STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LXIX**, Number 1, 2024
DOI: 10.24193/subbi.2024.1.04

COMPETITIVE INFLUENCE MAXIMIZATION IN TRUST-BASED SOCIAL NETWORKS WITH DEEP Q-LEARNING

ANIKÓ KOPACZ

Abstract. Social network analysis is a rapidly evolving research area having several real-life application areas, e.g. digital marketing, epidemiology, spread of misinformation. Influence maximization aims to select a subset of nodes in such manner that the information propagated over the network is maximized. Competitive influence maximization, which describes the phenomena of multiple actors competing for resources within the same infrastructure, can be solved with a greedy approach selecting the seed nodes utilizing the influence strength between nodes. Recently, deep reinforcement learning methods were applied for estimating the influence strength. We train a controller with reinforcement learning for selecting a node list of given length as the initial seed set for the information spread. Our experiments show that deep Q-learning methods are suitable to analyze the competitive influence maximization on trust and distrust based social networks.

1. Introduction

Monitoring the information spread in social networks is beneficial for public opinion analysis, evaluating and marketing strategies. The influence maximization [3] problem aims to maximize the information coverage, while minimizing the cost associated with the degree of information spread. Competitive influence maximization [1] refers to the optimization problem, when multiple entities operate on the same social network and each of them attempts to maximize the information spread in parallel. In viral marketing, multiple companies often target the same audience with similar products. The goal of each individual company is to maximize their own revenue and persuade the

Received by the editors: 1 March 2023.

2010 *Mathematics Subject Classification.* 68T05.

1998 *CR Categories and Descriptors.* G.2.2 [**DISCRETE MATHEMATICS**]: Graph Theory – *Network problems*; G.3. [**PROBABILITY AND STATISTICS**]: Markov Processes; I.2.6 [**ARTIFICIAL INTELLIGENCE**]: Learning – *Connectionism and neural nets*.

Key words and phrases. influence maximization, reinforcement learning, Q-learning.

most possible individuals within a social network to choose their product over a competitor's.

Reinforcement learning is a computational approach to learn a specific task based on an agent-environment interaction. The agent's learning process is guided by a reward received: successful steps towards the completion of a predefined task are associated with positive feedback. Reinforcement learning has been applied to several real-world inspired optimization problems, such as robotics, epidemiology, scheduling and routing problems. Moreover, the reinforcement learning setting can be extended to the influence maximization problem [2, 6, 11].

The present work argues that deep reinforcement learning is suitable for constructing initial seed sets for competitive influence maximization on social networks displaying trust-distrust relationships. Two distinct mechanisms are analyzed to construct initial seed sets for two competing actors: joint- and iterative seed selection. The effectiveness of possible seed sets is compared based on the number of activated nodes after simulating polarity related independent cascade on trust-distrust networks.

This article is organized as follows. In Section 2, the polarity related competitive influence maximization problem is described. Section 3 presents the reinforcement learning setting and a deep reinforcement learning method, namely Deep Q-network. The conducted experiments and results are shown in Section 5. Finally, our conclusions and opportunities for improvement are summarized in Section 6.

2. Influence maximization

Influence maximization [3] targets the optimization of information spread in social networks starting from a set of source nodes. Let K denote the maximum number of nodes in the seed set. In [3] the authors also proposed a greedy baseline algorithm under the two main existing diffusion models, namely Independent Cascade and Linear Thresholds. The influence maximization problem is NP-hard [3], therefore, in addition to approaches proposed specifically optimize influence maximization, various soft computing methods can be applied to alleviate the computational requirements e.g., reinforcement learning.

Carnes et al. extended the independent cascade model to the competitive scenario, where actors with opposing interests are present, introducing two influence spread mechanisms: the distance based and the wave propagation model. Both models are suitable for constructing a seed set greedily to address the problem of competitive influence maximization (CIM).

Conventional influence maximization methods are biased simulating influence spread groups with different attributes [7]. Thus, the balanced influence maximization was proposed to examine influence spread in attributed social networks [7]. The baseline algorithm for the balanced influence maximization problem is Attribute-based Reverse Influence Sampling algorithm, that achieves the efficiency of conventional Influence Maximization methods and manages to conserve the initial attribute distribution of the sampled social network [7].

