THEOREM 13 (LEZAUD [14]). *Let P be a transition matrix of a irreducible and reversible Markov chain on a finite set V, having a stationary distribution π. Let (P,π) be a irreducible and reversible Markov chain on a finite set V. Let $f : V \to \Re$ be such that $E_\pi[f] = 0$, $\|f\|_\infty \le 1$ and $0 < E_\pi[f^2] \le b^2$. Then, for any initial distribution q, any positive integer r and all $0 < \gamma \le 1$,*

$$\Pr_q\left[r^{-1}\sum_{i=1}^{r} f(X_i) \ge \gamma\right] \le e^{-\varepsilon(P)/5} S_q \exp\left(-\frac{n\gamma^2\varepsilon(P)}{4b^2(1+h(5\gamma/b^2))}\right)$$

where $\varepsilon(P) = 1 - \lambda_1(P)$, $\lambda_1(P)$ being the second largest eigenvalue of P, $S_q = \|q/\pi\|_2$ and

$$h(x) = \frac{1}{2}(\sqrt{1+x} - (1 - x/2)).$$

If $\gamma \ll b^2$ and $\varepsilon(P) \ll 1$, the bound is

$$(1+o(1))S_q \exp\left(-\frac{r\gamma^2\varepsilon(p)}{4b^2(1+o(1))}\right).$$

Using the expectation and second moment calculations from Theorem 8, we can derive the following result about the random walk based Smooth estimator. The proof is omitted.

COROLLARY 14. *Let $\varepsilon(\tilde{P})$ denote the gap between the first and second eigenvalues of the transition matrix $\tilde{P}$ of the augmented graph $\tilde{G}$. Suppose $c = \alpha d_{\text{avg}}$. If the random walk is assumed to start from the stationary distribution itself, then by using $r = \Theta\left(\frac{1}{\varepsilon(\tilde{P})}\max\left(\alpha, \frac{1}{\alpha}\right)\frac{\log(1/\delta)}{\epsilon^2}\right)$ samples, the* Smooth *estimate is a $(1\pm\epsilon)$-approximation of d_{avg} with probability $1-\delta$.*

5.4 Sample complexity of Guess&Smooth

In this section we prove that Algorithm 1 (Guess&Smooth) computes a constant factor approximation $\hat{d}$. We begin with showing that the probability of sampling a low degree node in $\mathcal{D}_{\text{d},c}$ is closely related to the ratio of c/d_{avg}.

LEMMA 15. *Let $\beta > 0$ and $c = \beta d_{\text{avg}}$, then it holds that*

$$\frac{\beta-1}{1+\beta} \le \Pr_{u\sim\mathcal{D}_{\text{d},c}}[\deg(u) \le c] \le \frac{2\beta}{1+\beta}.$$

PROOF. Observe that $\sum_{\deg(u)\le c}(\deg(u)+c) \le n(c+c)$. Therefore it holds that $\Pr[\deg(u) \le c] \le \frac{2nc}{n(d_{\text{avg}}+c)} = \frac{2\beta}{1+\beta}$.

Also note that from Markov's inequality with the uniform measure it follows that

$$|\{u : \deg(u) \ge \beta d_{\text{avg}}\}| \le n/\beta.$$

Therefore $\sum_{\deg(u)\le c}(\deg(u)+c) \ge (n - n/\beta)c$ and we have that $\Pr[\deg(u) \le c] \ge \frac{(n-n/\beta)c}{n(d_{\text{avg}}+c)} = \frac{\beta-1}{1+\beta}$. □

THEOREM 16. *For $\hat{d}$ returned by Algorithm 1 (*Guess&Smooth*) with probability at least $1-\delta$ it holds that $d_{\text{avg}}/3 \le \hat{d} < 6d_{\text{avg}}$.*

PROOF. Let $p(c) = \Pr[\deg(v) \le c]$. From the Hoeffding bound, Theorem 1, combined with the choice of r and the union bound it follows that with probability at $1-\delta$ in all $\log_2(U/L)$ iterations it holds that

$$|N/r - p(c)| \le \epsilon_0. \tag{4}$$

If $c < d_{\text{avg}}/3$, then from Equation (4) and from the r.h.s. of Lemma 15 it follows that $N/r \le p(c) + \epsilon_0 < 1/2 + \epsilon_0$, i.e., the algorithm never returns in these iterations. On the other hand, if $c \ge 3d_{\text{avg}}$, then from Equation (4) and from the l.h.s. of Lemma 15 it follows that $N/r \ge p(c) - \epsilon_0 \ge 1/2 - \epsilon_0$, i.e., the algorithm always returns in these iterations. Therefore the lowest $\hat{d}$ the algorithm returns is $d_{\text{avg}}/3$ and in the worst case it doubles c from slightly below $3d_{\text{avg}}$ to almost $6d_{\text{avg}}$ in the last iteration. □

