On Approximating the Stationary Distribution of Time-Reversible Markov Chains

Marco Bressan\(^1\) · Enoch Peserico\(^2\) · Luca Pretto\(^2\)

Published online: 5 April 2019
© Springer Science+Business Media, LLC, part of Springer Nature 2019

Abstract
Approximating the stationary probability of a state in a Markov chain through Markov chain Monte Carlo techniques is, in general, inefficient. Standard random walk approaches require $\tilde{O}(\tau/\pi(v))$ operations to approximate the probability $\pi(v)$ of a state $v$ in a chain with mixing time $\tau$, and even the best available techniques still have complexity $\tilde{O}(\tau^{1.5}/\pi(v)^{0.5})$; and since these complexities depend inversely on $\pi(v)$, they can grow beyond any bound in the size of the chain or in its mixing time. In this paper we show that, for time-reversible Markov chains, there exists a simple randomized approximation algorithm that breaks this “small-$\pi(v)$ barrier”.

Keywords Markov chains · MCMC sampling · Large graph algorithms · Randomized algorithms · Sublinear algorithms

1 Introduction
We investigate the problem of approximating efficiently a single entry of the stationary distribution of an ergodic Markov chain. This problem has two main motivations. First, with the advent of massive-scale data, even complexities linear in the size of the input are often excessive [21]; therefore computing explicitly the entire stationary distribution is often infeasible.

This article is part of the Topical Collection on Special Issue on Theoretical Aspects of Computer Science (2018)

Marco Bressan
bressan@di.uniroma1.it

Enoch Peserico
enoch@dei.unipd.it

Luca Pretto
pretto@dei.unipd.it

1 Dipartimento di Informatica, Sapienza Università di Roma, Roma, Italy
2 Dipartimento di Ingegneria dell’Informazione, Università degli Studi di Padova, Padova, Italy
distribution, e.g. via the power method \cite{11}, can be simply infeasible. As an alternative one can then resort to approximating only individual entries of the vector, in exchange for a much lower computational complexity \cite{14, 22}. In fact, if such a complexity is low enough one could efficiently “sketch” the whole vector by quickly getting a fair estimate of its entries. Second, in many practical cases one is really interested in just a few entries at a time. A classic example is that of network centralities, many of which are stationary distributions of an ergodic Markov chain \cite{4}. Indeed, the problem of approximating the Personalized PageRank score of a few nodes in a graph has been repeatedly addressed in the past \cite{5, 6, 16, 17}.

In this paper we seek efficient algorithms for approximating the stationary probability $\pi(v)$ of a target state $v$ in the state space of a discrete-time ergodic Markov chain. Along with the motivations above, the problem arises in estimating heat kernels and graph diffusions, testing the conductance of graphs and chains, developing local algorithms, and has applications in machine learning; see \cite{3, 13} for a thorough discussion. We adopt a simple model where with a single operation one can either (i) simulate one step of the chain or (ii) retrieve the transition probability between a pair of states. Although recent research has provided encouraging results, existing algorithms suffer from a crucial bottleneck: to guarantee a small relative error in the approximation of $\pi(v)$, they incur a cost that grows with $1/\pi(v)$ itself (basically because estimating $\pi(v)$ via repeated sampling requires $1/\pi(v)$ samples). This is a crucial issue since in general there is no lower bound on $\pi(v)$; even worse, if the state space has $n$ states, then most states have mass $\pi(v) = O(\frac{1}{n})$, and one can easily design chains where they have mass exponentially small in $n$. In general, then, the cost of existing algorithms can blow up far beyond $O(n)$ for almost all input states $v$. It is thus natural to ask if the dependence of the complexity on $\pi(v)$ is unavoidable. Unfortunately, one can easily show that $\Omega(\frac{\tau}{\pi(v)})$ operations can be necessary to estimate $\pi(v)$ within any constant multiplicative factor if one makes no assumption on the chain (see Section 6). To drop below this complexity barrier one must then necessarily look at special classes of Markov chains.

We present an algorithm that breaks this “small-$\pi(v)$ barrier” for time-reversible Markov chains. Time-reversible chains are a well-known subclass of Markov chains which lie at the heart of the celebrated Metropolis-Hastings algorithm \cite{12} and are equivalent to random walks on weighted undirected graphs \cite{15}. Formally, given any $\epsilon, \delta > 0$ and any state $v$ in a time-reversible chain, our algorithm with probability $1 - \delta$ returns a multiplicative $(1 \pm \epsilon)$-approximation of $\pi(v)$ by using $\tilde{O}(\tau \|\pi\|^{-1})$ operations, where $\tau$ is the mixing time of the chain, $\|\cdot\|$ is the Euclidean norm and $\tilde{O}(\cdot)$ hides polynomials in $\epsilon^{-1}, \ln(\delta^{-1}), \ln(\|\pi\|^{-1})$. The complexity is independent of $\pi(v)$, and for all but a vanishing fraction of states in the chain improves by factors at least $\sqrt{n}$ or $\sqrt{\tau}$ over previous algorithms. The heart of our algorithm is a randomized scheme for approximating the sum of a nonnegative vector by sampling its entries with probability proportional to their values. This scheme requires $\tilde{O}(\|\pi\|^{-1})$ samples if $\pi$ is the distribution over the vector entries, which generalizes the $\tilde{O}(\sqrt{n})$ algorithm of \cite{18} and is provably optimal. We prove that our algorithm for estimating $\pi(v)$ is essentially optimal as a function of $\tau$, $n$ and $\|\pi\|$; in fact one cannot do
better even under a stronger computational model where all transition probabilities to/from all visited states are known. Along with running time bounds, that are our primary goals, we also provide trade-offs between the time and space complexity of our algorithms, as well as bounds on the number of distinct chain states they explore. All our algorithms are simple to implement, require no tuning, and experimentally they appear faster than existing alternatives already for medium-sized chains.

The rest of the paper is organized as follows. Section 1.1 pins down definitions and notation; Section 1.2 formalizes the problem; Section 1.3 discusses related work; Section 1.4 summarizes our results. Section 2 presents our vector sum approximation algorithm. Section 3 presents our approximation algorithm for $\pi(v)$. Section 4 discusses time-space trade-offs for our algorithm. Section 5 analyzes the number of distinct states it visits. Section 6 provides the proofs of our lower bounds. Finally, Section 7 describes our experimental results.

1.1 Preliminaries

A discrete-time, finite-state Markov chain is a sequence of random variables $X_0, X_1, \ldots$ taking value over a set of states $V = \{1, \ldots, n\}$, such that for all $i \geq 1$ and all $u_0, \ldots, u_i \in V$ with $\Pr(X_0 = u_0, \ldots, X_{i-1} = u_{i-1}) > 0$ we have $\Pr(X_i = u_i|X_0 = u_0, \ldots, X_{i-1} = u_{i-1}) = \Pr(X_i = u_i|X_{i-1} = u_{i-1})$. Denote by $P = [p_{uu'}]$ the transition matrix of the chain, so that $p_{uu'} = \Pr(X_i = u'|X_{i-1} = u)$. We assume the chain is ergodic, and thus has a limit distribution that is independent from the distribution of $X_0$; the limit distribution then coincides with the stationary distribution $\pi$. Thus $\pi$ is the unique distribution vector such that for any distribution vector $\pi_0$:

$$\pi = \pi P = \lim_{t \to \infty} \pi_0 P^t \quad (1)$$

We denote by $\pi(u)$ the stationary probability, or mass, of $u$, and we always denote by $v$ the target state whose mass is to be estimated. For any $V' \subseteq V$ we let $\pi(V')$ denote $\sum_{u \in V'} \pi(u)$. We also assume the chain is time-reversible, i.e. that for any pair of states $u$ and $u'$ we have:

$$\pi(u) p_{uu'} = \pi(u') p_{u'u} \quad (2)$$

We denote by $\tau$ the standard $\frac{1}{4}$-mixing time of the chain. In words, $\tau$ is the smallest integer such that after $\tau$ steps the total variation distance between $\pi$ and the distribution of $X_\tau$ is bounded by $\frac{1}{4}$, irrespective of the initial distribution. Formally, $\tau := \min\{t : d(t) \leq \frac{1}{4}\}$, where

$$d(t) := \max_{\pi_0} \|\pi_0 P^t - \pi\|_{TV} = \max_{\pi_0} \frac{1}{2}\|\pi_0 P^t - \pi\|_1 \quad (3)$$

After $\tau$ steps, the distribution of $X_\tau$ converges to $\pi$ exponentially fast; that is, if $t = \eta \tau$ with $\eta \geq 1$, then $\|\pi_0 P^t - \pi\|_{TV} \leq 2^{-\eta}$. In the rest of the paper, $\|\cdot\|$ always
denotes the $\ell^2$ norm. Note that $\frac{1}{\sqrt{n}} \leq \|\pi\| \leq 1$ since $\frac{1}{\sqrt{n}}\|x\|_1 \leq \|x\| \leq \|x\|_1$ for any $x \in \mathbb{R}^n$. One may refer to [15] for a detailed explanation of the notions recalled here.

Unless necessary, we drop multiplicative factors depending only on $\epsilon, \delta$ (see below) from the asymptotic complexity notation. Furthermore, we use the tilde notation to hide polylogarithmic factors, i.e. we denote $O(f \cdot \text{poly}(\log(f)))$ by $\tilde{O}(f)$.

### 1.2 Problem Formulation

We consider a discrete-time, finite-state, time-reversible, ergodic Markov chain on $n$ states. The chain is initially unknown and can only be accessed via two operations (also called queries):

\[
\begin{align*}
\text{step}(): & \text{ accepts in input a state } u, \text{ and returns state } u' \text{ with probability } p_{uu'} \\
\text{probe}(): & \text{ accepts in input an ordered pair of states } (u, u'), \text{ and returns } p_{uu'}
\end{align*}
\]

These queries are the de facto model of previous work. $\text{step}()$ is used in [3, 5, 6, 13, 16, 17] to simulate the walk, assuming each step costs $O(1)$. $\text{probe}()$ is used in [3, 16, 17] to access the elements of the transition matrix, assuming again one access costs $O(1)$. Here, too, we assume $\text{step}()$ and $\text{probe}()$ as well as all standard operations (arithmetic instructions, memory accesses, ...) cost $O(1)$. This includes set insertion and set membership testing; in case their complexity is $\omega(1)$, our bounds can be adapted correspondingly. The problem can now be formalized as follows. The algorithm is given in input a triple $(v, \epsilon, \delta)$ where $v$ is a state in the state space of the chain and $\epsilon, \delta$ are two reals in $(0, 1)$. It must output a value $\hat{\pi}(v)$ such that, with probability $1 - \delta$, it holds $(1 - \epsilon)\pi(v) \leq \hat{\pi}(v) \leq (1 + \epsilon)\pi(v)$. The complexity of the algorithm is counted by the total number of operations it performs. Obviously we seek an algorithm of minimal complexity.

A final remark. We say state $u$ has been visited if $u = v$ or if $u$ has been returned by a $\text{step}()$ call. In line with previous work, we adopt the following “locality” constraint: the algorithm can invoke $\text{probe}()$ and $\text{step}()$ only on visited states.

### 1.3 Related Work

Two recent works address precisely the problem of estimating $\pi(v)$ in Markov chains. The key differences with our paper are that they consider general (i.e. not necessarily time-reversible) chains, and that we aim at a small relative error for any $\pi(v)$ and not only for large $\pi(v)$.