In [2] the authors proposed a reinforcement learning framework regarding influence maximization problem in random graphs. For the selection of initial source points for the information cascade a Markov Decision Process is proposed. Markov Decision Processes may be solved by applying single agent reinforcement learning [10]. The autonomous agent selects the source nodes to broadcast an initial message, policy improvement is applied to approximate the action-value function. The reward received for selecting certain source nodes shows the degree of information dissemination in the network after simulating information cascade with a finite time-horizon.

Recently, hierarchical generative embedding was implemented with the goal to map the network nodes to a lower-dimensional embedding space [11]. The learned node representation is utilized for estimating the influence strength between two nodes and the most influential nodes are selected greedily in regard to the learned representation. The method is evaluated on various social networks based on real-world data, such as citation networks [8].

In [6] deep reinforcement learning was studied to construct an estimator to determine the expected influence of nodes. Network embedding is applied to construct a vector representation of nodes, the obtained vectors are utilized as an input for a deep Q-network [9] that approximates the expected influence. The seed set optimizing the influence spread is constructed by selecting nodes with the objective of maximizing the expected influence. The node selection is performed in one iteration, all embeddings are computed and the top k candidates are appointed as the seed set.

2.1. **Polarity related influence maximization.** The influence maximization problem formalized in [3] features social networks having a single type of relation between individuals. Polarity related influence maximization [5] operates taking into account two opposing type of relationships. Methods addressing influence maximization can be extended to solve the polarity related influence maximization by applying polarity related independent cascade [5].

In the competitive influence maximization setting, where two distinct actors attempt to influence vertices in of the same network, multiple node activation statuses occur. A vertex is considered inactive if none of the actors managed to influence them yet. In the case of activated nodes, two additional states are distinguished representing the polarity of activated nodes. Furthermore,

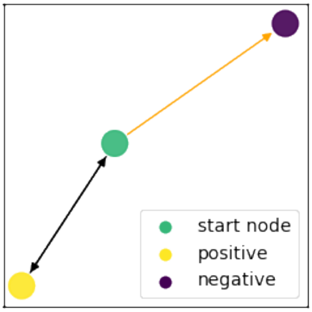


FIGURE 1. Example for polarized activation of nodes. The green node is marked as positive activating its neighbors following outgoing edges, the neighbors that trust the starting node (black edges) turn positive, while nodes that distrusted the starting node (orange edges) become negative

the polarity of nodes also marks the actor which influenced the current node. The vertices activated by the reinforcement learning agent are called positive vertices, whereas the vertices activated by the adversary are called negative ones.

We construct a seed set by selecting a k number of nodes to be activated in the beginning of the simulation. Then, a polarized node activation is simulated (see Figure 1) in accordance with [5]: if a given vertex would activate one of its' neighbors, the neighbor will choose for itself: (i) the same sign (positive or negative) if the edge between the source node and the activated neighbor is positive, or (ii) opposite sign if the edge between the two nodes is negative. If an n node is activated in a simulation step t, n tries to activate its' inactive neighbors in the next time step $t + 1$. In following time steps starting from $t+2$, n no longer broadcast information toward its' inactive neighbors and no longer activates individuals in the social network. In this model, each node will be activated only once and will preserve its positive or negative status over the simulation.

3. Reinforcement learning

Reinforcement learning (RL) trains an action policy to optimize the behavior of an agents in an observable or partially observable environment [10]. Previous experiences of agent-environment interaction characterized by a reward signal are utilized to optimize solving a predefined task. The optimization problem of training an RL agent can be formalized as a Markov Decision Process (MDP) [10]. The $\{\mathcal{S}, \mathcal{A}, p, \mathcal{R}, \gamma\}$ tuple is an MDP if the Markov property holds true, which states that the immediate reward r and the agent's next state s_{i+1} is defined by the the previous state s_t and the a_t action taken:

$$p(s', r \mid s, a) = Pr\{s_{t+1} = s', R_{t+1} = r \mid s_t = s, a_t = a\}$$

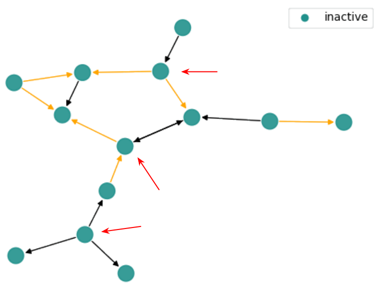
Optimal action selection policies may utilize action-value estimators to assess the potential benefits of selecting an action a in a given state S. We denote with $Q(s, a)$ the pay-off for the agent for taking action a in the s state.