Network	n	m	d_{avg}
SKITTER	1696415	11095298	13.1
DBLP	317080	1049866	6.62
LIVEJOURNAL	3997962	34681189	17.34
ORKUT	3072441	117185083	76.28

Table 1: Description of datasets.

5.5 Sample complexity of Combined

The following theorem demonstrates that by bootstrapping Algorithm 2, that requires a rough estimate of the average degree, with the constant factor approximation of Algorithm 1, we are able to estimate d_{avg} with high precision and few samples in any scenario.

THEOREM 17. *Let $0 < \epsilon \le 1/2$ and $0 < \delta < 1$ and $L \le d_{\text{avg}} \le U$. Then the estimate $\hat{d}$ returned by Algorithm 3 (*Combined*) satisfies $(1-2\epsilon)d_{\text{avg}} \le \hat{d} \le (1+4\epsilon)d_{\text{avg}}$ with probability at least $1-\delta$. Furthermore, the number of samples used is $O\left(\frac{\log(1/\delta)}{\epsilon^2} + \log(\frac{U}{L})\left(\log(\frac{1}{\delta}) + \log\log(\frac{U}{L})\right)\right)$.*

PROOF. From Theorem 16 it follows that $d_{\text{avg}}/6 \le \widetilde{d} \le 6d_{\text{avg}}$ holds with probability at least $1-\delta/2$. Assuming the latter, from Theorem 8 and the choice of $\widetilde{r}$ by Combined, with probability at least $1-\delta/2$ we have that $(1-2\epsilon)d_{\text{avg}} \le \hat{d} = \text{Smooth}(G,\widetilde{r},\widetilde{d}) \le (1+4\epsilon)d_{\text{avg}}$. Thus the first claim follows from the union bound.

To count the total number of samples, observe that in addition to the $\widetilde{r} = O(\frac{\log(1/\delta)}{\epsilon^2})$ samples used by Smooth, Guess&Smooth executes at most $\log_2(U/L)$ iterations, each with $\Theta(\log(1/\delta) + \log\log(U/L))$ samples. □

6. EXPERIMENTS

In this section we compare the performance of the different degree estimators empirically using four datasets of undirected networks. All the datasets were obtained from SNAP (`http://snap.stanford.edu`). Table 1 summarizes the basic statistics of the datasets. While the datasets LIVEJOURNAL and ORKUT are explicit social networks, the dataset DBLP is the co-authorship network between 3.1 million authors of computer science research papers. SKITTER, on the other hand, represents an autonomous system (AS) network, where the edges denote which AS exchanges traffic with whom using the border gateway protocol.

Algorithms and metrics. We test the following baseline algorithms in our experiments: Feige's algorithm ($\hat{d}_{\text{Feige}}$) that relies on uniform sampling, the variant of it by Goldreich and Ron [9], denoted by $\hat{d}_{\text{GR}}$, the algorithm by Motwani, Panigrahy, and Xu [18] denoted by $\hat{d}_{\text{MPX}}$ that utilizes $n^{1/2}$ samples but has better behavior (than the theoretically best algorithm in [18]) in terms of ϵ and is suggested as the one suitable for practical implementation. We also test two collision-based estimators, referred to in Section 4. Finally, we also examine the variants of our Smooth algorithms. Rather than

run the guessing version Guess&Smooth, we run our Smooth estimator using a small number of different values of the parameter c. Since the degree of the networks examined were small enough constants, we set $c \in \{0, 1, 5, 50\}$ ($c = 0$ represents the usual random walk on the network).

We present two different set of plots characterizing the performance of the estimators. The first is the normalized *mean absolute error (MAE)*, measured as $|\hat{d} - d_{\text{avg}}|/d_{\text{avg}}$. For each sample size, we compute 100 different experiments, and then compute the average mean absolute error, averaged over these 100 experiments. In order to characterize the variability of the estimates, we also compute the 10% and 90% estimates, normalized by the ground truth, for each sample size, also empirically computed over these 100 experiments.

