Deep Learning Approaches to Random-Noise k-Topic Influence Maximization in Social Networks

Tan D. Tran[1], Second Author[2], and Third Author[3]

[1] tan.trandinh@phenikaa-uni.edu.vn
[2] ABC
@gmail.com
[3] CFD
@gmail.com

Abstract. The abstract should briefly summarize the contents of the paper in 150–250 words.

Keywords: First keyword · Second keyword · Another keyword.

1 Introduction

2 Related works

3 Preliminaries

Given a ground set $V = \{e_1, e_2, \ldots, e_n\}$ and an integer k, we define $[k] = \{1, 2, \ldots, k\}$ and let $(k+1)^V = \{(V_1, V_2, \ldots, V_n) | V_i \subseteq V \ \forall i \in [k], V_i \cap V_j = \emptyset \ \forall i \neq j\}$ be a family of k disjoint subsets of V, called k-**set**.

For $\mathbf{x} = (X_1, X_2, \ldots, X_k) \in (k+1)^V$, we define $supp_i(\mathbf{x}) = X_i$, $supp(\mathbf{x}) = \cup_{i \in [k]} X_i$, X_i as i**-th set of x** and an empty k-set $\mathbf{0} = (\emptyset, \ldots, \emptyset)$. We set if $e \in X_i$ then $\mathbf{x}(e) = i$ and i is called the **position** of e in $\mathbf{x}$, otherwise $\mathbf{x}(e) = 0$. Adding an element $e \notin supp(\mathbf{x})$ into X_i can be represented by $\mathbf{x} \sqcup (e, i)$. We also write $\mathbf{x} = \{(e_1, i_1), (e_2, i_2), \ldots, (e_t, i_t)\}$ for $e_j \in supp(\mathbf{x}), i_j = \mathbf{x}(e_j), \forall 1 \leq j \leq t$. When $X_i = \{e\}$, and $X_j = \emptyset, \forall j \neq i$, $\mathbf{x}$ is denoted by (e, i).

For $\mathbf{x} = (X_1, X_2, \ldots, X_k), \mathbf{y} = (Y_1, Y_2, \ldots, Y_k) \in (k+1)^V$, we denote by $\mathbf{x} \sqsubseteq \mathbf{y}$ iff $X_i \subseteq Y_i \ \forall i \in [k]$. For simplicity, we assume that f is non-negative, i.e., $f(\mathbf{x}) \geq 0$ for all $\mathbf{x} \in (k+1)^V$ and normalized, i.e., $f(\mathbf{0}) = 0$.

The objective function. The function $f : (k+1)^V \mapsto \mathbb{R}_+$ is k-**submodular** iff for any $\mathbf{x} = (X_1, X_2, \ldots, X_k)$ and $\mathbf{y} = (Y_1, Y_2, \ldots, Y_k) \in (k+1)^V$, we have:

$$f(\mathbf{x}) + f(\mathbf{y}) \geq f(\mathbf{x} \sqcap \mathbf{y}) + f(\mathbf{x} \sqcup \mathbf{y}) \tag{1}$$

where

$$\mathbf{x} \sqcap \mathbf{y} = (X_1 \cap Y_1, \ldots, X_k \cap Y_k),$$

and

$$\mathbf{x} \sqcup \mathbf{y} = (Z_1, \ldots, Z_k), \text{ where } Z_i = X_i \cup Y_i \setminus (\bigcup_{j \neq i} X_j \cup Y_j).$$

In this work, we consider f is *monotone*, i.e., for any $\mathbf{x} \in (k+1)^V, e \notin supp(\mathbf{x})$ and $i \in [k]$, we have the *marginal gain* when adding an element e to the i-set X_i of $\mathbf{x}$ nonnegative:

$$\begin{aligned}\Delta_{(e,i)} f(\mathbf{x}) =& f(X_1, \ldots, X_{i-1}, X_i \cup \{e\}, X_{i+1}, \ldots, X_k) \\ &- f(X_1, \ldots, X_k) \geq 0.\end{aligned}$$