Deep Q-Networks (DQN) were introduced in [9] utilizing neural networks to approximate the associated gain for possible state-action pairs. The action-value function for a state-action pair – denoted by $Q(s, a)$ – measures the goodness of choosing the a action over any other available action in state s. The optimal policy receiving a state as input is constructed as a greedy action selector regarding the estimated Q-values. The experience replay mechanism is implemented to generate training batches for the Q-network. The agent's interaction with the environment is saved into a buffer. In each training step, Q-values are calculated for $s_t, a_t, s_{t+1}, r_{t+1}$ state transitions drawn from the experience buffer. The weights of the neural network are updated with the objective to minimize the temporal difference [10] calculated for the current batch of state transitions. The temporal difference error is computed using a target network, a periodic backup of the trained Q-network. The target network is robust in regard to abrupt changes of the Q-values, hence, a more stable training approach is obtained.

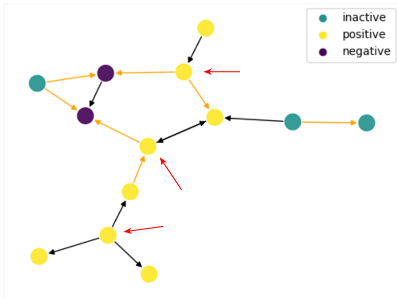
4. Proposed methods

Several approaches exist for interpreting the influence maximization as a reinforcement learning problem. With the scope of formalizing the influence maximization as a Markov Decision Process, which may be solved by applying deep Q-learning methods, we describe two models.

4.0.1. *Joint seed selection.* The activation states for the vertices of the social network(s) are encoded with integers, establishing the state representation of the reinforcement learning problem. Inactivated nodes are labelled as 0, activated and positive nodes get 1, activated and negative nodes get -1 labels, respectively. The seed set for the first agent producing positively activated nodes is selected as one action of the RL agent. The possible seed sets are obtained based on the social network infrastructure before the deep Q-learning takes place. The episode consists of 1 iteration: both the agent and its' adversary select their seed set, given the independent cascade model, the degree



(A) Selecting 3 seed nodes (red arrows)



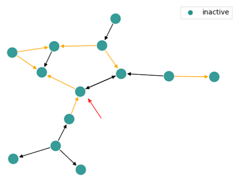
(B) Activation status of the nodes

FIGURE 2. Select 3 seed nodes with joint seed selection. Red arrows point to seed nodes that activate their neighbors based on outgoing edges: positive edges preserve the sign of activation, while activation is switched when following negative edges

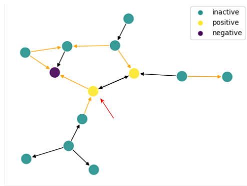
of information dissemination is obtained, and the reward is the number of activated and positive nodes.

Figure 2 illustrates the joint seed selection in a directed graph that has positive and negative edges. The budget allocated for the seed set is 3 and the seed nodes will be marked as positive. The RL agent receives as input the activation statuses of the network nodes and selects a 3 length list of seed nodes generated from all nodes present in the network (Fig. 2a). Neighbors are activated simultaneously according to the polarized independent cascade model described in Section 2.1. Following the outgoing edges of the seed nodes, inactive nodes are activated positive edges leading to positive neighbors, while negative edges result in negative neighbors (Fig. 2b). Then, the activated neighbors may activate the remaining inactive neighbors. The neighbors of the seed nodes do not have outgoing edges that point to inactive nodes (see Fig. 2b), thus, the simulation of polarized independent cascade is finished. The RL agent receives the total number of activated and positive nodes, which is 8 in this case.

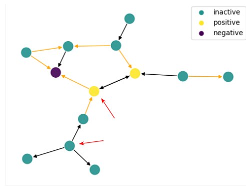
4.0.2. *Iterative seed selection.* The state representation and the encoding of the nodes is identical as described in Section 4.0.1. However, the initial seed set is assembled in an iterative manner. The reinforcement learning episode consists of a maximum k number of iterations, in each iteration the actors select an inactivated node to be added to their respective seed sets. The immediate reward received by the agents in each iteration is going to be the change in the number of activated nodes.