6.1 Results

Ideal setting. Our first experiments are in the ideal setting, where we sample nodes from the corresponding distributions directly. Figure 1 presents the results for MAE in this setting for four of the algorithms first, on each of the four datasets. The names `ideal.feige` and `ideal.gr` are self explanatory, `ideal.sr.1` denotes the Smooth with $c = 1$ and `ideal.mpx` the $\hat{d}_{\mathsf{MPX}}$ estimator. The first observation is that the number of samples required is indeed small. In order to get average MAE of less than 0.1, it is enough to work with number of samples as only 0.1% of the total number of nodes. The maximum sample size used in all the experiments was 2048, and the minimum (averaged) error in each case drops to less than 2%. Beyond this sample size, the algorithms become virtually indistinguishable. The algorithms Smooth (denoted by `ideal.sr.1`), $\hat{d}_{\mathsf{Feige}}$ and $\hat{d}_{\mathsf{MPX}}$ perform essentially similarly for the datasets DBLP, LiveJournal, and Orkut, with possibly a very slight edge to Smooth. The $\hat{d}_{\mathsf{GR}}$ algorithm performs worse than the others for smaller sample sizes, but its performance improves rapidly with same size. Note that the performance of $\hat{d}_{\mathsf{Feige}}$ is indeed much better than what Theorem 3 predicts. Also, it is indeed pleasantly surprising that $\hat{d}_{\mathsf{GR}}$ does perform reasonably accurately, since the algorithm relies on an exponential degree bucketing scheme that seems tailored to theoretical bounds, not practical implementations. The Smooth estimator, with $c = 1$, is the best, with a clear edge over the others in the Skitter dataset. This is possibly because of the more heavy-tailed nature of the autonomous system degree distribution, where a larger fraction of the total volume is tied up in large degree nodes than in social networks, and so sampling from the combined distribution $\mathcal{D}_{\text{d},c}$ is beneficial.

The confidence intervals plots in Figure 2 have two lines per algorithm, corresponding to the (normalized) 10% and 90% estimates over the multiple iterations. Again, the confidence interval for Smooth is almost as tight, or strictly tighter, than the intervals generated from the other algorithms. The comparative advantage of Smooth is again best observed in the Skitter dataset.

Random walk-based implementation. Next, in Figures 3 and 4 we observe the empirical behavior of the same set of algorithms where the samples were taken from an appropriate random walk. For uniform sampling, we used the Metropolis–Hastings method with corresponding stationary distribution. In order to sample from $\mathcal{D}_{\text{d},1}$, we used the walk described in Section 5.3. In each case, the first 100 nodes of the walk were discarded as a "burn-in" period, and were not accounted for in the sampling cost. Since our aim is to actually calibrate the performance against the number of queries made to the graph, we added in samples from consecutive steps rather than choosing a node only after every "mixing-time" intervals. This introduces higher correlation among the samples, and is reflective of the setting that Theorem 13 formulates. Using this setting, the Smooth algorithm comes out as a more definitive winner in the MAE error metric. The confidence intervals of the different algorithms are more or less comparable, again with Smooth being marginally better than the rest.

6.2 Comparison with collision-based algorithms

In Figure 5 we compare the collision based estimators $\hat{d}_{\mathsf{eCol}}$, $\hat{d}_{\mathsf{nCol}}$, and $\hat{d}_{\mathsf{hit}}$ (described in Section 4), along with our candidate Smooth. Note that these collision estimators are really trying to estimate the number of edges, which is a harder problem, and we assume that they all know n, the number of nodes. Therefore, it is not a surprise that these estimators are unsuitable for the task of estimating average degree, since, at the range of sample sizes that Smooth already provides 1% error-rate, we rarely observe any collisions among the samples.

6.3 Comparison among variants of Smooth

Finally, in the random walk setting we compare among 4 different variants of the Smooth algorithm, for $c \in \{0, 1, 5, 50\}$ in Figures 6 and 7. The performance of the Smooth variants, both in terms of the normalized MAE, as well as the confidence intervals, are more or less similar for for this range of c. It is important to note that, as mentioned in Section 5, $c = 0$ itself produces a good estimate. Note that there is a inherent tension here between the mixing time of the walk, and the appropriate value of c—increasing c to make the stationary distribution closer to the $\mathcal{D}_{\text{d},\Theta(d_{\text{avg}})}$ ideal distribution might also potentially increases the mixing time, and the resulting effect on the required number of samples for a target accuracy is unclear. But based on the performances in Figures 6 and 7, we suggest using $\mathcal{D}_{\text{d},c}$ with a small constant c as an practically viable algorithm with small mixing-time and theoretically guaranteed accuracy.

