On Influence Tail Bounds in Online Social Networks

Michael Simpson
University of Victoria
Victoria, BC, Canada
michaelesimp@gmail.com

Laks V. S. Lakshmanan
University of British Columbia
Vancouver, BC, Canada
laks@cs.ubc.ca

Venkatesh Srinivasan
Santa Clara University
Santa Clara, CA, USA
vsrinivasan4@scu.edu

Alex Thomo
University of Victoria
Victoria, BC, Canada
thomo@uvic.ca

Abstract

The influence estimation and maximization problems study the *expected* reach of a seed set in social networks under a stochastic propagation model. Motivated by the practical utility of characterizing the distribution of reach values, we systematically analyze the *tail* behaviour of the reach of a seed set. We study *tail bound query problems* that, for a given seed set, compute either the maximum reach for a given probability threshold or the highest probability of achieving a target reach. We prove #P-hardness and propose algorithms that balance efficiency and accuracy. We also examine *tail bound optimization problems* that find a seed set maximizing reach for a target probability or maximizing the probability of achieving a target reach, and establish strong inapproximability results.

Experiments on real datasets demonstrate the effectiveness of our algorithms, showing that good approximations of the actual reach for a desired probability can be computed efficiently and that the actual reach can be very different from the expected reach.

CCS Concepts

• **Information systems** → **Data mining**; • **Mathematics of computing** → **Mathematical optimization**.

Keywords

Social Network Analysis, Influence Maximization, Influence Estimation, Graph Analytics, Hardness Results

ACM Reference Format:
Michael Simpson, Laks V. S. Lakshmanan, Venkatesh Srinivasan, and Alex Thomo. 2025. On Influence Tail Bounds in Online Social Networks. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management (CIKM '25), November 10–14, 2025, Seoul, Republic of Korea.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3746252.3760910

1 Introduction

Social networks have evolved into powerful platforms for information dissemination in contexts such as viral marketing, political campaigns, and public health. The scale and speed of propagation via social ties—often termed the "word-of-mouth" effect—has enabled highly effective outreach strategies [1, 2, 6, 11, 13, 23–25].

ACM ISBN 979-8-4007-2040-6/2025/11
https://doi.org/10.1145/3746252.3760910

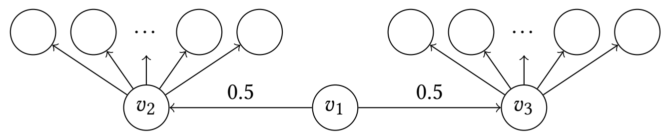

Figure 1: Network instance exhibiting large deviation in actual vs. expected reach.

Information in these networks typically spreads through stochastic cascades initiated by "seeding" selected users. Two central problems in this space are influence estimation (IE), which assesses the reach of a given seed set, and influence maximization (IM), which seeks seed sets that maximize influence. These were formalized in the seminal work of [11], and much subsequent research has focused on computing or approximating the *expected* reach of seed sets under stochastic models like Independent Cascade (IC) and Linear Threshold (LT) [4, 15, 16, 20, 29]. This line of work also includes scalable oracle-based IE techniques [15, 20], efficient IM algorithms using reverse influence sampling [10, 17, 26, 27], and robust or adaptive variants of IM [8, 9, 18, 19].

However, reliance on expected reach ignores the variance in the distribution of actual reach. This can result in seed sets with optimal expected influence but much smaller actual reach. In the network of Figure 1, all unlabelled edges have propagation probability 1, and both v_2 and v_3 connect to l leaf nodes. Node v_1 yields the highest expected reach of $l+2$ among single nodes, but with 0.25 probability, it may fail to activate v_2 and v_3, reaching only 1 node. In contrast, v_2 and v_3 each guarantee a reach of $l+1$ with probability 1, making them better choices when high certainty is preferred.