The first work, [13], gives a local approximation algorithm for $\pi(v)$ based on estimating return times via truncated random walks. Given any $\Delta > 0$, if $\pi(v) \geq \Delta$ the algorithm with probability $1 - \delta$ outputs a multiplicative $\epsilon Z(v)$-approximation of $\pi(v)$, where $Z(v)$ is a “local mixing time” that depends on the structure of the chain. The cost is $\tilde{O}(\ln(1/\delta)/\epsilon^3 \Delta)$ $\text{step}()$ calls. If one wants a multiplicative $(1 \pm \epsilon)$-approximation of $\pi(v)$ for a generic $v$, the cost becomes $\tilde{O}(\tau/\pi(v))$ $\text{step}()$ calls since one must wait for the walks to hit $v$ after having mixed.
The second work, [3], gives an algorithm to approximate the \( \ell \)-step transition probability \( p^\ell_{uv} \) from a state \( u \) to a state \( v \), based on coupling a local exploration of the transition matrix \( P \) with simulated random walks. Let \( d \) be the average number of neighbors of the states in the chain. If \( v \) is drawn uniformly at random from the chain state space, and if \( p^\ell_{uv} \geq \Delta \) for some \( \Delta > 0 \), then with probability \( 1 - \delta \) the algorithm gives a multiplicative \( (1 \pm \epsilon) \)-approximation of \( p^\ell_{uv} \) performing \( \tilde{O}(\ell^{1.5} \sqrt{d \ln(1/\delta) / \epsilon \Delta^{0.5}}) \) calls to step() and probe() in expectation. To estimate \( \pi(v) \) for a given \( v \), one must then set \( \ell = \tau \) and \( \Delta = \pi(v) \); and since if the chain is irreducible then \( d = \Omega(1) \), the bound stays at \( \tilde{O}(\tau^{1.5} / \pi(v)^{0.5}) \). Note that this does not contradict our \( \Omega(\tau / \pi(v)) \) lower bound for general Markov chains (see Section 6.2), which is worst-case rather than average-case.

The algorithms cited above require either \( \tilde{O}(\tau / \pi(v)) \) or \( \tilde{O}(\tau^{1.5} / \pi(v)^{0.5}) \) step() and probe() calls to ensure a \( (1 \pm \epsilon) \)-approximation of \( \pi(v) \) for a generic state \( v \). Note also that their complexity and approximation guarantees depend on knowledge of \( \tau \); our algorithms are no exception, and we prove our bounds as a function of \( \tau \).

For the sake of completeness we mention a few interesting results that have been obtained for specific Markov chains, and in particular for PageRank. We remark that these results adopt graph access models based on adjacency lists or adjacency matrix representations, which are different from the step() and probe() model of this paper; and they also exploit the specific structure of PageRank itself, including the fact that for PageRank \( \pi(v) = \Omega(1/n) \) and \( \tau = O(1) \) in an \( n \)-node graph. The algorithm of [5, 6] approximates the PageRank \( \pi(v) \) of the nodes \( v \) having \( \pi(v) \geq \Delta \), at the cost of \( \tilde{O}(1/\Delta) \) step() calls; if one desires a multiplicative \( (1 \pm \epsilon) \)-approximation of \( \pi(v) \), the cost becomes \( \tilde{O}(1/\pi(v)) \). The results of [7, 8] show how to obtain a \( (1 \pm \epsilon) \)-approximation of \( \pi(v) \) for any given \( v \) by accessing only \( \tilde{O}(n^{4/5} d^{1/5}) \) nodes and arcs of the graph, where \( d \) is the average degree of the graph; for nodes with a small \( \pi(v) \) this bound is \( o(1/\pi(v)) \) unless the graph is dense. For Personalized PageRank, [17] gives an algorithm for estimating \( \pi(v) \) using techniques similar to [3]; if one aims at a multiplicative \( (1 \pm \epsilon) \)-approximation of \( \pi(v) \), if \( v \) is drawn uniformly at random from the graph, then the algorithm makes \( \tilde{O}(d^{0.5} / \pi(v)^{0.5}) \) step() and probe() calls in expectation. Similar bounds have been ported to the worst case for Personalized PageRank on undirected graphs – see [16].

Finally, for the problem of estimating the sum of a nonnegative \( n \)-entry vector \( x \) by sampling its entries \( x_i \) with probability \( \pi_i = x_i / \sum_i x_i \), the only algorithm existing to date is that of [18]. That algorithm takes \( \tilde{O}(\sqrt{n}) \) samples independently of \( \pi \), while ours needs \( O(\sqrt{n}) \) samples only in the worst case, i.e. if \( \pi \) is (essentially) the uniform distribution.

1.4 Our Results

Our first contribution is SUMAPPROX, a randomized algorithm for estimating the sum \( \gamma \) of a nonnegative vector \( x \), assuming one can sample its entries according to the probability distribution \( \pi = x / \gamma \). Formally, we prove:
Theorem 1 Given any $\epsilon, \delta \in (0, 1)$, SUMAPPROX($\epsilon, \delta$) with probability at least $1 - \delta$ returns a multiplicative $(1 \pm \epsilon)$-approximation of $\gamma$ by taking $O\left(\|\pi\|^{-1} \epsilon^{-3} (\ln \frac{1}{\delta})^{3/2}\right) = O(\|\pi\|^{-1})$ samples.

SUMAPPROX is extremely simple, yet it improves on the state-of-the-art $\tilde{O}(\sqrt{n})$ algorithm of [18]. We prove $\Omega(\|\pi\|^{-1})$ samples are necessary, too, to get a fair estimate of $\gamma$.

We then employ SUMAPPROX to build MASSAPPROX, a randomized algorithm for approximating $\pi(v)$. Random-walk-based sampling and time reversibility are the ingredients that allow one to make the connection. We prove:

Theorem 2 Given any $\epsilon, \delta \in (0, 1)$ and any state $v$ in a time-reversible Markov chain, MASSAPPROX($\epsilon, \delta, v$) with probability $(1 - \delta)$ returns a multiplicative $(1 \pm \epsilon)$ approximation of $\pi(v)$ using $\tilde{O}(\tau \|\pi\|^{-1} \epsilon^{-3} (\ln \frac{1}{\delta})^{3/2}) = \tilde{O}(\tau \|\pi\|^{-1})$ elementary operations and calls to step() and probe().

Algorithms previously presented in the literature work also for non-reversible chains; but on the $n - o(n)$ states with mass $\pi(v) = O(1/n)$, their complexity becomes at least $\tilde{O}(\tau n)$ [13] or $\tilde{O}(\tau^{1.5} \sqrt{n})$ [3]. In fact, $\pi(v)$ can be arbitrarily small (even exponentially small in $n$ and $\tau$) for almost all states in the chain, so for almost all states the complexity of previous algorithms explodes while that of MASSAPPROX remains unchanged: since $\|\pi\|^{-1} \leq \sqrt{n}$ for any $\pi$, the complexity of MASSAPPROX is at most $\tilde{O}(\tau \sqrt{n})$.

Next, we show that MASSAPPROX is optimal as a function of $\tau$, $n$ and $\|\pi\|$, up to small factors. In fact no algorithm can perform better even if, instead of using step() and probe(), it is equipped with a more powerful query neigh() that, invoked on a state $u$, reveals all incoming and outgoing transition probabilities of $u$. Formally, we prove:

Theorem 3 For any function $v(n) \in \Omega(1/\sqrt{n}) \cap O(1)$ there is a family of time-reversible chains on $n$ states where (a) $\|\pi\| = \Theta(v(n))$, and (b) there is a target state $v$ such that, to estimate its mass $\pi(v)$ within any constant multiplicative factor with constant probability, any algorithm needs $\Omega(\tau \|\pi\|^{-1} / \ln n)$ neigh() calls where $\tau$ is the mixing time of the chain.

We then turn to bound the space requirements of our algorithms. The naive space bounds follow from the running time bounds of Theorem 1, which depend on $\|\pi\|^{-1}$ and can grow as fast as $\sqrt{n}$. This may be overly expensive if $n$ is too large. To this end we show how one can reduce the space used by SUMAPPROX and MASSAPPROX by any given factor, if one is willing to increase the running time correspondingly. For example, one can ensure MASSAPPROX uses just $O(1)$ space, for a running time of $\tilde{O}(\tau \|\pi\|^{-2})$. In addition, we prove no algorithm can achieve a substantially better trade-off.
Next, we analyze the footprint of our algorithm \textsc{MassApprox}, i.e. the number of distinct states it visits. This is useful if visiting a new state is expensive (e.g. if states are users in a social network). We develop an alternative algorithm, \textsc{FullMassApprox}, that has the same worst-case running time guarantees as \textsc{MassApprox}, but whose footprint can be smaller than that of \textsc{MassApprox} depending on the structure of the Markov chain (and at the expense of space). For example, on some chains \textsc{FullMassApprox} provably visits only roughly $\sqrt{\tau n}$ distinct states, whereas \textsc{MassApprox} visits $\tau \sqrt{n}$ ones. In addition, \textsc{FullMassApprox} can have a smaller running time than \textsc{MassApprox}, and in fact in our experiments this is always the case.

Finally, we provide an experimental evaluation of \textsc{MassApprox} and \textsc{FullMassApprox} by running them on two synthetic Markov chains against the competing state-of-the-art algorithms of [13] and [3]. For the same target accuracy, our algorithms are always faster in terms of number of queries used. In fact, for all $\epsilon \leq 1$ \textsc{FullMassApprox} is at least two orders of magnitude faster than [13] and [3] in all our experiments.

2 Estimating Sum by Weighted Sampling

As anticipated in Section 1, and as we show formally in Section 3, the problem of estimating the stationary probability $\pi(v)$ of $v$ can be reduced to the following problem. We are given a vector of nonnegative reals $\gamma_u$ indexed by the elements $u$ of a set $V$. The vector is unknown, including its dimension, but we can access its entries via weighted sampling. More precisely, we can draw pairs $(u, \gamma_u)$ according to the distribution $\pi$ where $(u, \gamma_u)$ has probability $\gamma_u / \sum_{u \in V} \gamma_u$. The goal is to approximate the vector sum $\gamma = \sum_{u \in V} \gamma_u$. Note that a sample $(u, \gamma_u)$ reveals the index of an entry together with its value. We describe here below a simple randomized algorithm, \textsc{SumApprox}, which estimates the sum $\gamma$ by drawing samples until witnessing a certain number of repeats (a sample is a repeat if it returns a pair already sampled before).

The idea behind \textsc{SumApprox} is the following. Each time a new element is to be drawn, the probability of a repeat is given by the sum of the probabilities of all the distinct elements drawn so far; this sum is $w_S / \gamma$, where $\gamma$ is unknown – indeed, it is the quantity to be determined. Now, consider a sequence of binary random variables $\chi_i$, $i = 0, 1, \ldots$ the $(i + 1)$-th of which assumes value 1 if and only if the $(i + 1)$-th draw is a repeat. The expectation of $\chi_i$ coincides with the probability that $\chi_i$ assumes value 1, and is thus given by the value assumed by $w_S / \gamma$ immediately before the $(i + 1)$-th execution of line 8. \textsc{SumApprox} keeps track of two key quantities. The first is the numerator of the sum of the expectations of the $\chi_i$, that is, the sum of the values of $w_S$ over all iterations so far; this quantity is stored in the variable $w$. The second is the sum of the realizations of the $\chi_i$, which is simply the number of repeats witnessed so far; this is stored in the variable $r$. If the sum of the realizations of the $\chi_i$ concentrates around its own expectation, as we prove, then $r$ is close to $w / \gamma$, and thus $w / r$ is an accurate estimate of $\gamma$. 
We prove the following guarantees:

**Theorem 4** \( \text{SUMAPPROX}(\epsilon, \delta) \) with probability at least \( 1 - \frac{2\delta}{3} \) returns an estimate \( \hat{\gamma} \) such that \( |\hat{\gamma} - \gamma| < \epsilon \gamma \).