7. CONCLUSIONS

In this paper we considered the natural problem of efficiently estimating the average degree of a network. We obtain estimators that provably use very few samples despite producing an arbitrary approximation to the average degree, outperforming other natural estimators for this problem. The experimental results on large real-world social networks confirm our theoretical findings. It will be interesting to see if neighbor of neighbors can be used to improve the performance further as was observed in social sampling [5]. It will also be interesting to see whether for directed graphs the in and out-degrees can be estimated using sampling distributions that are efficiently implementable by random walks.

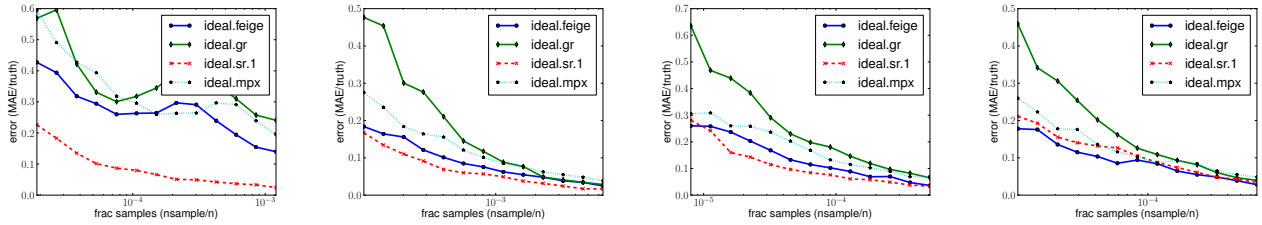

Figure 1: Normalized MAE in the ideal implementation of four algorithms for 1) SKITTER 2) DBLP 3) LIVEJOURNAL and 4) ORKUT datasets. X-axis is sample size normalized by number of nodes. `ideal.sr.1` is Smooth with $c = 1$.

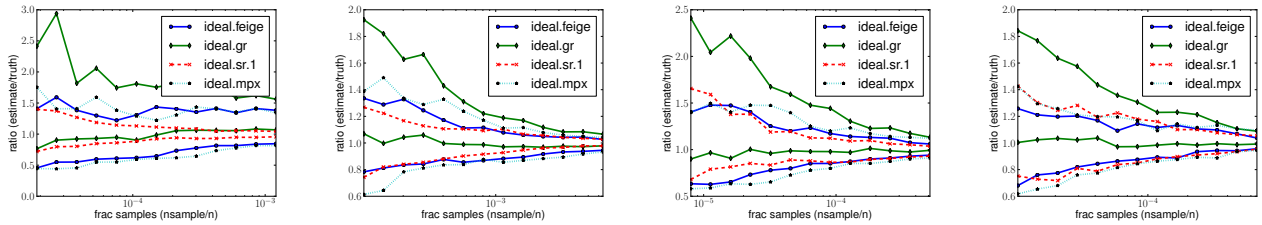

Figure 2: 10% and 90% confidence intervals in the ideal implementation of four algorithms for 1) SKITTER 2) DBLP 3) LIVEJOURNAL and 4) ORKUT datasets. X-axis is sample size normalized by number of nodes.

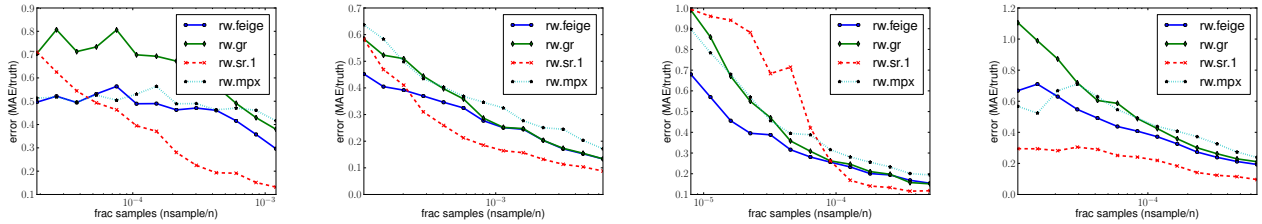

Figure 3: Normalized MAE in the random walk implementation of four algorithms for 1) SKITTER 2) DBLP 3) LIVEJOURNAL and 4) ORKUT datasets. X-axis is sample size normalized by number of nodes. `rw.sr.1` is Smooth with random walk with $c = 1$.