We assume that there exists an *oracle query*, which, when queried with the k-set $\mathbf{x}$ returns the value $f(\mathbf{x})$. We recap some properties of the k-submodular function that will be used for designing our algorithms. From [4], the k-submodularity of f implies the *orthant submodularity*, i.e.,

$$\Delta_{(e,i)} f(\mathbf{x}) \geq \Delta_{(e,i)} f(\mathbf{y}) \tag{2}$$

for any $\mathbf{x}, \mathbf{y} \in (k+1)^V, e \notin supp(\mathbf{y}), \mathbf{x} \sqsubseteq \mathbf{y}$ and $i \in [k]$; and the *pairwise monotonicity*, i.e., for any $i, j \in [k], i \neq j$:

$$\Delta_{(e,i)} f(\mathbf{x}) + \Delta_{(e,j)} f(\mathbf{x}) \geq 0 \tag{3}$$

k-topic Independent Cascade with Random Parameter Noise (k-IC–RN).

Consider a directed social network $G = (V, E)$ and an integer $k \geq 2$ denoting the number of topics. For every edge $(u, v) \in E$ and every topic $i \in [k]$ we are given a *baseline propagation probability* $\hat{p}^i_{u,v} \in [0, 1]$. Before diffusion starts these probabilities are perturbed by independent additive noise

$$p^i_{u,v} = \operatorname{clip}\big(\hat{p}^i_{u,v} + \varepsilon^i_{u,v}\big), \qquad \varepsilon^i_{u,v} \sim \mathcal{U}[-\delta, \delta],$$

where $\delta \in (0, 1)$ is a fixed noise bound and $\operatorname{clip}(x) = \min\{1, \max\{0, x\}\}$ ensures $p^i_{u,v} \in [0, 1]$.

Diffusion dynamics. A *seed assignment* is a vector $\mathbf{s} \in (k+1)^V$; for each topic i the set $\operatorname{supp}_i(\mathbf{s}) = \{v \in V : \mathbf{s}(v) = i\}$ is activated at step 0. Once the noisy probabilities $\{p^i_{u,v}\}$ are fixed, the diffusion for every topic proceeds independently as in the classic Independent Cascade model:

1. When a vertex u becomes active for topic i at step t, it receives a single chance to activate each inactive neighbour v with probability $p^i_{u,v}$.
2. Regardless of success, u never attempts to activate v again.
3. The diffusion for topic i terminates when no further activations are possible.

Noisy influence spread. Let $A_i(\operatorname{supp}_i(\mathbf{s}); \boldsymbol{\varepsilon})$ denote the set of vertices activated by topic i under the noise realisation $\boldsymbol{\varepsilon} = \{\varepsilon^i_{u,v}\}$. The expected multi-topic spread in the presence of noise is

$$\sigma_{\mathrm{RN}}(\mathbf{s}) = \mathbb{E}_{\boldsymbol{\varepsilon}}\Big[\big|\bigcup_{i \in [k]} A_i(\operatorname{supp}_i(\mathbf{s}); \boldsymbol{\varepsilon})\big|\Big].$$

Optimisation task. Given a budget $B \in \mathbb{N}$, the *k-topic Influence Maximisation under Random Noise* problem is

$$\max_{\mathbf{s} \in (k+1)^V,\ |\operatorname{supp}(\mathbf{s})| \le B} \sigma_{\mathrm{RN}}(\mathbf{s}), \qquad \text{where } \operatorname{supp}(\mathbf{s}) = \textstyle\bigcup_{i \in [k]} \operatorname{supp}_i(\mathbf{s}).$$

Setting $\delta = 0$ recovers the classical k-IC model.

4 Methodology

4.1 k-TIE: Deep learning k-Topics Influence Estimation

In this section, we introduce k-TIE, a GNN model that aims to learn how to estimate the influence of seed set $\mathbf{s} = \{S_1, S_2, \ldots, S_k\}$ over a graph $G = (V, E)$. Let $A^{(i)} \in \mathbb{R}^{n \times n}$ be the adjacency matrix corresponding to the influence of topic i, and $X \in \mathbb{R}^{n \times (k+1)}$ be the features of nodes, where each node is marked as belonging to one of the seed sets $S_1, S_2, \ldots, S_k$:

$$X_u = \begin{bmatrix} \mathbb{I}[u \in S_1] \\ \mathbb{I}[u \in S_2] \\ \vdots \\ \mathbb{I}[u \in S_k] \end{bmatrix} \tag{4}$$