**Theorem 5** \( \text{SUMAPPROX}(\epsilon, \delta) \) draws at most \( \lceil 45\|\pi\|^{-1} \epsilon^{-3} (\ln \frac{3}{\delta})^{3/2} \rceil \) samples with probability at least \( 1 - \frac{\delta}{3} \).

We remark that the previous existing algorithm for the sum estimation problem \([18]\) needs knowledge of \( n = |V| \) and uses \( O(\sqrt{n} \epsilon^{-7/2} \log n \left(\log \frac{1}{\delta} + \log \frac{1}{\epsilon} + \log \log n\right)) = O(\sqrt{n} \log n \log \log n) \) samples. \( \text{SUMAPPROX} \) is simpler, oblivious to \( n \), and its running time bounds are tighter – they save a factor that ranges from \( \log n \log \log n \) when \( \pi \) is (essentially) the uniform distribution to \( \sqrt{n} \log n \log \log n \) when \( \pi \) is heavily skewed. In fact, we show that \( \text{SUMAPPROX} \) is optimal up to constant factors, by proving that \( \Omega(\|\pi\|^{-1}) \) samples are in general necessary to estimate \( \gamma \) even if \( n \) is known. This extends to arbitrary distributions the \( \Omega(\sqrt{n}) \) lower bound given by \([18]\) for the uniform distribution.

**Theorem 6** For any function \( \nu(n) \in \Omega(n^{-\frac{1}{2}}) \cap O(1) \) there exist vectors \( \mathbf{x} = \gamma \pi = (\gamma_1, \ldots, \gamma_n) \) with \( \|\pi\| = \Theta(\nu(n)) \) such that \( \Omega(\|\pi\|^{-1}) \) samples are necessary to estimate \( \gamma \) within constant multiplicative factors with constant probability, even if \( n \) is known.

The rest of this section is devoted to the proofs.

### 2.1 Proof of Theorem 4

We make use of a martingale tail inequality originally from \([10]\) and stated (and proved) in the following form as Theorem 2.2 of \([1]\), p. 1476:
\begin{theorem}[\cite{1}, Theorem 2.2] \label{thm:approx}
Let \((Z_0, Z_1, \ldots)\) be a martingale with respect to the filter \((\mathcal{F}_t)\). Suppose that \(Z_{i+1} - Z_i \leq M\) for all \(i\), and write \(V_i = \sum_{i=1}^{t} \text{Var}(Z_i | \mathcal{F}_{i-1})\). Then for any \(z, v > 0\) we have

\[
\Pr\left[ Z_t \geq Z_0 + z, V_i \leq v \text{ for some } i \right] \leq \exp\left[ -\frac{z^2}{2(v + Mz)} \right]
\]

Let us plug into Theorem 7 the appropriate quantities from \textsc{SumApprox}:

- Let \((X_i, \gamma X_i)\) be the random pair drawn at the \((i + 1)\)th execution of line 8.
- Let \(\chi_i\) be the indicator variable of the event that \((X_i, \gamma X_i)\) is a repeat, that is, \(\chi_i = 1\) if \(X_i \in \bigcup_{j=0}^{i-1} \{X_j\}\).
- Let \(\mathcal{F}_i\) be the event space generated by \(X_0, \ldots, X_i\), so that for any random variable \(Y\), with \(\mathbb{E}[Y | \mathcal{F}_i]\) we mean \(\mathbb{E}[Y | X_0, \ldots, X_i]\) and with \(\text{Var}(Y | \mathcal{F}_i)\) we mean \(\text{Var}(Y | X_0, \ldots, X_i)\).
- Let \(P_i = \mathbb{E}[\chi_i | \mathcal{F}_{i-1}]\) be the probability that \((X_i, \gamma X_i)\) is a repeat as a function of \((X_0, \gamma X_0, \ldots, (X_{i-1}, \gamma X_{i-1})\), that is, \(P_i = \sum_{u \in \bigcup_{j=0}^{i-1} \{X_j\}} \frac{Y_u}{\gamma} \).
- Let \(Z_i = \sum_{j=0}^{i} (\chi_j - P_j)\); it is easy to see that \((Z_i)_{i \geq 0}\) is a martingale with respect to the filter \((\mathcal{F}_i)_{i \geq 0}\), since \(Z_i\) is obtained by adding to \(Z_{i-1}\) the indicator variable \(\chi_i\) and subtracting \(P_i\) i.e. its expectation in \(\mathcal{F}_{i-1}\). More formally, \(\mathbb{E}[Z_i | \mathcal{F}_{i-1}] = \mathbb{E}[Z_{i-1} + \chi_i - P_i | \mathcal{F}_{i-1}]\), and since \(Z_{i-1}\) and \(P_i\) are completely determined by \(X_0, \ldots, X_{i-1}\), the right-hand term is simply \(Z_{i-1} + (\mathbb{E}[\chi_i | \mathcal{F}_{i-1}] - P_i) = Z_{i-1}\). Note also that \(Z_0 = 0\).
- Let \(M = 1\), noting that \(|Z_{i+1} - Z_i| = |\chi_{i+1} - P_{i+1}| \leq 1\) for all \(i\).

Finally, note that \(\text{Var}(Z_j | \mathcal{F}_{j-1}) = \text{Var}(\chi_j | \mathcal{F}_{j-1})\) as \(Z_j = Z_{j-1} + \chi_j - P_j\) and, again, \(Z_{j-1}\) and \(P_j\) are completely determined by \(X_0, \ldots, X_{j-1}\). Since \(\text{Var}(\chi_j | \mathcal{F}_{j-1}) = P_j(1 - P_j) \leq P_j\), we have \(V_i = \sum_{j=1}^{i} \text{Var}(Z_j | \mathcal{F}_{j-1}) \leq \sum_{j=1}^{i} P_j\).

Theorem 7 then yields the following:

\begin{lemma} \label{lem:approx}
For all \(z, v > 0\) we have

\[
\Pr\left[ Z_i \geq z, \sum_{j=1}^{i} P_j \leq v \text{ for some } i \right] \leq \exp\left[ -\frac{z^2}{2(v + z)} \right]
\]

(4)

Recall now \textsc{SumApprox}. Note that \(\sum_{j=1}^{i} P_j\) and \(Z_i\) are respectively the value of \(w\) and of \(r - \frac{w}{\gamma}\) just after the \textit{while} loop has been executed for the \((i + 1)\)-th time. Note also that, when \textsc{SumApprox} returns, \(r = k_{\epsilon, \delta}\). Therefore the event that, when \textsc{SumApprox} returns, \(\frac{w}{\gamma} \leq \gamma(1 - \epsilon)\) i.e. \(\frac{w}{\gamma} \leq r(1 - \epsilon) \leq (1 - \epsilon)k_{\epsilon, \delta}\) corresponds to the event that \(Z_i \geq \epsilon r = \epsilon k_{\epsilon, \delta}\) and \(\sum_{j=1}^{i} P_j \leq (1 - \epsilon)k_{\epsilon, \delta}\). Invoking Lemma 1 with \(z = \epsilon k_{\epsilon, \delta}\) and \(v = (1 - \epsilon)k_{\epsilon, \delta}\):

\[
\Pr\left[ \frac{w}{r} \leq \gamma(1 - \epsilon) \right] \leq \exp\left[ -\frac{\epsilon^2 k_{\epsilon, \delta}^2}{2(\epsilon k_{\epsilon, \delta} + (1 - \epsilon)k_{\epsilon, \delta})} \right] = \exp\left[ -\frac{\epsilon^2 k_{\epsilon, \delta}}{2} \right]
\]

(5)
\end{lemma}
which is smaller than \( \delta/3 \) since clearly \( k_{\epsilon,\delta} > \frac{2}{\epsilon^2} \ln \frac{3}{\delta} \). Consider instead the event that, when \textsc{SumApprox} returns, \( \frac{u}{r} \geq \gamma(1+\epsilon) \) i.e. \( \frac{u}{r} \geq r(1+\epsilon) = k_{\epsilon,\delta}(1+\epsilon) \).

This is the event that \( Z_i \leq -\epsilon k_{\epsilon,\delta} \), or equivalently \( -Z_i \geq \epsilon k_{\epsilon,\delta} \). Note that Lemma 1 still holds if we replace \( Z_i \) with \(-Z_i \), as \((-Z_i)_{i\geq 0} \) too is obviously a martingale with respect to the filter \((F_i)_{i\geq 0} \), with \(-Z_0 = 0 \). Let then \( i_0 \leq i \) be the smallest time such that \(-Z_{i_0} \geq \epsilon k_{\epsilon,\delta} \). Since \(|Z_j - Z_{j-1}| \leq 1 \), it must be \(-Z_{i_0} < \epsilon k_{\epsilon,\delta} + 1 \).

Also, since \( \sum_{j=0}^{i_0} X_j \) is non-decreasing with \( i \), then \( \sum_{j=0}^{i_0} X_j \leq k_{\epsilon,\delta} \). It follows that \( \sum_{j=0}^{i_0} P_j = -Z_{i_0} + \sum_{j=0}^{i_0} X_j \leq \epsilon k_{\epsilon,\delta} + 1 + k_{\epsilon,\delta} = (1+\epsilon)k_{\epsilon,\delta} + 1 \). Invoking again Lemma 1 with \( z = \epsilon k_{\epsilon,\delta} \) and \( v = (1+\epsilon)k_{\epsilon,\delta} + 1 \), we obtain:

\[
\Pr \left[ \frac{u}{r} \geq \gamma(1+\epsilon) \right] \leq \exp \left[ -\frac{\epsilon^2 k_{\epsilon,\delta}^2}{2((1+\epsilon)k_{\epsilon,\delta} + 1)} \right] \tag{6}
\]

Note that \( \frac{1}{k_{\epsilon,\delta}} < \frac{e^2}{2+4.4\epsilon} < 0.2\epsilon \) since \( \epsilon \leq 1 \); so \( 2((1+2\epsilon) + \frac{1}{k_{\epsilon,\delta}}) < 2 + 4.4\epsilon \), and since \( k_{\epsilon,\delta} \geq \frac{2+4.4\epsilon}{e^2} \ln \frac{3}{\delta} \) the right-hand term is at most \( \frac{1}{2} \). Finally, by a simple union bound the probability that \( |\hat{\gamma} - \gamma| \geq \epsilon \gamma \) is at most \( 2\Delta^2 \), and the proof of Theorem 4 is complete.

