Introduction

Typical Influence Maximization
- Relies on diffusion simulation models.
- Influence probabilities are set at random.
- Faces scalability issues.

Influence Learning:
- Ignores higher order influence between nodes.
- Is based on propagation network which has high computational cost.

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 with the diffusion probabilities.

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.
- S and T form the influence likelihood between nodes.
- |S| captures the nodes’ cascade size.

Table: The layers of INFECTOR

III. Influence Maximization with INFECTOR

Predict diffusion probabilities.

Capture higher order correlations that other IM techniques fail to.

Optimize ρ(Δ) in greedy manner and remove the node added in each iteration1. ρ(Δ) is submodular → retains theoretical guarantees and can use CELF.

*because there are no higher order paths.

Data
- Static Directed Network + Diffusion cascades.
- 80-20% train and test split on cascades based on time precedence.

IV. Results

Methods
- Top nodes based on k-cores decomposition (Malliaros et al. 2018).
- Top nodes based on the average size of their train cascades (Bakshy et al. 2011).
- IMINFECTOR with 5 epochs, 0.1 learning rate, P=40 for Digg and 10 for the rest.

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.

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.
- The cost of the propagation network can be alleviated by focusing on the initiators and taking time into account.

Future Work
- Use complementary representations derived from Graph Neural Networks.
- Combine learnt representations with online IM.
- Compare with purely diffusion-based IM techniques.

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

References
- CentraleSupélac and Inria Saclay
- Michalis Vazirgiannis

george.panagopoulos@lix.polytechnique.fr