Deep Graph Representation Learning and Optimization For Influence Maximization

Chen Ling, Junji Jiang, Junxiang Wang, My T. Thai, Lukas Xue, Meikang Qiu, and Liang Zhao

The 40th INTERNATIONAL CONFERENCE ON MACHINE LEARNING

Full Paper Code

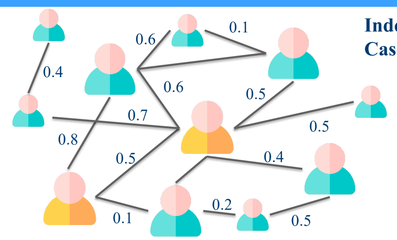
EMORY UNIVERSITY
Department of Computer Science

1. Abstract

Influence maximization (IM), is formulated as selecting a set of initial users to maximize the expected number of influenced users, has seen notable improvements through traditional methods. Yet, their design and performance are near their limits. Recent learning-based IM methods provide enhanced generalization to unknown graphs but face fundamental challenges, including: 1) efficient objective function resolution; 2) representation of varied diffusion patterns; and 3) adaptation to node-centrality-constrained IM variants. To address these, we introduce a novel framework uses latent representation of seed sets to learn the diverse diffusion pattern in a data-driven, end-to-end manner. Additionally, we formulate a new objective function to infer ideal seed sets under flexible node-centrality budget constraints.

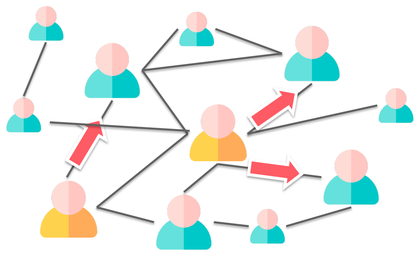
2. Challenges and Motivation

Traditional IM solutions achieve great performance with approximation ratio guarantee, but their solutions are less flexible since they need explicit diffusion model as input.



Independent Cascade (IC)

Learning-based IM methods are proposed to tackle the **NP-Hard** problem. They improve the solution generalization ability but also bring difficulties in 1) objective solving efficiency, 2) identifying diffusion process, and 3) adapting their solutions under various constraints.



Representation Learning

Reinforcement Learning

3. Our Solution: DeepIM

Key Insights. In this paper, we propose a novel framework **DeepIM** to tackle the IM problem in a more robust and generalized way than existing learning-based IM methods.

1. To characterize the complex nature of the seed set, we propose to character the ***probability*** of the seed set and directly search for a more optimal seed set in continuous space.
2. We further design a learning-based diffusion model with ***knowledge distillation*** to characterize the underlying diffusion dynamics in an end-to-end manner.
3. we develop a generic seed set inference framework to ***directly optimize and generate*** set embeddings under a uniform budget constraint in order to meet various IM variant problems.

4. Training Phase

Problem Formulation. Given a graph $G = (V, E)$. The *seed node set* is defined over V as $\boldsymbol{x} = \{0,1\}^{|V|}, x_i = 1$ or 0 denotes seed node or not. The *total number* of infected nodes is denoted as $y \in \mathbb{R}^+$. IM aims at selecting $\tilde{\boldsymbol{x}} = \arg\max_{|\boldsymbol{x}| \le k} M(\boldsymbol{x}, G; \theta)$, where $y = M(\cdot)$ denotes a diffusion estimation function.

1. To build an efficient objective function:
→ Characterize the probability of the seed node set $p(\boldsymbol{x})$ over $\boldsymbol{x}$ given the graph G.

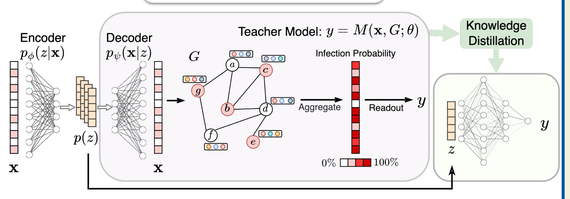
2. To model the underlying diffusion pattern:
→ Two different diffusion models $M(\cdot)$ and $M_S(\cdot)$ to characterize the diversified diffusion dynamics with efficiency and efficacy guarantee.

GNN-based Influence Estimation Model (Efficacy)

MLP-based Student Estimation Model (Efficiency)

$$\mathcal{L}_{\text{train}} = \min_{\theta,\lambda,\psi,\phi} \mathbb{E}\Big| - \log\big[p_\theta(y|\mathbf{x}, G)\big] - \log\big[p_\lambda(y_s|z)\big] - \log\big[p_\psi(\mathbf{x}|z) \cdot (p_\phi(z|\mathbf{x})\big]\Big|, \text{s.t. } \theta \ge 0.$$

Seed Node Set Probability Quantification



5. Inference Phase

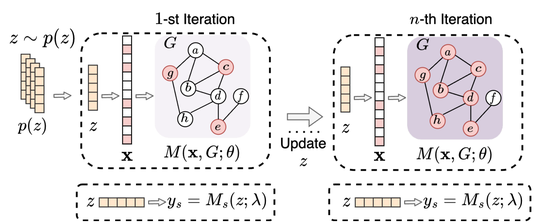
Optimization for Seed Set Inference. We propose to search the optimal seed node set $\tilde{\boldsymbol{x}}$ in the lower-dimensional latent space $p(z)$ by iteratively optimizing the latent variable z sampled from $p(z)$.