### 2.2 Proof of Theorem 5

We show that the probability that \( s = \lceil 45\|\pi\|^{-1} e^{-3} (\ln \frac{3}{\delta})^{3/2} \rceil \) draws yield less than \( k_{\epsilon,\delta} \) repeats is less than \( \frac{\delta}{3} \). To this end we make use of the following Chernoff-type probability bounds, which can be found in e.g. [2] and can be derived from [20]. Let \( X_1, \ldots, X_n \) be binary random variables. We say that \( X_1, \ldots, X_n \) are non-positively correlated if for all \( I \subseteq \{1, \ldots, n\} \) and \( B \in \{0, 1\} \) we have:

\[
\Pr[\forall i \in I : X_i = B] \leq \prod_{i \in I} \Pr[X_i = B] \tag{7}
\]

The following lemma holds:

**Lemma 2** Let \( X_1, \ldots, X_n \) be independent or, more generally, non-positively correlated binary random variables. Let \( a_1, \ldots, a_n \in [0, 1] \) and \( X = \sum_{i=1}^{n} a_i X_i \). Then, for any \( \epsilon > 0 \), we have:

\[
\Pr[X < (1-\epsilon)\mathbb{E}[X]] < e^{-\frac{\epsilon^2}{2}\mathbb{E}[X]} \tag{8}
\]

\[
\Pr[X > (1+\epsilon)\mathbb{E}[X]] < e^{-\frac{\epsilon^2}{2\epsilon^2}\mathbb{E}[X]} \tag{9}
\]

Let now \( \tilde{\rho} = \frac{5}{18} \|\pi\| \epsilon (\ln \frac{3}{\delta})^{-1/2} \). We consider two cases.

**Case 1:** \( \exists u \in V \) with \( \pi(u) > \tilde{\rho} \). Let then \( C_u^s \) be the random variable counting the number of times \( u \) appears in \( s \) draws. Since if \( C_u^s > k_{\epsilon,\delta} \) then \( u \) causes at least \( k_{\epsilon,\delta} \) repeats, the probability that \textsc{SumApprox} needs more than \( s \) draws is bounded from above by \( \Pr[C_u^s \leq k_{\epsilon,\delta}] \). Now \( \mathbb{E}[C_u^s] = s \pi(u) > s \tilde{\rho} > 45 \frac{5}{18} \epsilon^2 \ln \frac{3}{\delta} = \frac{125}{e^2} \ln \frac{3}{\delta} \geq 1.7 \left( \frac{2+4.4\epsilon}{e^2} \ln \frac{3}{\delta} \right) = 1.7k_{\epsilon,\delta} \), therefore \( C_u^s \leq k_{\epsilon,\delta} \).
implies $C_u^s < \frac{1}{1.7} \mathbb{E}[C_u^s] < (1 - 0.41)\mathbb{E}[C_u^s]$. Since $C_u^s$ is a sum of independent binary random variables, the bounds of Lemma 2 give $\Pr[C_u^s \leq k_{\epsilon, \delta}] < \exp\left(-\frac{1}{2}0.41^2 \mathbb{E}[C_u^s]\right) < \exp\left(-0.5 \cdot 0.41^2 \cdot \frac{12.5}{\epsilon^2} \ln \frac{3}{\delta}\right) < \exp\left(-1.05 \ln \frac{3}{\delta}\right) < \frac{\delta}{3}$.

Case 2: $\pi(u) \leq \bar{p}$ for all $u \in V$. Let then $\bar{s} = \lceil \bar{p}^{-1} \rceil$, let $\bar{S}$ be the set of distinct elements in the first $\bar{s}$ draws, and let $w(\bar{s}) = \sum_{u \in \bar{S}} \pi(u)$. First we show that $\mathbb{E}[w(\bar{s})] \geq \frac{1}{2\bar{s}}\|\pi\|^2$. Write $\mathbb{E}[w(\bar{s})] = \sum_{u \in V} \pi(u)(1 - (1 - \pi(u))^\bar{s})$. Since for all $x \in [0, 1]$ and $k \geq 1$ it holds $(1 - x)^k \leq (1 + kx)^{-1}$, by setting $x = \pi(u)$ and $k = \bar{s}$ we obtain $1 - (1 - \pi(u))^\bar{s} \geq 1 - (1 + \bar{s}\pi(u))^{-1} = \bar{s}\pi(u)(1 + \bar{s}\pi(u))^{-1}$. Moreover note that $\bar{p}^{-1} \geq \frac{18}{5} = 3.6$ and thus $\lceil \bar{p}^{-1} \rceil \leq \frac{5}{4}\bar{p}^{-1}$. Therefore $\bar{s}\pi(u) \leq \lceil \bar{p}^{-1} \rceil \bar{p} \leq \frac{5}{4}$ for all $u$, and thus $\bar{s}\pi(u)(1 + \bar{s}\pi(u))^{-1} \geq \bar{s}\pi(u)\frac{1}{1 + \frac{5}{4}} = \frac{4}{5}\bar{s}\pi(u)$. Therefore $\mathbb{E}[w(\bar{s})] \geq \frac{4}{5\bar{s}} \sum_{u \in V} \pi(u)^2 \geq \frac{4}{5\bar{s}} \|\pi\|^2$. Now we consider two cases.

First, suppose the event $w(\bar{s}) \geq 0.4 \mathbb{E}[w(\bar{s})]$ takes place. For $i = \bar{s} + 1, \ldots, s$ let $\chi_i$ be the indicator random variable of the event that the $i$-th draw is an element of $\bar{S}$, and let $C_s = \sum_{i = \bar{s} + 1}^s \chi_i$. Clearly SUMAPPROX witnesses at least $C_s$ repeats in the last $s - \bar{s}$ draws, and thus overall. We shall then bound $\Pr[C_s < k_{\epsilon, \delta}]$. First, since by hypothesis the total mass of $\bar{S}$ is $w(\bar{s}) \geq 0.4 \mathbb{E}[w(\bar{s})]$, we also have $\mathbb{E}[\chi_i] \geq 0.4 \mathbb{E}[w(\bar{s})] \geq \frac{1.6}{\bar{s}} \|\pi\|^2$. Therefore $\mathbb{E}[C_s] = \sum_{i = \bar{s} + 1}^s \mathbb{E}[\chi_i] \geq \frac{1.6}{\bar{s}}(s - \bar{s})\|\pi\|^2$. Now note that $s - \bar{s} > 10\bar{s}$, therefore $\mathbb{E}[C_s] \geq \frac{16}{5}\bar{s}^2\|\pi\|^2$. Finally, since $\bar{s} = \lceil \bar{p}^{-1} \rceil \geq \frac{18}{5}\|\pi\|^2\epsilon^{-1}(\ln \frac{3}{\delta})^{1/2}$, it holds $\mathbb{E}[C_s] > \left(\frac{18}{5}\right)^2 \frac{16}{5} \frac{1}{\epsilon^2} \ln \frac{3}{\delta} > 23\frac{1}{\epsilon^2} \ln \frac{3}{\delta} > 3.14k_{\epsilon, \delta}$. It follows that the event $C_s < k_{\epsilon, \delta}$ implies $C_s < \frac{1}{3.14}\mathbb{E}[C_s] < (1 - 0.68)\mathbb{E}[C_s]$. By the concentration bounds of Lemma 2, the probability of the latter is $\Pr[C_s < k_{\epsilon, \delta}] \leq \exp\left(-\frac{1}{2}0.68^2 \mathbb{E}[C_s]\right) < \exp\left(-\frac{1}{2}0.68^2 \frac{23}{\epsilon^2} \ln \frac{3}{\delta}\right) \leq \exp\left(-5\ln \frac{3}{\delta}\right) < \frac{\delta}{243}$.

The second event corresponds to the event $w(\bar{s}) < 0.4 \mathbb{E}[w(\bar{s})] = (1 - 0.6)\mathbb{E}[w(\bar{s})]$, of which we shall bound the probability. Let $\chi_u^\bar{s}$ be the indicator variable of the event $u \in \bar{S}$, so $w(\bar{s}) = \sum_{u \in V} \chi_u^\bar{s} \pi(u)$. Since $\pi(u) \leq \bar{p}$ for all $u$, we can write $w(\bar{s}) = \bar{p} \sum_{u \in V} \chi_u^\bar{s} \bar{p}^{-1} \pi(u)$ so that the coefficients $\bar{p}^{-1} \pi(u)$ are in $[0, 1]$. Clearly, the $\chi_u^\bar{s}$ are non-positively correlated. We can thus invoke Lemma 2 and get $\Pr[w(\bar{s}) < 0.4 \mathbb{E}[w(\bar{s})]] \leq \exp\left(-0.5 \cdot 0.6^2 \bar{p}^{-1} \mathbb{E}[w(\bar{s})]\right)$. By replacing the definitions and bounds for $E[w(\bar{s})]$, $\bar{s}$ and $\bar{p}^{-1}$ from above, we get $\Pr[w(\bar{s}) < 0.4 \mathbb{E}[w(\bar{s})]] \leq \exp\left(-2.88 \ln \left(\frac{3}{\delta}\right)\right) < \frac{\delta}{23}$. Again by a union bound, the probability that SUMAPPROX draws more than $s$ samples is less than $\frac{\delta}{243} + \frac{\delta}{23} < \frac{\delta}{3}$.

The proof of Theorem 5 is complete.

2.3 Proof of Theorem 6

Let $k \in \Theta((\nu(n))^{-2})$ with $1 \leq k \leq \frac{\nu}{2}$. Consider the two vectors $x = (\gamma_1, \ldots, \gamma_n)$ and $x' = (\gamma_1', \ldots, \gamma_n')$ defined as follows:

$$
\gamma_j = 1: j \leq k, \quad \gamma_j = \sqrt{k/n}: j > k
$$

$$
\gamma_j' = 1: j \leq 2k, \quad \gamma_j' = \sqrt{k/n}: j > 2k
$$
Now let $\gamma = \sum_{i=1}^{n} \gamma_i$ and $\gamma' = \sum_{i=1}^{n} \gamma'_i$. One can check that $\gamma \leq 2k$ and $|\gamma - \gamma'| \geq \frac{k}{2}$. Hence, to obtain an estimate $\gamma'$ of $\gamma$ with $\gamma' \leq \frac{5}{4} \gamma$, one must distinguish $x$ from $x'$. Note that the norm of $\pi = x/\gamma$ is in $\Theta(1/\sqrt{k}) = \Theta(v(n))$, as requested. Now, for each one of $x$ and $x'$ in turn, pick a permutation of $\{1, \ldots, n\}$ uniformly at random and apply it to the entries of the vector. Suppose then we sample $o(\|\pi\|^{-1}) = o(\sqrt{k})$ entries from $x$. We shall see that, with probability $1 - o(1)$, we cannot distinguish $x$ from $x'$. First, note that the total mass of the entries with value $\sqrt{k}/n$ is at most $1/\sqrt{k}$. Hence the probability of drawing any of those entries with $o(\sqrt{k})$ samples is $o(1)$, and we can assume all draws yield entries having value 1. Since there are $O(k)$ such entries in total, the probability of witnessing any repeat is also $o(1)$, and we can assume no repeat is witnessed. Furthermore, because of the random permutation, the indices of samples are distributed uniformly over $\{1, \ldots, n\}$ (recall that we actually sample from the index set $\{1, \ldots, n\}$, so we could use the distribution of the indices to distinguish $x$ from $x'$). The same argument applies to $x'$, so drawing $o(\sqrt{k})$ samples from $x'$ yields exactly the same distribution and the two vectors are indistinguishable.

To adapt the construction to larger approximation factors, set $\gamma'_j = 1 : j \leq \eta k$ for $\eta$ large enough.

