Influence at Scale: Distributed Computation of Complex Contagion in Networks
Brendan Lucier, Joel Oren, Yaron Singer
Microsoft Research, University of Toronto, Harvard University

Scalable estimation of the influence of individuals in massive social networks

1. Motivation

- Information requirements: how much of the graph do we need to examine? (query complexity).
- Efficient parallel computation: how to distribute computation; maintain high-volume intermediate data.

In the figure, there is a graph denoted by $G$ with nodes $a, b, c, d$ and edges $p_{ad}, p_{bd}, p_{ab}$. The graph is used to illustrate the influence model.

2. The Framework

The influence model: The Independent Cascade Model [KKT’03]
- Input: edge-weighted graph $G = (V, E, p)$, initial seeds $S_0 \subseteq V$.
- At every synchronous step $t > 0$, every node previously infected node $u \in S_t - S_{t-1}$ successfully infects an uninfected neighbor $v$ w.p. $p_{uv}$.
- $S_{t+1} = S_t \cup \{v \mid v \in S_t \text{ and } (u,v) \in E\}$, until every node in $G$ is infected.
- Influence function: $f(S_0) = E[|S_t|]$ for large enough $t$.

The MRC parallel computation model [KSV’10]
- Inspired by the MapReduce paradigm.
- Synchronous round computation on $N/(k \cdot r)$ on tuples. On every round:
  1. Map: apply local transformation on tuples in a streaming fashion.
  2. Reduce: polytime computation on aggregates of tuples with same key.
- Constraints: $N^{1-c}$ machines and space per machine; $\log^k N$ rounds.
- Graph access model: Link server.

3. The Information Requirements of Influence Estimation

- Question: how much of the graph do we need to know in order to estimate the influence?
- Concretely: what is the query complexity of approximating $f(S_0)$?

Theorem: Let $\varepsilon \in \left(0, \frac{1}{2}\right)$, $\delta \in \left(0, \frac{1}{2}\right]$. For large enough $n = |V|$, $\exists G = (V, E, p)$ for which getting an estimate $\hat{f}(S_0)$ of $f(S_0)$ s.t. $(1-\varepsilon)\hat{f}(S_0) \leq f(S_0) \leq (1+\varepsilon)\hat{f}(S_0)$ with confidence $\geq 1-\delta$ requires $O((1-2\delta)\sqrt{n})$ link server queries.

4. Algorithm: Bottom Layer; Bounded Samples

- Input: Seeds $S_0 \subseteq V$, link-server oracle $Q(\cdot)$, $L$-# samples, bound $t$.
  1. Simultaneous prob-BFS in MapReduce, one layer per round.
  2. Finally a sample when at least $t$ nodes infected.
  3. Return fraction of samples that reached bound.

Theorem (informal): 1) If $L$ is suff. small, # of tuples is $\Omega(m^{1.5}\log^4 m)$.
  2) If $\text{diam}(G) = \log^2 n$, algorithm takes polylog rounds.

5. Main algorithm: Intermediate Layer; Estimating the Expectation

Algorithm VerifyGuess: The algorithm for verifying whether a guess of influence value is correct.
- Input: An edge weighted graph $G = (V, E, w)$, initial seed set $S \subseteq V$ and guess $\tau$.
  1. for $t \in \{\tau, (1+\varepsilon)\tau, (1+\varepsilon)^2\tau, \ldots, n\}$ do
  2. Sample $L$ nodes in $G$.
  3. If $\sum_{v \in S_t} \pi_t(v) \geq (1-2\delta)\tau$ return True
  4. return False

Main intuitions:
1. High/low influence $\Rightarrow$ few/many samples needed,
2. Can compute the Riemann integral for CDF $\int_{f(S_0)}^{inf} 1 - F(I) \text{ d}I$

Lemma (informal): w.h.p., VerifyGuess returns True if $\tau > \text{influence}$.

6. Main Algorithm: Top Layer; Iterating on Candidate Values

Algorithm InfEst: The approximation algorithm for estimating the spread for the independent cascade model.
- Input: An edge weighted graph $G = (V, E, w)$, initial seed set $S \subseteq V$, precision $\epsilon$.
  1. for $\tau \in \{n, n/(1+\epsilon), n/(1+\epsilon)^2, \ldots, |S|\}$ do
  2. If VerifyGuess($G, S, \tau$) = 1 return $\tau$
  3. return 1

Theorem: For any $\epsilon \in (0, \frac{1}{2})$, InfEst provides a $(1 + 8\epsilon)$-approx. w.h.p.

7. Experimental Results

- Running time
- Approx. ratios (Epinions)
- Heuristics (Epinions)