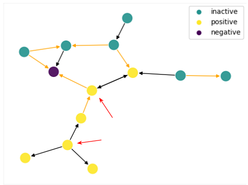
(A) Select the first seed node



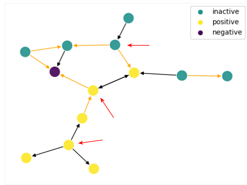
(B) Activate the neighbors of the first seed node



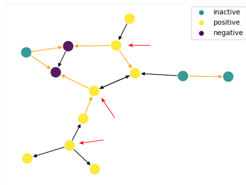
(C) Add the second seed node



(D) Inactive neighbors are activated



(E) Select the last seed node



(F) Activate the neighbors of the last seed nodes

FIGURE 3. Selecting 3 nodes for a seed set in an iterative manner in a directed graph with positive and negative edges. Red arrows point to seed nodes that activate their neighbors based on outgoing edges: positive edges preserve the sign of activation, while activation is switched when following negative edges

Figure 3 illustrates the iterative seed selection process in a directed graph that has positive and negative labels associated with each edge with a budget of $k = 3$. The seed set is initialized as empty at the beginning. Each node is selected independently, after activating the new seed node and marking the seed node as positive, a polarized independent cascade step is performed. The first seed node shown on Figure 3a activates one of its neighbors as positive, while the other neighbor becomes negative (Figure 3b). After updating the activation statuses, the RL agent receives as a reward the number of positive nodes including the seed node; the reward for the first seed node is 2. The second seed node is chosen from the inactive nodes (Figure 3c) and attempts to activate its neighbors. Previously activated nodes remain with the original activation status indifferent to their neighbors becoming activated, as shown on Figure 3d, the first seed node remains positive although it is connected to another positive node. The reward associated to selecting the second seed node for the given activation state is 4. Finally, the third seed node is selected (Fig. 3e), the 2 neighbors are newly activated (Fig. 3f) yielding a reward of 2.

5. Experiments

5.1. **Data.** We conducted our experiments on trust and distrust based social networks constructed from user relations on the `epinions.com` consumer review site [4]. Users from the website are considered nodes and nodes are connected (by edges) if the corresponding users trust or distrust one another in the context of the review site. Weights assigned to edges detail the kinship between the respective users: 1 encodes trust between users, while -1 refers to distrust. For modelling trust relations within users, directed graphs are preferred because trust (or distrust) between users is not necessarily symmetrical, i.e. *user A* trusts *user B*, however *user B* does not trust *user A*.

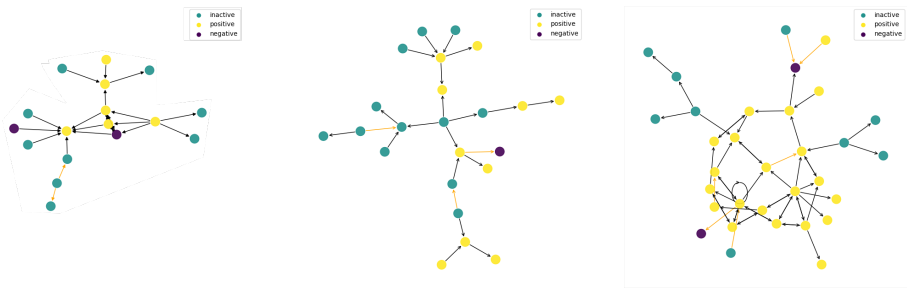
The social network constructed from `epinions.com` contains over 100000 nodes and 800000 edges.[4]. Analysis of the constructed social network showed that the probability of edges being positive is higher for nodes with a larger number of neighbors, while negative edges tend to act as bridges between positive clusters [4]. The original network was sub-sampled to experiment with deep Q-learning methods alleviating computational demands needed for processing large networks and assessing scalability of reinforcement learning approaches.

Weakly connected graphs are generated by selecting *nodes* from the original network using the following method. First, an initial node is chosen from the original network generated by a uniform distribution. A neighbor-pool that will contain the nodes connected to the sample graph nodes but not in the sample graph is initialized as an empty set. Whenever a new node is added to the sub-graph, the neighbor-pool is extended with all the nodes connected to the selected node not already in the pool, both incoming and outgoing edges are considered. In the following steps, nodes are selected from the neighbor-pool by a uniform random distribution and added to the sub-sample graph. The sampling terminates when the neighbor-pool does not have any more candidates, or the size of the sub-sample network reaches a certain threshold. Small-scale sub-networks were obtained by selecting an upper limit of 17, 23 and 32 number of nodes within a sample graph. Medium scale graphs were generated with 93 and 340 nodes, respectively.