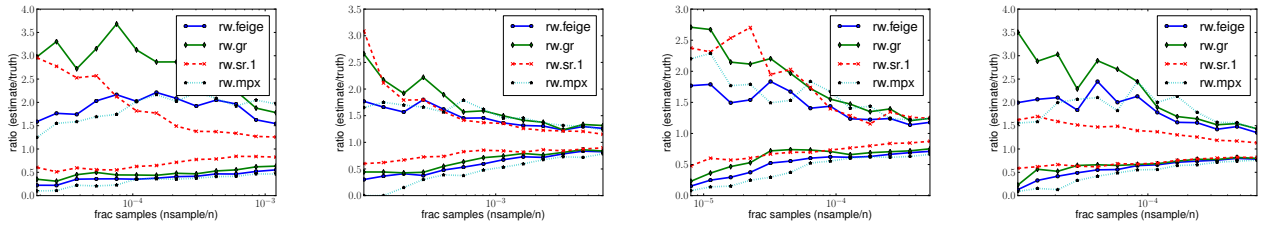

Figure 4: 10% and 90% confidence intervals in the random walk implementation of four algorithms for 1) SKITTER 2) DBLP 3) LIVEJOURNAL and 4) ORKUT datasets. X-axis is sample size normalized by number of nodes.

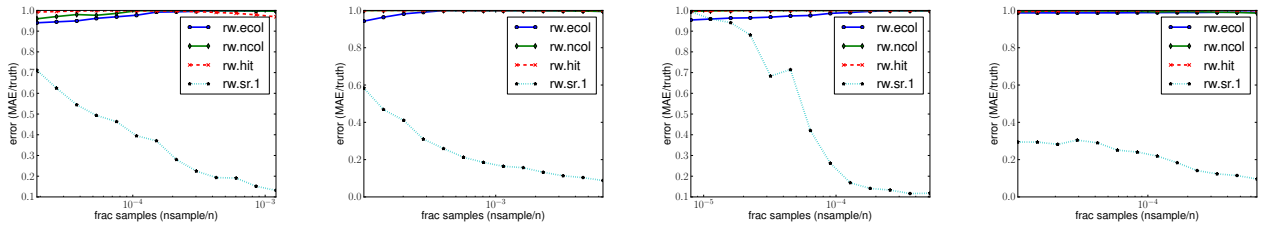

Figure 5: Normalized MAE in the random walk implementation of four collision based algorithms for 1) SKITTER **2)** DBLP **3)** LIVEJOURNAL **and 4)** ORKUT **datasets.** `rw.ecol` **is** $\hat{d}_{\text{eCol}}$**,** `rw.ncol` **is** $\hat{d}_{\text{nCol}}$ **and** `rw.hit` **is** $\hat{d}_{\text{hit}}$**.** `rw.sr.1` **is** Smooth **with** $c = 1$**, using random walk.**

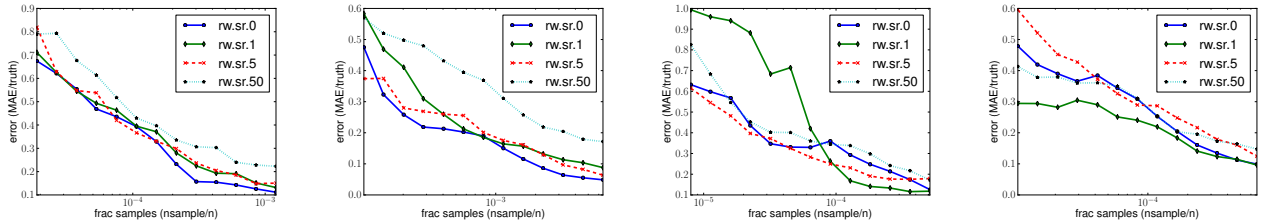

Figure 6: Normalized MAE in the random walk implementation of four Smooth **variants for 1)** SKITTER **2)** DBLP **3)** LIVEJOURNAL **and 4)** ORKUT **datasets.**

