Ron. On testing expansion in bounded-degree graphs. Technical Report TR00-020, ECCC, 2000.

[13] O. Goldreich, S. Goldwasser, and D. Ron. Property testing and its connection to learning and approximation. In *FOCS 37*, pages 339–348. IEEE, 14–16 Oct. 1996.

[14] O. Goldreich and D. Ron. Property testing in bounded degree graphs. In *STOC 29*, pages 406–415, 1997.

[15] G. H. Golub and C. F. van Loan. *Matrix Computations*. The John Hopkins University Press, Baltimore, MD, 1996.

[16] R. Kannan. Markov chains and polynomial time algorithms. In S. Goldwasser, editor, *FOCS 35*, pages 656–673. IEEE Computer Society Press, Nov. 1994.

[17] S. Kannan and A. C.-C. Yao. Program checkers for probability generation. In J. L. Albert, B. Monien, and M. Rodríguez-Artalejo, editors, *ICALP 18*, volume 510 of *Lecture Notes in Computer Science*, pages 163–173, Madrid, Spain, 8–12 July 1991. Springer-Verlag.

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

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

[20] B. N. Parlett. *The symmetric eigenvalue problem*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1998. Corrected reprint of the 1980 original.

[21] M. Parnas and D. Ron. Testing the diameter of graphs. In D. Hochbaum, K. Jensen, J. D. Rolim, and A. Sinclair, editors, *Randomization, Approximation, and Combinatorial Optimization*, volume 1671 of *Lecture Notes in Computer Science*, pages 85–96. Springer-Verlag, 8–11 Aug. 1999.

[22] R. Rubinfeld and M. Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM J. Comput.*, 25(2):252–271, Apr. 1996.

[23] A. Sahai and S. 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 Oct. 1997.

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

[25] W. 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.

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

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