5.2. **Experimental setup.** Two distinct reinforcement learning models are described to address the competitive influence maximization problem. Given the competitive nature of the optimization problem, two actors are distinguished to operate on the social network. We study optimal (policy) configurations for the actors separately, the currently analyzed actor is going to be referred to as the *agent* and RL methods are applied for generating possible initial seed sets for the selected actor. The two competing actors determine

Graph	Number of nodes	Number of edges	$K_{positive}$	Number of positive nodes	Number of negative nodes
G-17	17	24	3	7	2
G-23	23	22	4	9	1
G-32	32	55	4	13	2

Table 1. Selecting the starting nodes with joint seed selection by a controller trained with DQN on small trust-distrust graphs



(a) G-17 (b) G-23 (c) G-32

Figure 4. Activation status of nodes in small trust-distrust networks determined by the DQN approach

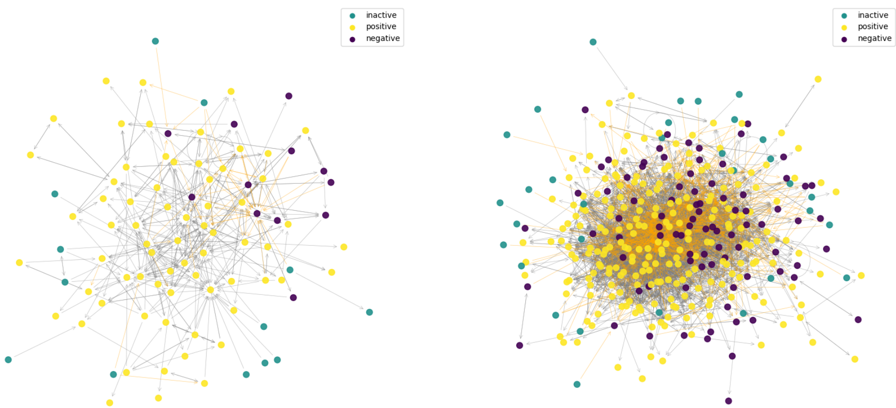
their initial seed sets simultaneously, then the influence spread is simulated under the competitive independent cascade [1] diffusion model. The actors take turns to activate nodes within the network: the vertices activated by the first actor are marked as positive, while vertices activated by the second actor are labeled negative.

For the sake of simplicity, the social network is assumed to be known during our experiments and only the policy generating the initial seed set is optimized. For different graph instances, new policies are trained using deep Q-learning. Training with a fixed social network architecture aims to reduce the magnitude of the optimization problem.

5.3. **Results.** In this paper, we evaluated deep Q-networks (DQN, [9]) approach for selecting initial seed sets for the competitive influence maximization problem [1] in signed trust based social networks. Two distinct reinforcement

Graph	Number of nodes	Number of edges	$K_{positive}$	Number of positive nodes	Number of negative nodes
G-93	93	444	1	68	12
G-340	340	4958	1	222	86

TABLE 2. Selecting the starting nodes with iterative seed selection by a controller trained with DQN on trust-distrust graphs



(A) G-93 (B) G-340

FIGURE 5. Activation status of nodes in medium size trust-distrust networks determined by the DQN approach

learning models were applied to formalize the influence maximization problem and construct the solution space.

Joint seed selection (see Section 4.0.1) is suitable to operate on trust-distrust based social networks with limited number of nodes. In Table 1, 3 distinct social networks are presented alongside with the seed sets for the influence maximization problem determined by the DQN method. The inactive, positively, and negatively activated nodes are determined by following polarity related independent cascade [5]. The structure of the evaluated small networks and the activation of nodes given the seed sets determined by the controller trained with DQN are shown on Figure 4.

When applying joint seed selection, the dimensions of the action space increase exponentially with the number of vertices in the social network. The memory consumption and required execution time can be reduced by determining the elements of the seed node set in an iterative manner (see Section 4.0.2). Table 2 summarizes the quantitative characteristics of training with DQN for the iterative selection of seed nodes in trust-distrust social networks in the context of competitive influence maximization. In case of the larger sample networks, the distribution of inactive, positively and negatively activated nodes for the evaluated social networks is shown on Figure 5.