In real-world scenarios, such as a health outreach initiative by the government or an advertiser purchasing social network exposure, stakeholders need probabilistic guarantees. Questions such as *"What is the probability of reaching at least 100K users?"* or *"What is the maximum reach achievable with 80% confidence?"* cannot be answered by expected values alone. Motivated by these challenges, we present a systematic study of the *tail behavior* of the reach distribution. We study two classes of problems: *tail bound queries*, which evaluate the maximum probability of achieving a target reach or the maximum reach achievable at a given probability; and *tail bound optimization*, which search for seed sets optimizing reach or probability under probabilistic constraints. We show that tail bound queries are #P-hard, while the optimization problems are inapproximable within any constant factor unless P=NP.

Our contributions are as follows: (1) We present a systematic study of tail bound query and optimization problems as practically motivated extensions of IE and IM, focused on actual rather than expected reach. (2) We establish a suite of hardness results for these problems, and complement them with algorithmic techniques, including a novel efficient Multiway Stopping Rule Algorithm. (3) We validate our methods on real-world datasets, showing that good approximations of actual reach can be computed efficiently and that it can be much less than expected reach.

2 Preliminaries

We adopt the Independent Cascade (IC) model [11], though our methods extend to other stochastic propagation models such as linear threshold and triggering models.

Independent Cascade Model. Let $G = (V, E)$ be a directed graph with $|V| = n$, $|E| = m$, and edge probabilities $p(u, v) \in (0, 1]$. The propagation process unfolds in discrete steps: at step 0, we *activate* the seed set $S \subset V$, while setting all other nodes *inactive*. At each time step, newly activated nodes each get one chance to activate each inactive out-neighbour v with probability $p(u, v)$. The process terminates when no further activations occur. Equivalently, sample a "possible world" g by retaining each edge (u, v) independently with probability $p(u, v)$; then the final activated set $R_g(S)$ is the set of all reachable nodes from S.

Expected Influence. Define the influence of a seed set S, $\sigma(S) = \mathbb{E}_g[|R_g(S)|]$, as the expected number of nodes activated using S at termination. The Influence Maximization problem seeks a size-k seed set S maximizing $\sigma(S)$. However, $\sigma(S)$ omits distributional variance, so actual reach can be much lower than its expectation.

Stopping-Rule Algorithm. The Generalized Stopping Rule Algorithm (SRA) [5, 16] estimates $\mathbb{E}[Y]$ for $Y \in [a, b]$ within relative error ϵ and failure probability δ. It computes a threshold Γ, draws samples of Y until their sum exceeds Γ, and returns the sample average. For indicator variables ($Y \in \{0, 1\}$), [31] provides an improved threshold $\Gamma_I = (1 + \epsilon)(1 + (2 + \frac{2}{3}\epsilon)\ln(\frac{2}{\delta})\frac{1}{\epsilon^2})$. We use this improved threshold in our algorithms for tail-probability estimation. Algorithm 1 provides the details of SRA from [16].

Algorithm 1 Stopping-Rule Algorithm (*SRA*)

Input: RVs $Y_i \in [a, b]$, $\epsilon > 0$, $\delta < 1$
Output: An (ϵ, δ)-approximation of $\mathbb{E}[Y]$
if $b - a < \epsilon b$ **then**
  return $\mu_Y = a$.
Set $\epsilon' \leftarrow \epsilon(1 - \frac{\epsilon b}{(2+2\epsilon/3)\ln(2/\delta)(b-a)})$;
$\Gamma \leftarrow (1 + \epsilon)(b - a)(2 + \frac{2}{3}\epsilon')\ln(\frac{2}{\delta})\frac{1}{(\epsilon')^2}$.
Initialize $h \leftarrow 0$, $T = 0$.
while $h < \Gamma$ **do**
  $h \leftarrow h + Y_T$, $T \leftarrow T + 1$.
return $\hat{\mu}_Y = h/T$.