The proof of Theorem 6 is complete.

3 Approximating the Stationary Distribution

In this section we address the problem of approximating $\pi(v)$. We start in Section 3.1, where we reduce this problem to the sum estimation problem of Section 2 by sampling states via random walks. In Section 3.2 we will then be able to prove Theorem 2, presenting our algorithm MASS APPROX for estimating $\pi(v)$.

3.1 From Estimating the Sum to Estimating the Mass

Consider a random walk of length $t + 1$ that starts at $v$. Obviously we can simulate such a walk by setting $u_0 = v$ and then invoking $step(u_i)$ to obtain the state $u_{i+1}$, for $i = 0, \ldots, t - 1$. Crucially, using the time-reversibility of the chain, for any visited state $u$ we can obtain the ratio $\gamma_u$ between $\pi(u)$ and $\pi(v)$ using $O(1)$ operations. Formally, let $\gamma_v = \pi(v)/\pi(v) = 1$, and in general let $\gamma_u = \pi(u)/\pi(v)$. Note that:

$$
\gamma_{u_{i+1}} = \frac{\pi(u_{i+1})}{\pi(v)} = \frac{\pi(u_{i+1})}{\pi(u_i)} \cdot \frac{\pi(u_i)}{\pi(v)} = \frac{\pi(u_{i+1})}{\pi(u_i)} \cdot \gamma_{u_i} \quad (10)
$$

The time-reversibility of the chain (see (2)) implies $\frac{\pi(u_{i+1})}{\pi(u_i)} = \frac{p_{u_i,u_{i+1}}}{p_{u_{i+1},u_i}} = \frac{\text{probe}(u_i,u_{i+1})}{\text{probe}(u_{i+1},u_i)}$, hence we can compute $\gamma_{u_{i+1}}$ with $O(1)$ operations if we know $\gamma_{u_i}$. But then we can keep track of $\gamma_u$ for any $u$ visited so far, starting with $\gamma_{u_0} = 1$ and computing $\gamma_{u_{i+1}}$ by (10) the first time $u_{i+1}$ is visited.
Suppose now to pick \( t \) large enough so that the chain reaches its stationary distribution, i.e. \( u_t \sim \pi \) irrespective of \( v \). One is then drawing state \( u \), as well as its associate weight \( y_u \), with probability \( \pi(u) \). Now if we let \( \gamma = \sum_{u \in V} y_u \), then \( \pi(u) = y_u \gamma^{-1} \) and in particular \( \pi(v) = \gamma^{-1} \). Therefore approximating \( \pi(v) \) amounts to approximating \( \gamma \); more formally, for any \( \epsilon \in (0, 1) \), if \( \gamma' \) is a \( (1 \pm \frac{\epsilon}{2}) \)-approximation of \( \gamma \) then \( \gamma'^{-1} \) is a \( (1 \pm \epsilon) \)-approximation of \( \pi(v) \). We can therefore reduce to the sum approximation problem of Section 3: compute with probability \( (1 - \delta) \) a \( (1 \pm \epsilon) \)-approximation of \( \gamma \), assuming we can draw pairs \( (u, y_u) \) according to \( \pi \). The only problem is that by simulating the chain we can only come close to (but not exactly on) the stationary distribution \( \pi \). We must then tie the approximation guarantees of \( \text{SUMAPPROX} \) to the length \( t \) of the random walks, or better to the distance \( ||\pi' - \pi||_{TV} \) between \( \pi \) and the distribution \( \pi' \) from which \( u_t \) is drawn. Formally, we show:

**Lemma 3** There exists some constant \( c < 80 \) such that the following holds. Choose any \( \delta, \epsilon \in (0, 1) \), and suppose we draw the pairs \( (u, y_u) \) from a distribution \( \pi' \) such that \( ||\pi - \pi'||_{TV} \leq \left( \frac{\epsilon ||\pi||}{\ln(3/\delta)} \right)^c \). Then \( \text{SUMAPPROX}(\frac{\epsilon}{2}, \delta) \) with probability at least \( 1 - \delta \) returns a multiplicative \( (1 \pm \epsilon) \)-approximation of \( \gamma \) by taking at most \( \lceil 720 ||\pi||^{-1} \epsilon^{-3} (\ln \frac{3}{\delta})^{3/2} \rceil \) samples.

**Proof** Let us start with the bound on the number of samples. Recall the proof of Theorem 5, and note that the argument depends on \( \pi \) but not on the values \( y_u \). Indeed, \( \pi \) alone determines the probability of repeats and thus controls the distribution of the number of samples drawn by \( \text{SUMAPPROX} \). Hence, by Theorem 5 \( \text{SUMAPPROX}(\frac{\epsilon}{2}, \delta) \) takes more than \( \lceil 45 ||\pi'||^{-1} 18 \epsilon^{-3} (\ln \frac{3}{\delta})^{3/2} \rceil = \lceil 360 ||\pi'||^{-1} \epsilon^{-3} (\ln \frac{3}{\delta})^{3/2} \rceil \) samples with probability less than \( \frac{\delta}{3} \). Now \( ||\pi - \pi'|| \leq 2 ||\pi - \pi'||_{TV} \leq 2 \left( \frac{\epsilon ||\pi||}{\ln(3/\delta)} \right)^c \leq ||\pi || (2 (\ln 3)^{3/2} )^{-c} \), which for \( c \geq 15 \) is bounded by \( \frac{1}{3} ||\pi|| \). Then, since \( ||\pi'|| \geq ||\pi|| - ||\pi - \pi'|| \), we have \( ||\pi'||^{-1} \leq 2 ||\pi||^{-1} \) and the bound above is in turn bounded by \( \lceil 720 ||\pi||^{-1} \epsilon^{-3} (\ln \frac{3}{\delta})^{3/2} \rceil \).

Let us now see the approximation guarantees. Recall the proof of Theorem 4. We want to show again that \( \Pr[|\gamma - \gamma'| \geq 2\delta \gamma'] \leq \frac{2\delta}{3} \). However, now the samples are drawn according to \( \pi' \) instead of \( \pi \). Let then \( P'_j = \sum_{u \in \cup_{h=1}^j X_h} \pi'(u) \) and \( Z'_i = \sum_{j=0}^i (X_j - P'_j) \); in a nutshell, \( P'_j \) and \( Z'_i \) are the analogous of \( P_j \) and \( Z_i \) under \( \pi' \). It is immediate to check that Lemma 1 holds with \( Z'_i \) and \( P'_j \) in place of \( Z_i \) and \( P_j \). Let now \( w'(i) = \gamma \sum_{j=1}^i P'_j \). Note that \( \sum_{j=1}^i P'_j \) and \( Z'_i \) are respectively the value of \( \frac{w'(i)}{\gamma} \) and of \( r - \frac{w'(i)}{\gamma} \) just after line 9 has been executed for the \( (i + 1) \)-th time. Therefore, the argument following Lemma 1 holds if we put \( w'(i) \) in place of the value taken by \( w \) after the \( (i + 1) \)-th execution of line 9. Hence the same bounds hold, and \( \text{SUMAPPROX}(\frac{\epsilon}{2}, \delta) \) ensures \( \Pr[|\gamma - \gamma'| \geq 2\delta \gamma] \leq \frac{\delta}{3} \), where \( s \) is the total number of draws. Now note that \( \text{SUMAPPROX} \) does not return \( \frac{w'(s)}{\gamma} \), but \( \frac{w(s)}{r} \) where \( w(i) = \gamma \sum_{j=1}^i P_j \) is the value of \( w \) in \( \text{SUMAPPROX} \) after line 9 has been executed.
for the \((i + 1)\)-th time. We shall now make \(\frac{|w(s) - w'(s)|}{r} \leq \frac{\epsilon}{2}\gamma\); by the triangle inequality we will then be done. First of all, by the definition of \(w(s)\) and \(w'(s)\) we have

\[
\left| \frac{w(s)}{r} - \frac{w'(s)}{r} \right| = \gamma r^{-1} \left| \sum_{j=1}^{s} P_j - \sum_{j=1}^{s} P'_j \right| \leq \gamma r^{-1} \sum_{j=1}^{s} |P_j - P'_j| \leq \frac{\epsilon}{2}\gamma \sum_{j=1}^{s} P_j \leq \frac{\epsilon\gamma}{2} \sum_{j=1}^{s} P_j
\]  

(11)

Now note that \(|P_j - P'_j| \leq \|\pi - \pi'\|_{TV}\), since \(P_j\) and \(P'_j\) are the probability of the same event under respectively \(\pi\) and \(\pi'\). Therefore the right-hand side of (11) is bounded by \(\gamma r^{-1} s \|\pi - \pi'\|_{TV}\). Now, when SUMAPPROX\((\epsilon, \delta)\) terminates \(r = k\frac{\epsilon}{\gamma} \ln \frac{3}{\delta} \geq 4 \frac{2 + 2.2\epsilon}{\epsilon^2} \ln \frac{3}{\delta}\), and by hypothesis \(\|\pi - \pi'\|_{TV} \leq (\frac{\epsilon\|\pi\|}{\ln(3/\delta)})^c\). Therefore:

\[
\left| \frac{w(s)}{r} - \frac{w'(s)}{r} \right| \leq \gamma s \frac{\epsilon^2}{4(2 + 2.2\epsilon) \ln(\frac{3}{\delta})} \left( \frac{\epsilon\|\pi\|}{\ln(\frac{3}{\delta})} \right)^c \leq \gamma s \|\pi\| \frac{\epsilon^{3+c}}{8\ln(\frac{3}{\delta})^{1+c}}
\]  

(12)

Finally, recall that with probability \(1 - \frac{3}{\delta}\) we have \(s \leq \lceil 720\|\pi\|^{-1}e^{-3\ln(\frac{3}{\delta})^{3/2}} \rceil\). In this case the equation above yields \(|\frac{w(s)}{r} - \frac{w'(s)}{r}| \leq \gamma \cdot 721e^c \ln(\frac{3}{\delta})^{0.5-c}\), which is smaller than \(\frac{\epsilon}{2}\gamma\) for \(c \geq \frac{1}{2} + \ln\frac{1442}{\ln 3} \approx 78\).

A simple union bound completes the proof.

\[\square\]

### 3.2 Proof of Theorem 2

We are now ready to introduce our algorithm MASSAPPROX and prove Theorem 2. MASSAPPROX, whose pseudocode is listed below, is obtained from SUMAPPROX by simply drawing the pair \((u, \gamma_u)\) by performing \(t\) steps of the random walk starting from \(v\). We shall now show that choosing an appropriate \(t\) yields Theorem 2.

Let \(t = \tau c \ln(\|\pi\|^{-1}e^{-1} \ln(\frac{3}{\delta})/\ln 2)\), where \(c < 80\) is the constant of Lemma 3 and \(\tau\) is the mixing time of the chain. Simulate the walk for \(t\) steps starting from \(v\), and let \(\pi'\) be the distribution of the final state. By the properties of the mixing time (see Section 1.1):

\[
\|\pi - \pi'\|_{TV} \leq 2^{-c \ln(\|\pi\|^{-1}e^{-1} \ln(\frac{3}{\delta}))}/\ln 2 \leq \left( \frac{\epsilon\|\pi\|}{\ln(3/\delta)} \right)^c
\]  

(13)

and therefore by Lemma 3 we obtain a \((1 \pm \epsilon)\) approximation of \(\gamma\). Note that for every \(\epsilon \in (0, 1)\) a \((1 \pm \epsilon)\) approximation of \(\pi(v)\) is given by a \((1 \pm \frac{\epsilon}{2})\) approximation of \(\gamma\) (see above). The total number of operations performed is clearly bounded by \(t = \tau c \ln(\|\pi\|^{-1}e^{-1} \ln(\frac{3}{\delta})))/\ln 2\) times the number of samples taken by SUMAPPROX, and by substituting this value in the bound of Theorem 1 we obtain Theorem 2.
Algorithm 2 MASSAPPROX(v, ϵ, δ).

