Influence Function Learning in Information Diffusion Networks

Nan Du  Yingyu Liang  Maria-Florina Balcan  Le Song
Georgina Institute of Technology

Motivation
- Problem: Given a set of influential earlier users, can we predict how many people will follow them in the future?

Previous Two-Stage Solutions
- Algorithm
  - Learn one of the following diffusion models
    - Discrete-Time Independent cascade Model (DIC)
    - Linear Threshold Model (LT)
    - Continuous-Time independent cascade Model (CIC)
  - Calculate the influence from the chosen model
- Weakness
  - The diffusion model may be misspecified.
  - Need to learn both hidden networks and model parameters.
  - Influence calculation is challenging.

Can we avoid diffusion model learning & influence computation?

Influence Function
- Definition: \( \sigma(S) : 2^V \rightarrow \mathbb{R} \) of a set of nodes \( S \subseteq V \), \( |V| = d \)
- \( \sigma(S) \) is the expected number of infected nodes by set \( S \).
- \( \sigma(S) \) is common to many diffusion models.

Property: \( \sigma(S) \) is a coverage function for DIC, LT and CIC model
- \( \sigma(S) = \sum_{i \in S} \mathbb{I}(a_i > u) \)
- a ground set \( U \) with weight \( a_i \geq 0, u \in U \)
- a collection of subsets \( \{A_i : A_i \subseteq U\} \) associated with each \( s \in V \)

Random Reachability Function
- View the diffusion process as a node reachability problem in a random graph sampled from a joint distribution induced by a diffusion model.
- Represent each sample \( s \) as a binary reachability matrix with \( R_{sd} = \begin{cases} 1, & j \text{ is reachable from source } s, \\ 0, & \text{otherwise}. \end{cases} \)
- Denote each set \( S \) as a binary vector \( \chi_S \in \{0,1\}^d, \chi_S(s) = 1 \iff s \in S \)
- Determine the reachability of node \( j \) from \( S \) by whether \( \chi_S \mathbb{I}(R_{j}) \geq 1 \)
- Transform \( \chi_S \mathbb{I}(R_{j}) \) into a binary function \( \phi(\chi_S \mathbb{I}(R_{j})) : 2^V \rightarrow \{0,1\} \)
- Denote the influence of \( S \) in \( \phi \) as \( \#(S) = \sum_{j=1}^{d} \phi(\chi_S \mathbb{I}(R_{j})) \).

Expectation of Random Reachability Functions
- Overall influence function
  \[ \mathbb{E}(|S| \mathbb{I}(R_{j})) = \sum_{j=1}^{d} \mathbb{E}(\phi(\chi_S \mathbb{I}(R_{j}))) = \sum_{j=1}^{d} \Pr(\phi(\chi_S \mathbb{I}(R_{j})) = 1 | \mathbb{S}) \]
- Simple Learning Strategy
  Learn each \( f(\chi_S) \) separately in parallel and sum them together.

Random Basis Function Approximation
- Denote \( f(\chi_S) = \mathbb{E}(\phi(\chi_S \mathbb{I}(R_{j}))) \) where \( r = R_{j} \), and \( p(r) \) is the marginal distribution of column \( j \) of \( R \) induced by \( P_R \).
- Let \( C \) be the minimum value such that \( p(r) \leq C q(r) \)
- Draw \( K \) random binary vectors \( \{t_1, t_2, \ldots, t_K\} \) from \( p(r) \) such that
  \[ f(\chi_S) = \sum_{k=1}^{K} w_k \phi(\chi_S) = w^T \phi(\chi_S) \]
- subject to \( \sum_{k=1}^{K} w_k = 1, w_k \geq 0 \)

Lemma
- Let \( p_i(\chi_S) \) be a distribution of \( \chi_S \).
- If \( K = O(\frac{\log \Delta_S}{\epsilon^2}) \) and \( t_1, \ldots, t_K \) are drawn i.i.d. from \( p_i(\chi_S) \), then with probability at least \( 1 - \delta \), there exists an \( f^* \in \mathcal{F} \) such that \( \mathbb{E}[f^*(\chi_S) - f^*(\chi_S)]^2 \leq \epsilon^2 \)
- Propose \( q(r) = \frac{1}{|S|} \sum_{i=1}^{K} q(r_i) \) where \( q(r_i) \) is the marginal distribution of the \( i \)-th dimension of \( r \) estimated by \( q(r_i) = \frac{1}{|S|} \sum_{i=1}^{K} q_i(r_i) \)

Efficient Learning Algorithm
- Truncate \( f^* \) to avoid zero probability \( f^*(\chi_S) = (1 - 2\lambda)f^*(\chi_S) + \lambda \) is a small threshold value.
- Draw \( m \) i.i.d. cascades \( D^{(m)} := \{(S_1, I_1), \ldots, (S_m, I_m)\} \) with source set \( S_i \) and the respective set of influenced nodes \( I_i \)
- Learn the parameters \( w \) by maximizing the log-likelihood for each node \( j \)
  \[ w = \arg\max_{w} \left( \sum_{i=1}^{m} \log f^*(\chi_S) + \log(1 - \chi_S) \right) \]
- subject to \( \sum_{j=1}^{d} w_j = 1, w_j \geq 0 \)

by using convex optimization techniques.

Overall Algorithm: Influer

Experimental Evaluation: Competitors
- Continuous-time Independent Cascade model with exponential pairwise transmission function (CIC).
- Continuous-time Independent Cascade model with exponential pairwise transmission function and given network Structure (CIC-S).
- Discrete-time Independent Cascade model (DIC).
- Discrete-time Independent Cascade model with given network Structure (DIC-S).
- Modified Logistic Regression
- Linear Regression

Experimental Evaluation: Synthetic Datasets
- Robustness to model mis-specifications

Experimental Evaluation: Real Data
- MAE on real data
- Effect of random features

Sample Complexity
- Suppose we set \( \lambda = O(\frac{1}{d}), K = O(\frac{1}{\epsilon^2}), \) and \( m = O(\frac{1}{\epsilon^2}) \). Then with probability at least \( 1 - \delta \) over the drawing of the random features, the output of Algorithm 1 satisfies \( \mathbb{E}[\|\mathbb{E}_R[\sum_{j=1}^{d} f^*(\chi_S) - \mathbb{E}(S)]^2 \leq \epsilon \]

Intuitively, when the gap \( C \) between \( p \) and \( q \) is large, we need more random features and more training data to learn the weights.