3 Seed Set Queries

A *seed set query* is a set of seed nodes $S \subset V$ for which a user seeks one of the following: (i) the maximum reach that S can attain for a given threshold probability η; or (ii) the maximum probability that S will attain a given target reach r, $0 \leq r \leq n$. We formally define the following problems.

Problem 1. *Given seed set S and threshold probability η, compute the maximal reach $r^* \in [0, n]$ for which* $\Pr_{g\sim G}[|R_g(S)| \geq r^*] \geq \eta$.

Problem 2. *Given seed set S and a target reach r, compute the maximal probability η^* for which* $\Pr_{g\sim G}[|R_g(S)| \geq r] \geq \eta^*$.

It is immediate that r^* decreases as η increases, and η^* decreases as r increases. [30] showed that Problem 1 is #P-hard. We show the following intractability result.

THEOREM 3.1. *Problem 2 is #P-hard.*

Remark. Although Problem 2 is #P-hard to compute exactly, one can approximate η^* to arbitrary precision via the Stopping-Rule Algorithm (SRA) if $\Pr[|R_g(S)| \geq r] > 0$, by testing connectivity in the possible-world graph. Henceforth, we focus on Problem 1 and develop three frameworks trading off efficiency and accuracy.

3.1 Query Algorithms

We present three approaches for answering tail bound queries. Given a query seed set, they leverage Monte-Carlo (MC) simulations of its reach. The first constructs an empirical CDF with *absolute* error guarantees. The other two feed indicator RVs to a Stopping Rule Algorithm (SRA), yielding stronger *relative* error guarantees.

Suppose we run s MC simulations of the reach of S and associate with the i-th simulation a random variable (RV) $X_i = |R_g(S)|$. X_i's are independent with mean $p = \sigma(S) = \mathbb{E}[|R_g(S)|]$. Thus, $\frac{1}{s}\sum_i X_i$ is an unbiased estimator of p. Our goal is to lower bound $\Pr[|R_g(S)| \geq r]$ and identify the maximal r satisfying a threshold.

Empirical CDF. If we had oracle access to the CDF F of $R_g(S)$, we could solve Problem 1 by computing $F(r) := \Pr[|R_g(S)| \leq r]$. Since this is expensive, we estimate $F(r)$ via Monte Carlo sampling to construct an empirical CDF, approximating F. We use the Dvoretzky-Kiefer-Wolfowitz (DKW) inequality [7, 14], which provides a tight uniform bound between the empirical and true CDFs.

Lemma 1. *[14] Let $Y_1, \ldots, Y_s \in [0, n]$ be i.i.d. RVs with cumulative distribution function $F(\cdot)$. Let $F_s(x) = \frac{1}{s}\sum_{i=1}^{s}\mathbf{1}_{X_i \leq x}$ with $x \in [0, n]$ denote the associated empirical distribution. Then, given a constant $\epsilon > 0$, we have* $\Pr\left[\sup_{x\in[0,n]}|F_s(x) - F(x)| > \epsilon\right] \leq 2\exp(-2s\epsilon^2)$.

This yields symmetric confidence bands: $L(x) = F_s(x) - \epsilon$, $U(x) = F_s(x) + \epsilon$. With $s \geq \frac{\ln\frac{2}{\delta}}{2\epsilon^2}$ samples, the bands enclose the true CDF with probability $1 - \delta$. Given confidence $1 - \delta$ and tolerance ϵ, Algorithm 2 draws s samples to estimate $F_s(x)$ and constructs bounds $L(x), U(x)$. It then computes a conservative lower bound r_{DKW} from the complement of $U(x)$ satisfying the η constraint, and an upper bound r^u_{DKW} from the complement of $L(x)$. Thus, with probability $\geq 1 - 2\delta$, $r^* \in [r_{DKW}, r^u_{DKW}]$, and the returned value r_{DKW} is a $\frac{r_{DKW}}{r^u_{DKW}}$-approximation of r^*. As $\epsilon \to 0$, $r_{DKW} \to r^*$.