1: $S \leftarrow \emptyset$  
2: $w_S \leftarrow 0$  
3: $w \leftarrow 0$  
4: $r \leftarrow 0$  
5: $k_{\epsilon, \delta} \leftarrow 4\left\lceil \frac{2+4.44\epsilon}{\epsilon^2} \ln \frac{3}{\delta} \right\rceil$  
6: $t \leftarrow \Theta(t)$
7: while $r < k_{\epsilon, \delta}$ do
8:     $w \leftarrow w + w_S$
9:     $(u, y_u) \leftarrow$ sample drawn by walking $t$ steps starting from $v$
10:    if $u \notin S$ then  
11:        $r \leftarrow r + 1$
12:    else
13:        $S \leftarrow S \cup \{u\}$
14:        $w_S \leftarrow w_S + y_u$
15: return $r/w$

4 Space Complexity Bounds

In the previous sections we focused on the running time of our algorithms, ignoring their space requirements. It is straightforward to see that SUMAPPROX and MASSAPPROX need $\Theta(|S|)$ space, where $S$ is the set of distinct elements/states drawn, assuming every element/state can be encoded with $\Theta(1)$ bits. Unfortunately, in general $|S|$ grows with $\|\pi\|^{-1}$ (see Theorem 1 and Theorem 6), and thus in the worst case with $\sqrt{n}$. Since we are interested especially in cases where $n$ is very large, this dependence might be problematic. It is thus natural to ask if we can reduce the space complexity. We show that one can reduce the space complexity of our algorithms by a factor $\alpha > 1$ while correspondingly increasing the running time, and that no algorithm can perform significantly better. Formally, we prove:

**Theorem 8** Let $s = \lceil 45\|\pi\|^{-1}\epsilon^{-3}(\ln \frac{3}{\delta})^{3/2} \rceil$ (see Theorem 5). Choose any integer $\kappa \geq 1$, and let $\alpha = \min(1, \kappa/s)$. Then one can adapt SUMAPPROX(ϵ, δ) so that at any point $|S| \leq \kappa$ and that, with probability at least $1 - \frac{\delta}{3}$, it draws at most $s/\alpha$ samples. Furthermore, if one can store at most $\kappa = o(\|\pi\|^{-1})$ distinct elements, then in the worst case $\Omega(\|\pi\|^{-2}/\kappa)$ samples are needed to estimate $\gamma$ within constant multiplicative factors with constant probability.

**Proof** For the upper bound we adapt SUMAPPROX(ϵ, δ) so that, at any point, rather than all the $|S|$ distinct entries drawn so far, it stores only the $\min(|S|, \kappa)$ entries with largest weights. Let $S$ still denote the set of all distinct entries drawn by the algorithm, and let $S' \subseteq S$ be the subset of elements that are stored. In SUMAPPROX(ϵ, δ) we
replace $S$ by $S'$, and replace lines 12-13 with the following lines ($u_1, \ldots, u_i$ denote the elements in $S'$, ordered so that $\pi(u_1) \geq \ldots \geq \pi(u_i)$):

1: if $|S'| < \kappa$ then
2: $S' \leftarrow S' \cup \{u\}$
3: $w_{S'} \leftarrow w_{S'} + \gamma_u$
4: else if $\gamma_u > \gamma_u$, then
5: $S' \leftarrow S' \cup \{u\} \setminus \{u_i\}$
6: $w_{S'} \leftarrow w_{S'} + \gamma_u - \gamma_{u_i}$

Now recall the proof of Theorem 5. If case 1 holds, then we consider in particular $u = \arg\max_{u \in V} \pi(u)$. Obviously $u \in S'$ since the first time $u$ is drawn; the whole argument then holds unchanged, and the claim follows since $s \leq s/\alpha$. If case 2 holds, then note that by construction after $\bar{s}$ draws we have $|S'| \geq \min(|\bar{S}|, \kappa)$. Moreover, since $\bar{s} \leq s$ and $\alpha \leq \kappa/s$, then $\kappa \geq \alpha s \geq \alpha \bar{s}$, and since clearly $\bar{s} \geq |\bar{S}|$, then $\kappa \geq \alpha |\bar{S}|$. Thus after $\bar{s}$ draws $|S'| \geq \alpha |\bar{S}|$, and therefore by construction its mass $w'(\bar{s}) = \sum_{u \in S'} \pi(u)$ satisfies $w'(\bar{s}) \geq \alpha w(\bar{s})$. But the proof shows that with probability at least $1 - \delta/23$ we have $w(\bar{s}) \geq 0.4 \mathbb{E}[w(\bar{s})]$, and thus $w'(\bar{s}) \geq 0.4\alpha \mathbb{E}[w(\bar{s})]$. The rest of the argument holds unchanged if we use $s/\alpha$ in place of $s$, proving the claim.

Let us move to the lower bound. Recall the proof of Theorem 6. Any entry of $x$ or $x'$ has probability $O(1/k)$ of being drawn. Thus, if one stores at most $\kappa$ elements, at any point in time the probability of the next draw yielding a repeat is bounded from above by $O(1/k)$. Therefore the expected number of draws to the first repeat is $\Omega(1/k) = \Omega(\|\pi\|^{-2}/\kappa)$. The rest of the argument follows.

5 Footprint Bounds

We have so far analysed the running time and space requirements of MASS APPROX. There is however one last performance measure we are interested in: the footprint, i.e. the total number of distinct states visited by the algorithm along the walk. This is a sensible measure when walking through a state is expensive only the first time that state is visited, e.g. in random walks over social graphs, where upon visiting a new user one needs to retrieve all its neighbors, but can then cache them for later reuse. The footprint of MASS APPROX is trivially bounded by its running time, which in turn is bounded by $\tau \|\pi\|^{-1}$ (see Theorem 2), which can be as large as $\tau \sqrt{n}$. In this section we describe FULL MASS APPROX, a variant of MASS APPROX that under certain conditions has a provably smaller footprint at the expense of memory requirements. Moreover, FULL MASS APPROX is potentially faster (in fact, in our experiments it is always much faster than MASS APPROX).

We obtain FULL MASS APPROX by making two modifications to MASS APPROX. First, instead of performing a new walk of length $t$ from $v$ for each sample, we perform a single long random walk of length $T$ and take one sample every $t$ steps. The correctness guarantees do not change, since although the samples do not come all from the same distribution, they are still drawn from a distribution sufficiently close
to $\pi$. Second, after checking if the current draw yields a repeat, we include in the set $S$ not only the draw but also all other states visited so far. This does not affect the guarantees, since (crucially) we do not need the set $S$ to be built on independent samples; but it makes the mass of $S$ grow potentially faster, which can lead to more repeats and thus a lower number of samples. The pseudocode of FULLMASSAPPROX is given below.

**Algorithm 3 FULLMASSAPPROX($\epsilon, \delta, v$).**

```
1: $S \leftarrow \{v\}$  \Comment{distinct states visited so far}
2: $D \leftarrow \{v : 1\}$  \Comment{dictionary mapping $u$ to $\gamma_u$}
3: $w_S \leftarrow 1$  \Comment{$\sum_{u \in S} \gamma_u$ for the current $S$}
4: $w \leftarrow 0$  \Comment{will accumulate $w_S$}
5: $r \leftarrow 0$  \Comment{number of repeats witnessed}
6: $k_{\epsilon,\delta} \leftarrow 4\left(\frac{2^{1+4.4\delta}}{\epsilon^2} \cdot \ln^3 \delta\right)$  \Comment{halting threshold on the number of repeats}
7: $t \leftarrow \Theta(\tau)$  \Comment{see Lemma 3}
8: $u \leftarrow v$  \Comment{current walk state}
9: \While{$r < k_{\epsilon,\delta}$}{
10: \For{$i = 1$ to $t$}{
11: \If{$\bar{u} \notin D$}{
12: \If{$\bar{u} \notin S$}{
13: $\bar{u} \leftarrow \text{step}(u)$
14: $u \leftarrow \bar{u}$
15: \Return{$r/w$}  \Comment{estimate of $1/\gamma$, i.e. of $\pi(v)$}
16: $w_S \leftarrow w_S + \sum_{u \in N} D[u]$  \Comment{detect repeat}
17: $r \leftarrow r + 1$  \Comment{new states visited}
18: $N \leftarrow N \cup \{\bar{u}\}$  \Comment{$\bar{u}$ never visited before}
19: }
20: $w \leftarrow w + w_S$
21: $S \leftarrow S \cup N$
```

Let now $N_{v,T}$ be the number of distinct states visited by a random walk of $T$ steps starting from $v$. This is in other words the number of distinct states visited by FULLMASSAPPROX over $T$ steps. We show that, if $N_{v,T}$ gets sufficiently concentrated around its expectation, then FULLMASSAPPROX terminates when $N_{v,T}$ itself is smaller than $\tau \sqrt{n}$. The concentration of $N_{v,T}$ is expressed by some function $\bar{\tau}$ of the chain; if $\Pr[N_{v,T} \notin \Theta(\mathbb{E}[N_{v,T}])] = o\left(\frac{\bar{\tau}}{\mathbb{E}[N_{v,T}]}\right)$, then the footprint of FULLMASSAPPROX is essentially $\sqrt{\bar{\tau}n}$. Ideally we would like $\bar{\tau} = \tau$, in which case the footprint becomes $\sqrt{\tau n}$, which we prove to be essentially optimal. Formally, we show:

**Theorem 9** Let $N_{v,T}$ be the number of distinct states visited by a random walk of $T$ steps starting from $v$. Assume for some function $\bar{\tau}$ of the chain we have $\Pr[N_{v,T} \notin \Theta(\mathbb{E}[N_{v,T}])] = o\left(\frac{\bar{\tau}}{\mathbb{E}[N_{v,T}]}\right)$. Then, given any $\delta, \epsilon \in (0, 1)$,
**Theorem 10** For any function $\tau(n) \in \Omega(\ln n) \cap O(n)$ there is a family of time-reversible chains on $n$ states where (a) the mixing time is $\tau = \Theta(\tau(n))$, and (b) there is a target state $v$ such that, to estimate its mass $\pi(v)$ within any constant multiplicative factor with constant probability, any algorithm requires $\Omega(\sqrt{\tau n/\ln n})$ neighbor calls.