For each topic i, we normalize the adjacency matrix $A^{(i)}$ by each row, forming a row-stochastic transition matrix:

$$A^{(i)}_{uv} = \begin{cases} \frac{p^{(i)}_{uv}}{\sum_{v' \in \mathcal{N}(u)} p^{(i)}{}_{uv'}}, & v \in \mathcal{N}(u) \\ 0, & v \notin \mathcal{N}(u) \end{cases}, \tag{5}$$

Based on the weighted cascade model [2], each row u stores the probability of node u being influenced by each of the other nodes that are connected to it by a directed link $v \to u$.

We assume that for each node u, only one topic will influence it, and that will be the topic with the highest influence probability. This means we compute the influence probability of node u for each topic and then select the topic with the maximum value:

$$\mathrm{Topic}(u) = \arg\max_{i \in [k]} p(u|S_i), \tag{6}$$

where $p(u|S_i)$ represents the probability of node u being influenced by seed set S_i under topic i. The corresponding influence probabilities are computed for each topic i using the adjacency matrix $A^{(i)}$.

We can use message passing to compute a well-known upper bound $\hat{p}(u|S_i)$ of the real influence probability for node u with i-th topic:

$$\begin{aligned}
\hat{p}(u|S_i) &= A_u^{(i)} \cdot X^i && (7)\\
&= \sum_{v \in \mathcal{N}(u) \cap S_i} p_{vu}^{(i)} && (8)\\
&\geq 1 - \prod_{v \in \mathcal{N}(u) \cap S_i} (1 - p_{vu}^{(i)}) && (9)\\
&= p(u|S_i), && (10)
\end{aligned}$$

where the second equality stems from the definition of the weighted cascade and the inequality comes from the proof in [5].

As the diffusion covers more than one hop, we repeat the multiplication to approximate the total influence spread. If we let $H_1^{(i)} = A^{(i)} \cdot X^i$, where X represents the initial node features, and we assume the new seed set $\mathbf{s}^t$ to be the nodes influenced in step $t-1$, their probabilities are stored in $H_t^{(i)}$. The diffusion process can be approximated iteratively as follows:

$$H_{t+1}^{(i)} = A^{(i)} \cdot H_t^{(i)}, \tag{11}$$

where $A^{(i)} \cdot H_t$ computes the influence from the i-th topic.

In our neural network architecture, each layer consists of a GNN with batch-norm and dropout omitted here, and starting from $H_0^{(i)} = X^i \in \mathbb{R}^{n \times k}$ we have:

$$H_{t+1}^{(i)} = \mathrm{ReLU}([H_t^{(i)}, A^{(i)} H_t^{(i)}] W_t^{(i)}). \tag{12}$$

The readout function that summarizes the graph representation based on all nodes' representations is a summation with skip connections:

$$H_{\mathbf{s}}^G = \sum_{v \in V} \max_{i \in [k]} H_t^{(i),v} \tag{13}$$

Theorem 1. *The repeated product $H_{t+1}^{(i)} = A^{(i)} \cdot H_t^{(i)}$ computes an upper bound to the real influence probabilities of each infected node at step $t+1$.*

Proof.

$$\begin{aligned}
\hat{p}(u|S_i) &= A_u^{(i)} \cdot H_t^{(i)} && (14)\\
&= \sum_{v \in \mathcal{N}(u) \cap S^t} \hat{p}_v p_{vu}^{(i)} && (15)\\
&\geq \sum_{v \in \mathcal{N}(u) \cap S^t} p_v p_{vu}^{i} && (16)\\
&\geq 1 - \prod_{v \in \mathcal{N}(u) \cap S^t} (1 - p_v p_{vu}^{(i)}) && (17)\\
&= p(u|S_i). && (18)
\end{aligned}$$

– (16) follows from the definition in Eq. (10).
– (18) can be proved by induction similar to [5].

Diffusion Process and Cycles. In real-world graphs, cycles can cause two problems. Firstly, repeated diffusion may increase the influence of the original seed nodes, which can contradict the independent cascade model. To mitigate this, we limit the number of diffusion steps (e.g., three layers).