SRA Searching. Let I_r be an indicator random variable equal to 1 if $|R_g(S)| \geq r$, and 0 otherwise. Then, $\mathbb{E}[I_r] = \Pr[|R_g(S)| \geq r]$, making I_r an unbiased estimator. Applying SRA to I_r yields an estimate μ_r for this probability. Due to SRA's relative error guarantees, if $\frac{\mu_r}{1+\epsilon} \geq \eta$, we conclude $\Pr[|R_g(S)| \geq r] \geq \eta$ with probability at least $1 - \delta$.

Algorithm 2 Empirical Cumulative Distribution Function (EmpCDF)

Input: $\epsilon > 0, \delta < 1, \eta \in [0, 1]$
Output: r_{DKW}, r^u_{DKW}
$s \leftarrow \frac{\ln(2/\delta)}{2\epsilon^2}$
Build $F_s(x)$ from s simulations; construct $L(x), U(x)$
Compute r_{DKW} from complement of $U(x)$ satisfying η
Compute r^u_{DKW} from complement of $L(x)$ satisfying η
return r_{DKW}, r^u_{DKW}

Our second approach proceeds in two phases using SRA on I_r to identify values r_1 and r_2 such that $\frac{\mu_{r_1}}{1+\epsilon} > \eta$ and $\frac{\mu_{r_2}}{1-\epsilon} > \eta$ In Phase 1, we perform a binary search over $r_1 \in [0, n]$, invoking SRA with $\delta_{r_1} = \frac{\delta}{2\ln n}$ to find the smallest r_1 satisfying the above. The union bound ensures this holds with probability $\geq 1 - \delta/2$. Phase 2 similarly searches for r_2 using the same $\delta_{r_2} = \delta_{r_1}$. We return r_1 as the solution, which satisfies $r^* \in [r_1, r_2]$ with probability at least $1 - \delta$, implying a $\frac{r_1}{r_2}$-approximation to r^*.

To improve efficiency, we enhance SRA with an *early stopping* mechanism tailored to indicator RVs and a known threshold η. During execution, SRA maintains an upper bound UB on the final estimate by assuming all future samples are 1. We scale UB by $(1 - \epsilon)$ to conservatively account for estimation error and use it as a stopping condition.

Multiway SRA. Standard SRA estimates the mean of a single random variable. In our query problem, we need to estimate the probability that S can reach r nodes for *all* possible r values. While binary search with early stopping helps, it still requires a separate SRA invocation per candidate \$r\$. To optimize further, we develop a single-pass *multiway SRA* that estimates all μ_r simultaneously.

Let $R_1, \ldots, R_s$ be Monte Carlo simulations of $|R_g(S)|$, and define $X_{r,i} = 1$ if $|R_i(S)| \geq r$, and 0 otherwise. For each fixed $r \in [0, n]$, the $X_{r,i}$ form an i.i.d. stream suitable for SRA. Crucially, since the seed set is fixed, we can update all relevant r in one pass: each sample contributes to all $X_{r,i}$ where $r \leq |R_i(S)|$.

As samples are generated, smaller r values satisfy the SRA threshold Γ sooner. For large r, where the estimated μ_r likely falls below the threshold η, we apply our early termination strategy. Once sampling completes, we identify the solution to Problem 1 by scanning up to find the largest r_u such that $\frac{\mu_{ru}}{1-\epsilon} > \eta$ and canning down to find the largest r_l such that $\frac{\mu_{r_l}}{1+\epsilon} > \eta$ We return r_l as the solution.

Algorithm 3 outlines the full procedure, beginning with computation of the improved threshold Γ_I and proceeding with sampling until early stopping is triggered.

4 Seed Set Search

In the *seed set search* problem, we are given a budget b and seek a seed set $S \subset V$ with $|S| \leq b$ that either: (i) maximizes the chance of reaching at least r nodes, or (ii) maximizes the guaranteed reach at a specified confidence level.