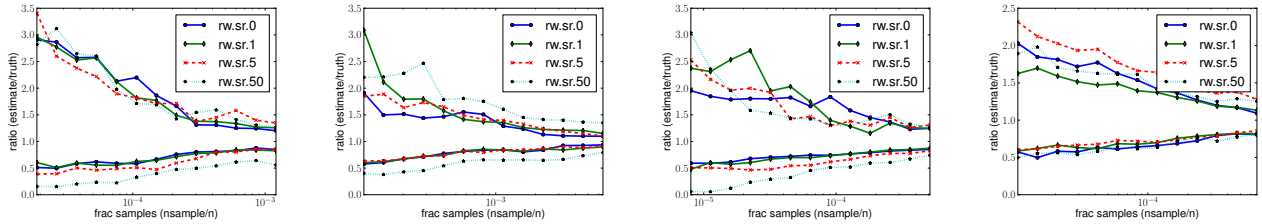

Figure 7: 10% and 90% confidence intervals of four random walk based Smooth **variants for 1)** SKITTER **2)** DBLP **3)** LIVEJOURNAL **and 4)** ORKUT **datasets.**

8. REFERENCES

[1] Z. Bar-Yossef and M. Gurevich. Random sampling from a search engine's index. *J. ACM*, 55(5), 2008.

[2] Z. Bar-Yossef and M. Gurevich. Efficient search engine measurements. *TWEB*, 5(4):18, 2011.

[3] K. Bharat and A. Broder. A technique for measuring the relative size and overlap of public web search engines. *Comput. Netw. ISDN Syst.*, 30(1-7):379–388, 1998.

[4] C. Cooper, T. Radzik, and Y. Siantos. Estimating network parameters using random walks. In *CASoN*, pages 33–40, 2012.

[5] A. Dasgupta, R. Kumar, and D. Sivakumar. Social sampling. In *KDD*, pages 235–243, 2012.

[6] D. P. Dubhash and A. Panconesi. *Concentration of Measure for the Analysis of Randomized Algorithms*. Cambridge University Press, 2009.

[7] U. Feige. On sums of independent random variables with unbounded variance and estimating the average degree in a graph. *SICOMP*, 35(4):964–984, 2006.

[8] M. Gjoka, M. Kurant, C. Butts, and A. Markopoulou. Walking in Facebook: A case study of unbiased sampling of OSNs. In *INFOCOM*, pages 1–9, 2010.

[9] O. Goldreich and D. Ron. Approximating average parameters of graphs. *RS&A*, 32(4):473–493, 2008.

[10] S. J. Hardiman and L. Katzir. Estimating clustering coefficients and size of social networks via random walk. In *WWW*, pages 539–550, 2013.

[11] L. Katzir, E. Liberty, and O. Somekh. Estimating sizes of social networks via biased sampling. In *WWW*, pages 597–606, 2011.

[12] M. Kurant, C. T. Butts, and A. Markopoulou. Graph size estimation. *CoRR*, abs/1210.0460, 2012.

[13] D. Levin, Y. Peres, and E. Wilmer. *Markov Chains and Mixing Times*. American Mathematical Society, 2009.

[14] P. Lezaud. Chernoff-type bound for finite Markov chains. *AAP*, pages 849–867, 1998.

[15] T. H. McCormick, A. Moussa, J. Ruf, T. A. DiPrete, A. Gelman, J. Teitler, and T. Zheng. A practical guide to measuring social structure using indirectly observed network data. *Journal of Statistical Theory and Practice*, 7(1):120–132, 2013.

[16] T. H. McCormick, M. J. Salganik, and T. Zheng. How many people do you know?: Efficiently estimating personal network size. *JASA*, 105(489):59–70, 2010.

[17] T. H. McCormick and T. Zheng. A latent space representation of overdispersed relative propensity in "How many X's do you know" data. In *Conf. Proc. Joint Stat. Meet.*, 2010.

[18] R. Motwani, R. Panigrahy, and Y. Xu. Estimating sum by weighted sampling. In *ICALP*, pages 53–64, 2007.

[19] A. Sinclair. *Algorithms for Random Generation and Counting: A Markov Chain Approach*. Springer, 1993.

[20] S. Ye and F. Wu. Estimating the size of online social networks. In *SocialCom*, pages 169–176, 2010.