The proof of Theorem 10 is deferred to Section 6. We shall thus prove Theorem 9, but before starting let us provide some intuition behind the concentration hypothesis. Suppose the walk runs for $T = k\bar{\tau}$ steps for some $\bar{\tau} = \tau \text{poly}(\|\pi\|^{-1})$. Such a process can be seen as a coupon collector over $k$ rounds (see e.g. [19]), where a subset of at most $\bar{\tau}$ states is collected (i.e. visited) at each round. Now, if we pick $\bar{\tau}' \leq \bar{\tau}$ with $\bar{\tau}' = \tau \text{poly}(\|\pi\|^{-1})$, then in each round the $\bar{\tau} - \bar{\tau}'$ middle steps are essentially independent of other rounds (more formally, the correlation is $O(\text{poly}(\|\pi\|))$). Each round is then in large part independent of the others; the issue is that the states visited within a single round are correlated. This correlation is responsible for the factor $\bar{\tau}$ in the concentration hypothesis and amounts for the (intuitive) fact that conditioning on the outcome of one step of the walk does not affect the distribution of those steps that are more than $\bar{\tau}$ steps away. We note that the concentration bounds of [9] give $\Pr[\sum_{i=1}^{T} f_i \neq (1 \pm \epsilon)\mathbb{E}[\sum_{i=1}^{T} f_i]] < \exp\left(-\Omega\left(\epsilon^2 \mathbb{E}[\sum_{i=1}^{T} f_i]/\tau\right)\right)$ where $f_i \in [0, 1]$ is a weight function of state $X_i$; however we could not use them to prove the concentration hypothesis of Theorem 9.

Let us now delve into the proof.

**Proof** Observe the random walk performed by **FULLMASSAPPROX**. Clearly if $\bar{\tau} = \Omega(\tau(n))$ then the walk visits $O(\tau \ln n + \sqrt{\tau n})$ distinct states, and the theorem holds unconditionally. Let us then assume $\bar{\tau} = o(\tau(n))$. We disregard the first $T_0 = \Theta(\tau \ln n)$ steps of the walk, which of course yield at most $T_0$ distinct states, and focus on the last $T$ steps, which we denote by $X_1, \ldots, X_T$ (one may thus plug $T + T_0$ in place of $T$ in the concentration hypothesis). Let $\pi_i$ denote the distribution of state $X_i$, $i = 1, \ldots, T$. Since $T_0 = \Theta(\tau \ln n)$, then we can make $||\pi_i - \pi||_{\text{TV}} \leq \frac{1}{\text{poly}(n)}$. One can adapt the proof of Lemma 3 to **FULLMASSAPPROX**, using the hypothesis $||\pi - \pi_i||_{\text{TV}} \leq \left(\frac{\epsilon \|\pi\| \ln n}{3/\delta}\right)^c$ for all $i \geq 1$. This changes the bounds of the lemma only by constant multiplicative factors. We can thus focus on proving the bound on the number of states visited by the walk. In the analysis we assume $X_i \sim \pi$, but again the same asymptotic bounds hold if $||\pi_i - \pi||_{\text{TV}} \leq \frac{1}{\text{poly}(n)}$. Let $S_{v, t} = \bigcup_{i=1}^t \{X_i\}$, let $N_{v, t} = |S_{v, t}|$, and let $M_{v, t} = \sum_{u \in S_{v, t}} \pi(u)$. For brevity we simply write $S_t, N_t, M_t$.

The crux is to show that $M_t$, the aggregate mass of $S_t$, grows basically as $N_t^2/t$. Formally we prove that, for any $\epsilon, \delta, q > 0$, if $\Pr[N_t \geq q] \geq 1 - \epsilon - \delta$ then $\Pr[M_t \geq q^2 \frac{\epsilon}{4n}] \geq 1 - \epsilon - \delta$. First, for any $\lambda > 0$ let $V_\lambda = \{u \in V : \pi(u) < \frac{\lambda}{n}\}$. Clearly $\Pr[X_t \in V_\lambda] = \sum_{u \in V_\lambda} \pi(u) < \lambda$. Therefore the number of steps $J_t(\lambda)$ the chain was on a state of $V_\lambda$ satisfies $\mathbb{E}[J_t(\lambda)] < t\lambda$. Now, by Markov’s inequality $\Pr[J_t(\lambda) \geq
and setting \( \lambda = \frac{q}{2} \), we obtain \( \Pr[J_\tau(\lambda) \geq \frac{q}{2}] < \epsilon \). Since by hypothesis \( \Pr[N_t < q] < \delta \), by a union bound we get \( \Pr[N_t \geq q, J_\tau(\lambda) < \frac{q}{2}] \geq 1 - \delta - \epsilon \). But if \( N_t \geq q \) and \( J_\tau(\lambda) < \frac{q}{2} \), then \( S_\tau \) contains at least \( \frac{q}{2} \) distinct states with individual mass at least \( \frac{\epsilon q}{2\sqrt{n}} \), and thus \( M_t \geq \frac{q^2 \epsilon}{2\sqrt{n}} = q^2 \frac{\epsilon}{4n} \).

Now choose \( t \) such that \( \mathbb{E}[N_t] = \Theta(\sqrt{n\tau}) \); note that \( \mathbb{E}[N_t] = \Omega(\bar{\tau}) \) since \( \bar{\tau} = o(n) \). By plugging \( \mathbb{E}[N_t] \) into the concentration bound for \( N_t \), we can then make \( \Pr[N_t < (1 - \bar{\epsilon})\mathbb{E}[N_t]] \) arbitrarily small for any \( \bar{\epsilon} > 0 \). Let then \( q = (1 - \bar{\epsilon})\mathbb{E}[N_t] \). By the bounds of the previous paragraph, for any \( \delta > 0 \) we have \( M_t \geq q^2 \frac{\bar{\epsilon}}{4n} = \Omega(n\bar{\tau}) \frac{\bar{\epsilon}}{4n} = \Omega(\frac{\bar{\epsilon}}{\bar{\tau}}) \). Conditioned on the event that \( M_t = \Omega(\frac{\bar{\epsilon}}{\bar{\tau}}) \), any sample drawn after \( t \) steps is a repeat with probability \( \Omega(\frac{\bar{\epsilon}}{\bar{\tau}}) \).

If we then draw \( \Theta(\frac{\bar{\epsilon}}{\bar{\tau}}) \) samples, which require \( \Theta(t) \) steps, we witness an expected \( \Omega(1) \) repeats, which can be made larger than \( k, \delta \) by appropriately increasing \( t \). Again by the concentration bounds on \( N_t \), the total number of states visited can be made \( O(2\mathbb{E}[N_t]) = O(\sqrt{n\tau}) = \tilde{O}(\sqrt{n\bar{\tau}}) \) with probability arbitrarily close to 1 by appropriately increasing \( t \).

\[ \square \]

## 6 Lower Bounds

In this section we prove the bounds of Theorem 3 and Theorem 10 (see Section 1.4), as well as the lower bound for non-time-reversible Markov chains mentioned in Section 1.

### 6.1 Proofs of Theorem 3 and Theorem 10

Both proofs follow the same line. As anticipated, the bounds are proven under a strengthened model providing a primitive \texttt{neigh}(\( u \)) that at cost \( O(1) \) returns all the incoming and outgoing transition probabilities of \( u \). We assume \texttt{neigh}(\( u \)) is invoked automatically when \( u \) is first visited, and we leave for free all subsequent \texttt{step()} and \texttt{probe()} calls on \( u \) and all elementary operations. It is clear that the cost incurred under this model is no larger than that incurred in the \texttt{step()} and \texttt{probe()} model. We see the chains as random walks on weighted undirected graphs. Recall that any undirected weighted graph \( G \) can be univocally associated to a time-reversible Markov chain: for any \( u, u' \in G \), \( p_{uu'} > 0 \) if and only if \( (u, u') \) is an edge of \( G \) with weight \( w_{uu'} = z\pi(u)p_{uu'}, \) for some constant \( z > 0 \) equal for all edges.

Consider a random \( d \)-regular expander graph \( G_0 \) on \( n_0 \) nodes (see e.g. [19]). Standard random walk theory states that the simple random walk on \( G_0 \) with mixing time \( \tau_0 = \Theta(\log_d n_0) \). Also, by a simple birthday argument, starting from any given node \( v \), any algorithm must visit \( \Omega(\sqrt{n_0}) \) nodes and thus perform \( \Omega(\sqrt{n_0}) \) queries to estimate \( n_0 \) within constant factors with constant probability. We now build our chain out of \( G_0 \). In particular, we create a (random) graph \( G = G(\Delta, n_0) \) on \( n = \Theta(n_0\Delta) \) nodes as follows. Take each arc \( \{u, v\} \) of \( G_0 \) and replace it with a star on \( \Delta + 1 \) nodes. More precisely: delete \( \{u, v\} \), add a new node \( s_{uv} \) and two arcs \( \{u, s_{uv}\} \) and \( \{v, s_{uv}\} \), and add \( \Delta - 2 \) nodes each having a self-loop and an arc to \( s_{uv} \). \( \Delta \) is a function of \( n_0 \) to be decided later. Assign weight \( d - 1 \) to each self-loop, and weight 1 to any other
arc. Let $n$ be the number of nodes in $G$; clearly $n = \Theta(n_0 \Delta)$. $G$ is now a version of $G_0$ where the random walk slows down by a factor roughly $d \Delta = \Theta(\Delta)$, since moving between two nodes $u$ and $u'$ that were neighbors in $G_0$ now takes $\Theta(\Delta)$ steps in expectation. The mixing time of $G$ is therefore $\tau = \Theta(\tau_0 \Delta)$. This holds also for an arbitrary algorithm: any two neighbors of a star center are indistinguishable until they are visited, since their transition probabilities are the same (1/$\Delta$ from the center, and 1/$d$ to the center). The same holds for the neighbors of the original nodes of $G_0$, whose transition probabilities from/to such a node are 1/$\Delta$ and 1/$d$ respectively. Therefore, starting from any node in $G$ any algorithm needs $\Omega(\sqrt{\Delta n})$ queries to estimate $n$. Finally, if $\pi$ is the stationary distribution of the random walk, then $\|\pi\| = \Theta(\sqrt{\Delta/n})$, since there are $\Delta/n$ nodes (the centers of the stars) all having the same mass which is also asymptotically larger than the mass of any other node, and which in aggregate is $\Omega(1)$.

Now consider the graph $G' = G(\Delta, \frac{n_0}{2})$, which has half the nodes of $G$. Choose any node $u \in G'$. We now add $k$ nodes to $G'$, with $k = \Theta(n)$, so that it has exactly the same number of nodes $n$ as $G$. Finally, we add $k$ arcs between each of those nodes and $u$, each one of weight $\frac{\epsilon}{k}$ for some $\epsilon > 0$. Both the overall mass of these $k$ nodes, and the probability of walking to any of them from $u$, is then less than $\epsilon$. Therefore, the mixing time of $G'$ is essentially unaltered, as well as its stationary distribution. However, any node in $G$ has roughly half the mass of its “homologue” in $G'$. Therefore, to estimate the mass of any given node in $G$, one must distinguish between $G$ and $G'$, i.e. determine whether the graph at hand comes from $G(\Delta, n_0)$ or $G(\Delta, \frac{n_0}{2})$; and as we have seen this requires $\Omega(\sqrt{\Delta n})$ queries.