Problem 3. *Given budget b and target reach r, search for a seed set S of size b that maximizes the probability* $\Pr_{g\sim G}[|R_g(S)| \geq r]$.

Algorithm 3 Multiway Stopping Rule Algorithm (mwSRA)

Input: MC reach simulations R_i, $\epsilon > 0$, $\delta < 1$, $\eta \in [0, 1]$
Output: An (ϵ, δ)-approximation $\hat{\mu}_r$ for each $r \in [0, n]$
$\Gamma \leftarrow (1 + \epsilon)\left(1 + (2 + \frac{2}{3}\epsilon)\ln(\frac{2}{\delta})\frac{1}{\epsilon^2}\right), T = 0$
Initialize $h_i \leftarrow 0, UB_i = 1$ for all $i \in [0, n]$
while $\min(h_i) < \Gamma$ **do**
  for all $i \in [0, n]$ such that $h_i < \Gamma$ **do**
    $h_i \leftarrow h_i + \mathbb{I}_T$
    if $h_i \geq \Gamma$ **then**
      Record $\theta_i = T$
    $UB_i \leftarrow \frac{\Gamma}{T+\Gamma-h_i}$
  if $\max\left(\frac{UB_i}{1-\epsilon}\right) < \eta$ for i such that $h_i < \Gamma$ **then**
    break
  $T \leftarrow T + 1$
return $\{\hat{\mu}_i = h_i/\theta_i : i \in [0, n]\}$

Problem 4. *Given budget b and a target probability η, search for a seed set S of size b that maximizes the reach r such that* $\Pr_{g\sim G}[|R_g(S)| \geq r] \geq \eta$.

Unlike classical IM, these objectives are neither submodular nor supermodular, reverse sampling techniques do not apply and metrics such as submodularity ratio and curvature can approach zero [3], precluding the usual greedy-approximation guarantees. We establish the following strong hardness results:

Theorem 4.1. *Problem 3 is NP-hard to approximate within any constant factor $c \in (0, 1]$.*

Theorem 4.2. *Problem 4 is NP-hard to approximate within $1/N^{1-\epsilon}$ for any $\epsilon > 0$, where $N = |V|$.*

A simple baseline algorithm that picks any b nodes achieves approximation ratio $b/N = O(1/N)$, which matches the lower bound for Problem 4. Thus, unless $P = NP$, no polynomial-time method can outperform this trivial strategy for Problem 4.

Practical approaches. In lieu of strong guarantees, one can apply a greedy heuristic that performs well empirically: at each step, add the node that yields the largest marginal improvement in the estimated objective, using a fixed number of Monte Carlo simulations for Problem 3, or the Stopping-Rule Algorithm for Problem 4.

5 Experiments

Setup. We perform experiments on 4 real networks of varying sizes from snap.stanford.edu. *EmailEU* is an email communication network with 1K nodes and 25K edges. *Facebook* is a subset of the popular social network with 4K nodes and 88K edges. *NetHEPT* is a collaboration network in the High Energy Physics community with 15K nodes and 62K edges. *Github* is a social network of github developers consisting of 37K nodes and 290K edges. Our algorithms implemented in C++ were tested on a dual 6-core 2.10GHz Intel Xeon machine with 32GB RAM, running Ubuntu 14.04.2.

Default Parameters. As in previous works [12, 21, 22, 26, 28] we set edge probabilities for $e = (u, v)$ to $1/indegree(v)$. We use $\epsilon = 0.05$ and $\delta = 1/n$ as default for all query algorithms: Empirical CDF (DKW), SRA Search (SS) and Multiway SRA (MW).

