Introduction to the Problem

Typical Influence Maximization:
- Relies on diffusion simulation models.
- Influence probabilities are set at random.
- Faces scalability issues.
- Estimated influence spread can wildly diverge from realistic spread [1].

Influence Learning Models:
- Assume influence independence, overlooking the effects of network assortativity [2].
- Ignore higher order influence between nodes.
- Are based on propagation network which has high computational cost [3].

Contribution:
- Analysis of influencer activity for efficient node-context creation.
- Multi-task neural network architecture to compute influencer embeddings.
- An algorithm to perform influence maximization using the learnt representations.

II. Learning INFluencer vECTORS (INFECTOR)

Embed at the same hidden layer a node’s:
- likelihood to influence another node.
- aptitude to create lengthy cascades.
S is updated alternately by both inputs.
- 0 and 7 form the diffusion probabilities between two nodes.
- |S| captures the nodes’ cascade size.

Table: The layers of INFECTOR

1. Predict diffusion probabilities:
   \[ D = \left[ \frac{\nu(S)}{\nu(S)} \right] \]

2. Reduce the number of candidate seeds by keeping the top P% based on their expected influence spread.
   \[ A_U = \frac{N}{\sum_{i=1}^{N} |S_i|} \]

3. IM by weighted bipartite matching.

IV. Results

Methods:
- Top nodes based on k-cores decomposition.
- Top nodes based on the average size of their train cascades.
- IMM: SOTA classic influence maximization algorithm [3].
- IMINFECTOR with 5 epochs, 0.1 learning rate, P=40 for Digg and 10 for the rest [4].

Evaluation:
- Precision: How many of the predicted seeds initiate test cascades.
- DNI: Total number of nodes influenced by the predicted seeds in the test cascades.

III. Influence Maximization with INFECTOR

1. Predict diffusion probabilities:
   \[ D = \left[ \frac{\nu(S)}{\nu(S)} \right] \]

2. Reduce the number of candidate seeds by keeping the top P% based on their expected influence spread.
   \[ A_U = \frac{N}{\sum_{i=1}^{N} |S_i|} \]

3. IM by weighted bipartite matching.

IV. Conclusion & Future Work

Conclusion:
- Diffusion cascades should be taken into account for realistic influence maximization.
- Representation learning can be used to bridge influence learning and maximization.

Future Work:
- Use complementary representations derived from Graph Neural Networks.
- Combine learnt representations from diffusion logs with online influence maximization.
- Compare with diffusion-based techniques and more metrics from network science.

Code and Instructions: https://github.com/GiorgosPanagopoulos/IMINFECTOR

I. Influence Analysis on Sina Weibo

Influencers create or copy more?
- Rank initiators in the test set based on success metrics.
- Successful initiators are more prone to start than participate in train cascades → Derive only the context of the cascade initiator.

Does copying time play a role?
- Average copying time in train cascades for every level of DNI in the test set.
- Influencers tend to initiate fast cascades. Sample context based on the inverse of copying time.

*11 months of rewet cascades as “train”, 1 month as “test”.

IV. References