3. To meet different node-centrality-constraints:
→ Propose a novel objective function that can be coupled with multiple constraints for seed node set inference, which can adapt to different IM application schemes.

$$\mathcal{L}_{\text{pred}} = \max_z \mathbb{E}\big[p_\theta(y|\mathbf{x}, G) \cdot p_\psi(\mathbf{x}|z)\big], \quad \text{s.t.} \sum_{i=0}^{|V|} \mathcal{F}(v_i, G) \cdot x_i \le k,$$

$\sum_{i=0}^{|V|} \mathcal{F}(v_i, G) \cdot x_i$ is a is a generalized budget constraint applied on individual nodes, and k is the actual budget (e.g., the # of nodes or the cost of selecting nodes).

$$\mathcal{L}_{\text{pred}} = \min_z \Big[- \log\big[\prod_{i=0}^{|V|} f_\psi(z_i)^{x_i}(1 - f_\psi(z_i)^{1-x_i}\big] + \big\|\tilde{y} - y\big\|_2^2\Big] \text{ s.t. } \sum_{i=0}^{|V|} \mathcal{F}(v_i, G) \cdot x_i \le k,$$



6. Experiments

Data and Baselines. We conducted experiments on six real-world networks and one synthetic network. We compare our method with both traditional and learning-based approaches.

	Digg	Weibo	Power Grid	Network Science	Cora-ML	Jazz	Synthetic
Nodes	279,613	2,251,166	4,941	1,565	2,810	198	50,000
Edges	1,170,689	225,877,808	6,594	13,532	7,981	2,742	250,000

The primary purpose is to evaluate the portion of influence spread y over the entire node set.

Methods	Cora-ML 1%	5%	10%	20%	Network Science 1%	5%	10%	20%	Power Grid 1%	5%	10%	20%	Jazz 1%	5%	10%	20%	Synthetic 1%	5%	10%	20%	Digg 1%	5%	10%	20%	Weibo 1%	5%	10%	20%
IMM	8.1	26.2	37.3	50.2	5.2	16.8	27.0	45.7	4.3	17.4	31.5	51.1	2.6	20.1	31.4	42.8	9.2	26.2	36.3	51.6	7.4	18.4	32.8	49.6	9.5	23.8	36.4	50.5
OPIM	13.4	26.9	37.4	50.9	6.6	19.4	28.9	48.6	5.7	17.7	29.7	50.1	2.4	20.1	34.4	46.8	9.6	25.3	36.6	51.7	7.6	18.5	32.9	48.9	9.7	23.7	36.6	50.3
SubSIM	10.1	25.7	36.8	51.1	4.8	15.4	27.9	44.8	4.6	19.2	31.7	50.2	3.6	18.8	37.6	44.7	9.5	26.7	36.5	51.5	7.5	18.9	33.3	49.4	9.3	23.1	36.5	50.6
OIM	8.9	27.6	38.0	51.3	4.2	16.7	26.5	48.2	5.7	17.5	31.9	50.8	2.0	18.5	36.3	42.2	9.6	26.2	36.7	51.3	7.8	18.2	33.1	49.6	-	-	-	-
IMINFECTOR	9.6	26.8	37.7	50.6	5.4	17.9	27.8	47.6	5.4	18.2	31.6	50.9	3.6	19.7	37.5	45.9	9.1	26.2	36.1	51.5	7.9	18.6	33.5	49.8	9.4	23.5	36.9	50.3
PIANO	9.8	25.2	37.4	51.1	4.7	16.3	27.1	47.2	5.3	18.1	31.7	50.2	2.2	19.2	36.6	43.2	9.1	26.4	36.2	51.6	-	-	-	-	-	-	-	-
ToupleGDD	10.6	27.5	38.5	51.5	6.3	17.8	28.3	50.5	5.4	19.3	31.6	51.3	3.3	20.4	37.2	45.7	9.5	26.8	37.1	51.4	-	-	-	-	-	-	-	-
DeepIM$_s$	13.6	27.7	38.5	51.8	6.9	19.1	29.3	50.5	5.9	20.2	31.7	51.5	3.8	21.4	38.9	47.1	10.2	26.8	37.5	51.8	7.9	18.8	33.7	50.3	10.1	24.7	36.8	50.8
DeepIM	14.1	28.1	39.6	52.4	7.8	20.9	31.5	51.2	6.3	21.0	32.5	52.4	4.9	23.3	41.5	49.9	11.6	27.4	38.7	52.1	8.4	19.3	34.2	51.3	11.2	26.5	37.9	51.8

Table 2. Performance comparison under IC diffusion pattern. – indicates out-of-memory error. (Best is highlighted with bold.)