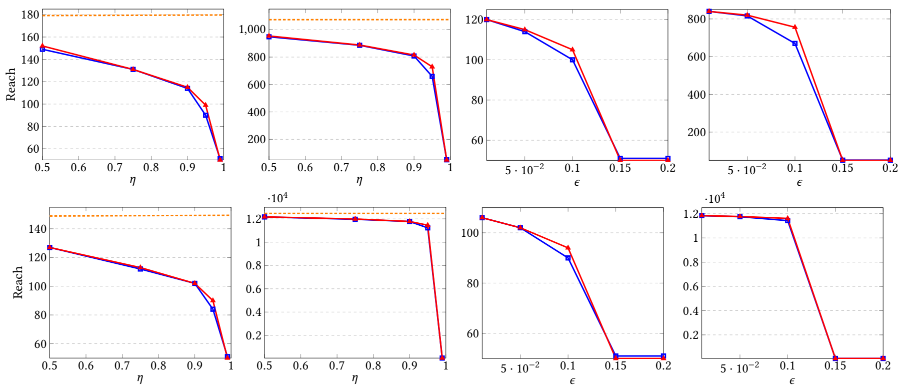

Figure 2: Reach values on NetHEPT, and Github are presented by rows, with Column 1 comparing DKW and SRA performance using 50 random seeds, and Column 2 using 50 IMM-selected seeds for different η. DKW and SRA results are similar, and quite distinct from the expected reach (dotted line). Columns 3 and 4 explore the impacts of varying ϵ in these settings.

Table 1: Runtime (sec) of algorithms DKW, MW, and SS on *EmailEU* and *Facebook*, $k = 50$.

η	EmailEU						Facebook					
	Random			Influential			Random			Influential		
	DKW	MW	SS	DKW	MW	SS	DKW	MW	SS	DKW	MW	SS
0.5	0.282	2.066	36.3	0.525	4.745	87.3	0.293	2.68	43.4	2.975	26.91	606
0.75	0.228	1.379	17.1	0.526	3.164	55.2	0.296	1.80	22.0	2.976	17.96	382
0.9	0.224	1.142	10.2	0.523	2.640	41.3	0.303	1.51	12.9	2.977	15.04	292
0.95	0.225	1.082	6.8	0.527	2.501	36.1	0.302	1.44	10.2	2.976	14.21	261
0.99	0.226	1.030	4.5	0.526	2.401	19.5	0.304	1.36	6.5	2.972	13.66	155

Table 2: Runtime (sec) on *Github*, $k = 50$. Results for SS not shown as its runtime is prohibitive for this dataset.

	Random		Influential	
η	DKW	MW	DKW	MW
0.5	0.079	1.010	9.314	82.64
0.75	0.076	0.449	8.916	55.05
0.9	0.075	0.369	8.945	46.19
0.95	0.076	0.696	9.615	44.11
0.99	0.076	0.681	9.324	41.67

	Random		Influential	
ϵ	DKW	MW	DKW	MW
0.2	0.007	0.034	0.566	4.081
0.15	0.010	0.054	1.030	6.784
0.1	0.021	0.107	2.273	12.87
0.05	0.076	0.374	9.104	45.61
0.01	2.151	8.399	224.8	1051

Results. Figure 2 reports reach versus probability threshold η (cols. 1–2) and versus error tolerance ϵ at $\eta = 0.9$ (cols. 3–4) on NetHEPT and GitHub, for 50 random seeds (cols. 1,3) and 50 IMM seeds (cols. 2,4). We show empirical CDF (absolute-error, blue), combined SRA methods (relative-error, red), and expected reach (dotted). All methods behave similarly on the other two datasets.

Accuracy. Although the CDF method uses only absolute-error guarantees, its reach estimates are competitive. SRA consistently outperforms CDF when η increases or ϵ decreases (more stringent requirements). In every case, expected reach (dotted) is overly optimistic, often substantially higher than any tail-guaranteed value.

Efficiency. Table 1 shows that EmpCDF runs up to two orders of magnitude faster than binary-search SRA, while Multiway SRA cuts search-based SRA time by a factor of 5–10, remaining under 10s on EmailEU and Facebook. On GitHub (Table 2), both EmpCDF and Multiway SRA scale well across all settings of η and ϵ showing their scalability. Similar runtime results were obtained for NetHEPT.