Secondly, cycles cause influence probabilities to interact, which means that the product of complementary probabilities in Eq. (18) doesn't factorize for non-independent neighbors. This was extensively analyzed in [3], and it was shown that the influence probability computed by $p(u|S)$ is an upper bound on the real influence probability in graphs with cycles. We can thus contend that the estimation $\hat{p}(u|\mathbf{s})$ provides an upper bound on the real influence probability, which can be used to approximate the influence spread.

Neural Network Architecture. In our neural network architecture, each layer consists of a GNN with message passing. We start with $H_0^{(i)} = X^i \in \mathbb{R}^{n\times k}$ and compute the next layer $H_{t+1}^{(i)}$ as:

$$H_{t+1}^{(i)} = \mathrm{ReLU}([H_t^{(i)}, A^{(i)}H_t^{(i)}]W_t^{(i)}), \tag{19}$$

where A^i corresponds to the adjacency matrix for each topic. The readout function summarizes the graph representation based on the node activations:

$$H_{\mathbf{s}}^G = \sum_{v\in V} \max_{i\in[k]} H_t^{(i),v} \tag{20}$$

The final predicted influence spread is given by:

$$\hat{\sigma}(S) = \mathrm{ReLU}(H_S^G W_o). \tag{21}$$

Loss Function. Our loss function is a simple least squares regression. Initially, the weight matrices W_t are untrained and initialized randomly. These weight matrices are updated during training to reduce the upper bound on the influence probability and produce more accurate estimates. The final layer W_o combines these estimates to provide the cumulative predicted influence spread.

Theorem 2. *The influence spread σ^m is submodular and monotone.*

Proof. Monotonocity, $\forall i < j, S_i \subset S_j$:

$$\mathcal{S}(X_j) \supset \mathcal{S}(X_i) \Rightarrow \mathcal{S}(X_jW) \supseteq \mathcal{S}(X_iW) \tag{22}$$

$$\mathcal{S}(AX_jW) \supseteq \mathcal{S}(AX_iW) \Rightarrow \mathcal{S}((R(AX_jW) - b_{tr})/st_{tr})$$

$$\supseteq \mathcal{S}((R(AX_iW) - b_{tr})/st_{tr}) \tag{23}$$

$$\mathcal{S}(H_j) \supseteq \int(H(S_i)) \Rightarrow \mathcal{S}(H_jP) \supseteq \mathcal{S}(H_iP) \tag{24}$$

$$|1_{>0}\{H_jP\}| \geq |1_{>0}\{H_iP\}| \Rightarrow \sigma^m(S_j) \geq \sigma^m(S_i) \tag{25}$$

(22) First subset is by definition. Second is because X_j is a convex hull that contains X_i [1]. We multiply both sides by a real matrix $W \in \mathbb{R}^{d\times hd}$

which can equally dilate both convex hulls in terms of direction and norm. This transformation cannot change the sign of the difference between the elements of X_i and X_j and hence cannot interfere with the support of X_j over X_i. This becomes more obvious for $X \in \{0,1\}^{n\times 1}$ and $W \in \mathbb{R}^{1\times 1}$. Note that both can result in zero matrices so we use subset or equal. (23) A is a non-negative matrix and ReLU is a nonnegative monotonically increasing function. Subtract by the same number and divide by the same positive number in the right inequality. (24) Definition in Eq. (12); P is positive. (25) By definition of the support and of L'_S.

References

1. Boyd, S., Boyd, S.P., Vandenberghe, L.: Convex optimization. Cambridge university press (2004)
2. Kempe, D., Kleinberg, J., Tardos, É.: Maximizing the spread of influence through a social network. In: KDD (2003)
3. Lokhov, A.Y., Saad, D.: Scalable influence estimation without sampling. CoRR (2019)
4. Ward, J., Živný, S.: Maximizing Bisubmodular and k-Submodular Functions, pp. 1468–1481. https://doi.org/10.1137/1.9781611973402.108, https://epubs.siam.org/doi/abs/10.1137/1.9781611973402.108
5. Zhou, C., Zhang, P., Zang, W., Guo, L.: On the upper bounds of spread for greedy algorithms in social network influence maximization. IEEE TKDE **27**(10), 2770–2783 (2015)