Methods	Cora-ML 1%	5%	10%	20%	Network Science 1%	5%	10%	20%	Power Grid 1%	5%	10%	20%	Jazz 1%	5%	10%	20%	Synthetic 1%	5%	10%	20%	Digg 1%	5%	10%	20%	Weibo 1%	5%	10%	20%
IMM	1.7	34.8	52.2	66.4	2.5	11.9	18.1	33.6	4.6	19.9	31.7	56.9	1.4	5.7	13.4	24.5	1.1	5.2	13.1	66.9	2.4	10.8	37.4	55.6	1.6	6.7	19.3	45.2
OPIM	2.3	36.9	51.2	71.5	1.6	12.0	18.8	34.1	4.4	21.6	29.4	55.5	1.4	6.9	12.6	20.9	1.3	5.2	12.6	62.1	2.1	11.3	38.2	57.1	1.8	6.1	18.7	46.6
SubSIM	1.7	33.6	54.7	70.1	1.8	10.4	19.2	34.1	4.5	21.1	31.2	57.4	1.4	5.9	11.4	21.2	1.4	5.5	13.1	69.6	2.4	11.3	37.9	56.9	1.7	6.7	19.2	46.8
IMINFECTOR	2.1	33.9	51.3	70.6	2.1	11.8	18.7	34.5	4.2	21.3	31.6	56.2	1.4	6.2	13.5	22.8	1.3	5.2	12.9	67.4	2.2	11.1	38.9	58.7	1.8	6.4	18.6	47.5
PIANO	2.1	33.5	53.3	69.8	2.1	11.3	19.1	33.9	4.3	21.3	31.4	57.1	1.1	6.2	12.1	22.4	1.2	5.2	12.9	67.4	-	-	-	-	-	-	-	-
ToupleGDD	2.3	36.2	54.5	70.9	2.8	12.4	19.8	34.6	4.8	21.9	32.6	58.1	1.4	6.5	12.9	23.6	1.3	5.5	13.4	70.2	-	-	-	-	-	-	-	-
DeepIM$_s$	10.7	65.6	75.1	85.2	3.5	14.6	23.8	37.8	5.1	22.9	40.3	65.1	1.4	6.5	14.2	85.3	1.5	6.0	14.2	90.3	3.1	13.3	39.2	67.9	2.5	7.1	32.6	68.4
DeepIM	13.4	69.2	83.5	94.1	4.1	16.6	26.7	41.5	6.3	24.4	46.8	71.7	1.9	6.5	16.4	99.1	1.5	6.5	15.5	99.9	3.5	15.9	41.3	76.2	3.1	7.6	39.3	72.4

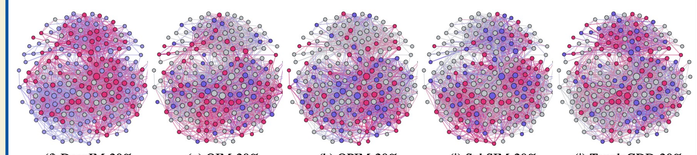
Table 3. Performance comparison under LT diffusion pattern. – indicates out-of-memory error. (Best is highlighted with bold.)

Scalability Analysis. We record the runtime of the seed set inference with regard to the increase in node size against other learning-based IM solutions. Compared to learning-based methods, both variants of DeepIM surpass others with an evident margin (on average 20% faster inference time and 1/3 of the second-best method's training time).

	10,000	20,000	30,000	50,000	50,000 (Training)
IMINFECTOR	3.478s	7.842s	12.376s	16.492s	4753.67s
PIANO	5.948s	10.532s	16.575s	28.437s	14732.63s
ToupleGDD	10.476s	19.583	32.792s	58.985s	–
DeepIM$_s$	0.312s	0.616s	0.847s	1.275s	503.12s
DeepIM	1.402s	2.798s	5.124s	12.882s	1244.56s

Table 4. The average inference runtime (in seconds) with regard to the increase of node size (10,000, 20,000, 30,000, and 50,000).

Visualization. The visualization of influence spread in Jazz dataset with 20% node being seeds: Node size is based on the node degree, and node color determines the infection status: blue means the node is in seed set, red means the node is infected, and grey means the node is not infected.



(f) DeepIM-20% (g) OIM-20% (h) OPIM-20% (i) SubSIM-20% (j) ToupleGDD-20%