Overall, due to the relatively small difference in the performance of DKW and MW and the substantially improved runtime achieved by DKW, we argue that the method based on building an empirical CDF is a strong approach to answering influence tail bound queries.

6 Conclusion

We study two innovative refinements of the widely studied influence estimation and maximization problems, which avoid the practical shortcomings of the latter, whereby the actual value of the objective attained in practice may fall short of the expected value being optimized. Specifically, we analyzed two tail bound query problems, established their #P-hardness, and proposed algorithms that balanced efficiency and accuracy. We also investigated two novel seed search problems and thoroughly analysed their complexity landscape. We believe this work will yield new insights into the tail distribution of the reach in IC and IM settings.

7 GenAI Usage Disclosure

We used generative AI tools solely for language edits such as grammar, punctuation, and clarity improvement, comparable to traditional writing assistants. These tools were not used to generate original content or ideas.

References

[1] Cigdem Aslay, Wei Lu3 Francesco Bonchi, Amit Goyal, and Laks VS Lakshmanan. 2015. Viral Marketing Meets Social Advertising: Ad Allocation with Minimum Regret. *Proceedings of the VLDB Endowment* 8, 7 (2015).

[2] Cigdem Aslay, Francesco Bonchi Laks VS Lakshmanan, and Wei Lu. 2017. Revenue Maximization in Incentivized Social Advertising. *Proceedings of the VLDB Endowment* 10, 11 (2017).

[3] Andrew An Bian, Joachim M Buhmann, Andreas Krause, and Sebastian Tschiatschek. 2017. Guarantees for greedy maximization of non-submodular functions with applications. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 498–507.

[4] Wei Chen, Yifei Yuan, and Li Zhang. 2010. Scalable Influence Maximization in Social Networks Under the Linear Threshold Model. In *ICDM'10*. doi:10.1109/ICDM.2010.118

[5] Paul Dagum, Richard Karp, Michael Luby, and Sheldon Ross. 2000. An optimal algorithm for Monte Carlo estimation. *SIAM Journal on computing* 29, 5 (2000), 1484–1496.

[6] Pedro Domingos and Matt Richardson. 2001. Mining the network value of customers. In *KDD '01* (San Francisco, California). ACM, New York, NY, USA, 57–66. doi:10.1145/502512.502525

[7] Aryeh Dvoretzky, Jack Kiefer, and Jacob Wolfowitz. 1956. Asymptotic minimax character of the sample distribution function and of the classical multinomial estimator. *The Annals of Mathematical Statistics* (1956), 642–669.

[8] Daniel Golovin and Andreas Krause. 2011. Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *Journal of Artificial Intelligence Research* 42 (2011), 427–486.

[9] Keke Huang, Jing Tang, Kai Han, Xiaokui Xiao, Wei Chen, Aixin Sun, Xueyan Tang, and Andrew Lim. 2020. Efficient approximation algorithms for adaptive influence maximization. *The VLDB Journal* 29 (2020), 1385–1406.

[10] Keke Huang, Sibo Wang, Glenn Bevilacqua, Xiaokui Xiao, and Laks VS Lakshmanan. 2017. Revisiting the stop-and-stare algorithms for influence maximization. *Proceedings of the VLDB Endowment* 10, 9 (2017), 913–924.

[11] David Kempe, Jon Kleinberg, and Éva Tardos. 2003. Maximizing the spread of influence through a social network. In *KDD'03*. doi:10.1145/956750.956769

[12] Wei Lu, Wei Chen, and Laks VS Lakshmanan. 2015. From competition to complementarity: comparative influence diffusion and maximization. *Proceedings of the VLDB Endowment* 9, 2 (2015), 60–71.

[13] Wei Lu and Laks VS Lakshmanan. 2012. Profit maximization over social networks. In *2012 IEEE 12th International Conference on Data Mining*. IEEE, 479–488.

[14] Pascal Massart. 1990. The tight constant in the Dvoretzky-Kiefer-Wolfowitz inequality. *The annals of Probability* (1990), 1269–1283.