Now to the bounds. Note that $\tau \|\pi\|^{-1} = \Theta(\tau_0 \Delta \sqrt{n/\Delta}) = O(\sqrt{\Delta n \ln n})$, and $\sqrt{\tau n} = \Theta(\sqrt{\Delta n \ln n}) = O\left(\sqrt{\Delta n \ln \ln n}\right)$; thus $\sqrt{\Delta n}$ is in both $\Omega(\tau \|\pi\|^{-1}/\ln n)$ and $\Omega(\sqrt{\tau n/\ln n})$. Since $\tau = \Theta(\tau_0) = \Theta(\frac{n_0}{n} \ln n_0)$ and $\Delta = \Theta(\frac{n_0}{n_0})$, by appropriately choosing $n_0 \in \Omega(1) \cap O(n)$ we can make $\tau$ range from $\Theta(\ln n)$ to $\Theta(n)$. In the same way, since $\|\pi\| = \Theta(\sqrt{\Delta/n}) = \Theta(\sqrt{1/n_0})$ we can make $\|\pi\|$ range from $\Theta(\frac{1}{\sqrt{n}})$ to $\Theta(1)$ (although not independently of $\tau$).

### 6.2 A Lower Bound for Non-Time-Reversible Markov Chains

**Lemma 4** For any functions $\tau(n) = \omega(1)$ and $p(n) = o(\frac{1}{n})$ there exists a family of ergodic non-time-reversible Markov chains on $n$ states having mixing time $\tau = \Theta(\tau(n))$, and containing a state $v$ with $\pi(v) = \Theta(p(n))$ such that any algorithm needs $\Omega(\frac{\tau}{\pi(v)})$ calls to step() to estimate $\pi(v)$ within constant multiplicative factors with constant probability.

**Proof** Consider a chain with state space $\{u\} \cup \{u_1, \ldots, u_{n-1}\}$ and the following transition probabilities (we assume $n$ large enough to set in $[0, 1]$ any quantity where needed). For $u$, set $p_{uu} = 1 - \frac{(n-1)p(n)}{\tau(n)}$, and $p_{uu} = \frac{p(n)}{\tau(n)}$ for all $i = 1, \ldots, n-1$. For all $i = 1, \ldots, n-1$, set $p_{u_i u} = 1 - \frac{1}{\tau(n)}$ and $p_{u_i u} = \frac{1}{\tau(n)}$. The chain is clearly ergodic. Note that $\frac{(n-1)p(n)}{\tau(n)} = o(\frac{1}{\tau(n)})$ and therefore the expected time to leave $u$ is asymptotically larger than the expected time to leave any of the $u_i$. One can then check
that (i) $\pi(u_i) = \Theta(p(n))$, and (ii) the mixing time is $\tau = \Theta(\tau(n))$ (essentially, the expected time to leave the $u_i$). Pick any $u_i$ as target state $v$. Suppose now to alter the chain as follows: pick some $u_j \neq v$ and set $p_{u_jv} = 1$. The new stationary probability of $v$ would then be roughly $2\pi(v)$. However one cannot distinguish between the two chains with constant probability with less than $\Omega(\frac{\tau}{\pi(v)})$ calls. Indeed, to distinguish between them one must at least visit $u_j$ (and then perform e.g. $\text{probe}(u_j, v)$). Since $u$ is the only state leading to $u_j$ with positive probability, one must invoke $\text{step}(u)$ until it returns $u_j$. But $p_{uu_j} = \frac{p(n)}{\pi(n)}$, hence one needs $\Omega(\frac{\tau(n)}{\pi(n)}) = \Omega(\frac{\tau}{\pi(v)})$ calls in expectation. The construction can be adapted to any constant approximation factor by adding more transitions towards $v$.

\section{Experiments}

We experimentally evaluate \textsc{MassApprox} and \textsc{FullMassApprox} against the algorithms of Lee et al. \cite{Lee2013} and Banerjee et al. \cite{Banerjee2015} (see Section 1.3). All algorithms were ran on synthetic time-reversible Markov chains on 1M states, created as follows. We start from an undirected torus graph (i.e. a grid with periodic boundary) of $n = 1000 \times 1000$ nodes. We then add $0.01n$ edges between random pairs of nodes, to reduce the mixing time and thus the cost (and running time) of the algorithms. We add self-loops to all nodes to ensure ergodicity. Finally, we weight the arcs according to two distributions:

- $\pi_U$ (uniform): each arc has weight 1. The resulting norm is $\|\pi_U\| = 0.001$, that is, $1/\sqrt{n}$.
- $\pi_S$ (skewed): for each arc independently we draw $X \sim U(0, 1]$ and we set the arc's weight to $1/X$. The resulting norm is $\|\pi_S\| \simeq 0.07$.

For each weighted graph, we consider the time-reversible chain of the associated random walk.

We take $v = 0$ as the target node (obviously, all nodes are equivalent). For all algorithms we set $\delta = 0.1$. For the algorithm of Banerjee et al. we set the minimum detection threshold at $\epsilon\pi(v)$, and for all other algorithms we set $\epsilon = 0.25$. One must then fix the random walk length: $l$ in our algorithms, $\ell$ in Banerjee et al., and $1/\Delta$ in Lee et al. Setting the lengths to $\simeq \tau \ln(n)$ would make all algorithms satisfy the desired guarantees. Since we do not know $\tau$, for each algorithm we proceed as follows. We initially set the length of random walks to $l = 10$. We then perform three independent executions of the algorithm. If all three executions return an estimate $\hat{\pi}(v)$ within a multiplicative factor $(1 \pm \epsilon)$ of $\pi(v)$, then we stop. Otherwise, we increase $l$ by a factor $\sqrt{2}$ and repeat. For each value of $l$ we record the average relative error $\hat{\epsilon} = \frac{|\hat{\pi}(v) - \pi(v)|}{\pi(v)}$ and the average total number of $\text{step}()$ and $\text{probe}()$ calls.

Figure 1 shows how $\hat{\epsilon}$ decreases as the number of calls increases.

\textsc{MassApprox} and \textsc{FullMassApprox} are the fastest candidates in all cases. In the uniform chain, \textsc{MassApprox} is approached by the algorithm of Banerjee et al. at high accuracies. This seems a confirmation of theory: \textsc{MassApprox} has complexity $\tilde{O}(\tau n^{-0.5})$ on a chain with uniform distribution, and the algorithm of Banerjee et
al. has complexity $\tilde{O}(\tau^{1.5}n^{-0.5})$ on the “typical” target state with mass $\pi(v) \approx 1/n$. If $\tau$ is not exceedingly large, the two complexities can translate into close performance in practice. On the other hand, FULLMASSAPPROX is neatly more efficient than previous algorithms. To obtain a fairly accurate estimate of $\pi(v)$, say $\pm 50\%$, it improves on their performance by two orders of magnitude – and possibly by more on the skewed chain. A final observation is that FULLMASSAPPROX outperforms MASSAPPROX, too, on the uniform chain. The complexity bounds we have are the same for both algorithms, but perhaps FULLMASSAPPROX takes advantage of some specific structural properties of the chain we have used, which further improves its performance.

8 Conclusions

We have given improved, optimal algorithms for approximating the stationary probability of a given state in a time-reversible Markov chain, and for approximating the sum of nonnegative real vectors by weighted sampling. Although time-reversible chains are of clear relevance, extending our results to other classes of Markov chains is an interesting open question. We have also shown that, under certain hypotheses on the chain, a simple modification of our main algorithm can reduce the total number of distinct states one needs to visit. Since this alternative algorithm appears to be the fastest one in practice, investigating it in greater detail can be a line for future research.

Acknowledgements Marco Bressan is supported in part by the ERC Starting Grant DMAP 680153, by a Google Focused Research Award, and by the “Dipartimenti di Eccellenza 2018-2022” grant awarded to the Dipartimento di Informatica at Sapienza. Enoch Peserico is supported in part by Università degli Studi di Padova under Project CAEPAE.
References

1. Alon, N., Gurel-Gurevich, O., Lubetzky, E.: Choice-memory tradeoff in allocations. Ann. Appl. Probab. 20(4), 1470–1511 (2010)
2. Auger, A., Doerr, B. (eds.): Theory of Randomized Search Heuristics: Foundations and Recent Developments, vol. 1. World Scientific Publishing Co., Inc, Singapore (2011)
3. Banerjee, S., Lofgren, P.: Fast Bidirectional Probability Estimation in Markov Models. In: Proceedings Of NIPS, pp. 1423–1431 (2015)
4. Bonacich, P., Lloyd, P.: Eigenvector-like measures of centrality for asymmetric relations. Soc. Netw. 23(3), 191–201 (2001)
5. Borgs, C., Brautbar, M., Chayes, J.T., Teng, S.: Multiscale matrix sampling and sublinear-time PageRank computation. Internet Math. 10(1-2), 20–48 (2014)
6. Borgs, C., Brautbar, M., Chayes, J.T., Teng, S.H.: A Sublinear Time Algorithm for PageRank Computations. In: Proceedings Of AW, pp. 41–53. Springer (2012)
7. Bressan, M., Peserico, E., Pretto, L.: Brief Announcement: On Approximating PageRank Locally with Sublinear Query Complexity. In: Proceedings Of ACMP, pp. 87–89 (2018)
8. Bressan, M., Peserico, E., Pretto, L.: Sublinear Algorithms for Local Graph Centrality Estimation. In: Proceedings Of IEEE FOCS, pp. 709–718 (2018)
9. Chung, K.M., Lam, H., Liu, Z., Mitzenmacher, M.: Chernoff-Hoeffding Bounds for Markov Chains: Generalized and Simplified. In: Proceedings Of STACS, pp. 124–135 (2012)
10. Freedman, D.A.: On tail probabilities for martingales. Ann. Probab. 3(1), 100–118 (1975)
11. Golub, G.H., Van Loan, C.F.: Matrix Computations. Johns Hopkins University Press, Baltimore (2012)
12. Hastings, W.K.: Monte Carlo sampling methods using Markov chains and their applications. Biometrika 57(1), 97–109 (1970)
13. Lee, C.E., Ozdaglar, A., Shah, D.: Computing the Stationary Distribution, Locally. In: Proceedings Of NIPS, pp. 1376–1384 (2013)
14. Lee, C.E., Ozdaglar, A.E., Shah, D.: Solving systems of linear equations: Locally and asynchronously. arXiv:1411.2647 (2014)
15. Levin, D.A., Peres, Y., Wilmer, E.L.: Markov Chains and Mixing Times. American Mathematical Society (2009)
16. Lofgren, P., Banerjee, S., Goel, A.: Bidirectional PageRank Estimation: from Average-Case to Worst-Case. In: Proceedings Of AW, pp. 164–176 (2015)
17. Lofgren, P.A., Banerjee, S., Goel, A., Seshadhri, C.: FAST-PPR: Scaling Personalized PageRank Estimation for Large Graphs. In: Proceedings Of ACM KDD, pp. 1436–1445 (2014)
18. Motwani, R., Panigrahy, R., Xu, Y.: Estimating Sum by Weighted Sampling. In: Proceedings Of ICALP, pp. 53–64 (2007)
19. Motwani, R., Raghavan, P.: Randomized Algorithms. Cambridge University Press, Cambridge (1995)
20. Panconesi, A., Srinivasan, A.: Randomized distributed edge coloring via an extension of the Chernoff–Hoeffding bounds. SIAM J. Comput. 26(2), 350–368 (1997)
21. Rubinfeld, R., Shapira, A.: Sublinear time algorithms. SIAM J. Discret. Math. 25(4), 1562–1588 (2011)
22. Shyamkumar, N., Banerjee, S., Lofgren, P.: Sublinear Estimation of a Single Element in Sparse Linear Systems. In: 2016 Annual Allerton Conference on Communication, Control, and Computing (Allerton), pp. 856–860 (2016)

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.