5.4. **Discussion.** Experimental results show that deep reinforcement learning is suitable for proposing seed sets for the competitive influence maximization problem on polarized networks. In this paper, two actors operate on social networks of various sizes that describe trust-based relationships. At the evaluation step, the same policy is used to select seed sets for both actors. The nodes activated by the first actor are reported as positive nodes, whereas the nodes activated by the second actor are reported as negative nodes. The actors select nodes for the seed set simultaneously; however, because nodes cannot change their activation in the conducted experiments, the final activation status of a node selected by both actors is positive. In the evaluated trust-distrust graphs, the first actor has an advantage over the second actor, and more nodes are positive than negative after both actors selected a seed set.

The joint seed selection described in Section 4.0.1 is feasible to determine seed sets for the competitive influence maximization problem. The joint seed selection method does not scale well due to the fact that the action space increases rapidly with the budget for the seed set. Experiments show that iterative seed selection (Section 4.0.2) can be utilized with the DQN approach to operate on medium-scale networks. In case of the G-93 and G-340 sub-networks, a $K = 1$ length seed set activates as positives a large proportion of the network nodes (see Table 2). The original social network is observed to contain positive clusters [4]. The occurrence of a larger number of positive connections of nodes facilitates the information spread in the medium-scale sub-networks.

6. Conclusions and future work

Reinforcement learning proceeds to extract meaningful information from past agent-environment interactions. Reinforcement learning is suitable for addressing NP-hard optimization problems, such as influence maximization. Deep Q-Networks, a well-known reinforcement learning method, were trained to optimize the competitive influence maximization problem on polarized networks. The solution space of the influence maximization problem increases rapidly with the number of nodes and the size of the seed set. To alleviate the impact on memory consumption and the time necessary for training of the models several techniques can be applied, e.g. introducing a filtering step to exclude infeasible actions, constructing the seed node set by selecting nodes one by one. Future work includes analyzing the application areas of influence maximization and applying network embedding methods to project the node representation into a latent space.

References

[1] Carnes, T., Nagarajan, C., Wild, S. M., and van Zuylen, A. (2007). Maximizing influence in a competitive social network: A follower's perspective. In *Proceedings of the Ninth International Conference on Electronic Commerce*, ICEC '07, page 351–360, New York, NY, USA. Association for Computing Machinery.

[2] Chen, M., Zheng, Q., Boginski, V., and Pasiliao, E. (2021). Influence maximization in social media networks concerning dynamic user behaviors via reinforcement learning. *Computational Social Networks*, 8.

[3] Kempe, D., Kleinberg, J., and Tardos, É. (2003). Maximizing the spread of influence through a social network. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 137-146.

[4] Leskovec, J., Huttenlocher, D. P., and Kleinberg, J. M. (2010). Signed networks in social media. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.*

[5] Li, D., Xu, Z., Chakraborty, N., Gupta, A., Sycara, K. P., and Li, S. (2014). Polarity related influence maximization in signed social networks. *PLoS ONE*, 9.

[6] Li, H., Xu, M., Bhowmick, S. S., Rayhan, J. S., Sun, C., and Cui, J. (2022). Piano: Influence maximization meets deep reinforcement learning. *IEEE Transactions on Computational Social Systems*, pages 1–13.

[7] Lin, M., Li, W., and Lu, S. (2020). *Balanced Influence Maximization in Attributed Social Network Based on Sampling*, page 375–383. Association for Computing Machinery, New York, NY, USA.

[8] McCallum, A. K., Nigam, K., Rennie, J., and Seymore, K. (2000). Automating the construction of internet portals with machine learning. *Information Retrieval*, 3(2):127–163.

[9] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602.*

[10] Sutton, R. S. and Barto, A. G. (2018). *Reinforcement Learning: An Introduction.* The MIT Press, second edition.

[11] Xie, L., Huang, H., and Du, Q. (2022). A hierarchical generative embedding model for influence maximization in attributed social networks. *Applied Sciences*, 12(3):1321.

Faculty of Mathematics and Computer Science, Babeş-Bolyai University, 1 Mihail Kogalniceanu str, Cluj-Napoca, Romania

Email address: aniko.kopacz@ubbcluj.ro