[15] Hung T Nguyen, Tri P Nguyen, NhatHai Phan, and Thang N Dinh. 2017. Importance sketching of influence dynamics in billion-scale networks. In *2017 IEEE International Conference on Data Mining (ICDM)*. IEEE, 337–346.

[16] Hung T Nguyen, Tri P Nguyen, Tam N Vu, and Thang N Dinh. 2017. Outward influence and cascade size estimation in billion-scale networks. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 1, 1 (2017), 1–30.

[17] Hung T Nguyen, My T Thai, and Thang N Dinh. 2016. Stop-and-stare: Optimal sampling algorithms for viral marketing in billion-scale networks. In *Proceedings of the 2016 International Conference on Management of Data*. 695–710.

[18] Binghui Peng. 2021. Dynamic influence maximization. In *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (Eds.), Vol. 34. Curran Associates, Inc., 10718–10731. https://proceedings.neurips.cc/paper_files/paper/2021/file/58ec72df0caca51df569d0b497c33805-Paper.pdf

[19] Binghui Peng and Wei Chen. 2019. Adaptive Influence Maximization with Myopic Feedback. In *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2019/file/f6b5f8c32c65fee991049a55dc97d1ce-Paper.pdf

[20] Gal Sadeh, Edith Cohen, and Haim Kaplan. 2020. Sample Complexity Bounds for Influence Maximization. In *11th Innovations in Theoretical Computer Science Conference (ITCS 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

[21] Michael Simpson, Venkatesh Srinivasan, and Alex Thomo. 2020. Reverse Prevention Sampling for Misinformation Mitigation in Social Networks. In *23rd International Conference on Database Theory (ICDT 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

[22] Jing Tang, Xueyan Tang, Xiaokui Xiao, and Junsong Yuan. 2018. Online processing algorithms for influence maximization. In *Proceedings of the 2018 International Conference on Management of Data*. 991–1005.

[23] Jing Tang, Xueyan Tang, and Junsong Yuan. 2016. Profit maximization for viral marketing in online social networks. In *2016 IEEE 24th International Conference on Network Protocols (ICNP)*. IEEE, 1–10.

[24] Jing Tang, Xueyan Tang, and Junsong Yuan. 2017. Profit maximization for viral marketing in online social networks: Algorithms and analysis. *IEEE Transactions on Knowledge and Data Engineering* 30, 6 (2017), 1095–1108.

[25] Jing Tang, Xueyan Tang, and Junsong Yuan. 2018. Towards profit maximization for online social network providers. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 1178–1186.

[26] Youze Tang, Yanchen Shi, and Xiaokui Xiao. 2015. Influence maximization in near-linear time: A martingale approach. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. ACM, 1539–1554.

[27] Youze Tang, Xiaokui Xiao, and Yanchen Shi. 2014. Influence Maximization: Near-optimal Time Complexity Meets Practical Efficiency. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data* (Snowbird, Utah, USA) *(SIGMOD '14)*. ACM, New York, NY, USA, 75–86. doi:10.1145/2588555.2593670

[28] Amo Tong, Ding-Zhu Du, and Weili Wu. 2018. On misinformation containment in online social networks. In *Advances in neural information processing systems*. 341–351.

[29] Chi Wang, Wei Chen, and Yajun Wang. 2012. Scalable influence maximization for independent cascade model in large-scale social networks. *Data Mining and Knowledge Discovery* 25, 3 (2012), 545–576.

[30] Peng Zhang, Wei Chen, Xiaoming Sun, Yajun Wang, and Jialin Zhang. 2014. Minimizing seed set selection with probabilistic coverage guarantee in a social network. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 1306–1315.

[31] Yuqing Zhu, Jing Tang, and Xueyan Tang. 2020. Pricing influential nodes in online social networks. *Proceedings of the VLDB Endowment* 13, 10 (2020), 1614–1627.