Cluster-Based Social Reinforcement Learning

Extended Abstract

Mahak Goindani
Department of Computer Science
Purdue University
West Lafayette, IN
mgoindan@purdue.edu

Jennifer Neville
Departments of Computer Science and Statistics
Purdue University
West Lafayette, IN
neville@purdue.edu

ABSTRACT

Social Reinforcement Learning considers multi-agent systems with large number of agents and relatively few interactions between them, which is challenging due to high-dimensional search space, inter-agent dependencies that increase computational complexity. Moreover sparse agent interactions produce insufficient data to capture higher-order relations (interactions) for learning accurate policies. To overcome these challenges, we present a dynamic cluster-based Social RL approach that utilizes the properties of the social network structure, agent interactions, and correlations to obtain a compact model to represent network dynamics.

ACM Reference Format:
Mahak Goindani and Jennifer Neville. 2020. Cluster-Based Social Reinforcement Learning. In *Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), Auckland, New Zealand, May 9–13, 2020,* IFAAMAS, 3 pages.

1 INTRODUCTION

Real-world social networks consist of thousands of users who interact with each other and are related in various ways [4–8, 17]. However, each user typically only interacts with a small number of other users in the network. This means the network interactions are overall sparse. Traditional MARL approaches do not scale for large numbers of agents, and state-of-the-art Social RL approaches [5, 8, 17] do not overcome the problem of sparse user interactions, resulting in noisy policy estimates [9, 10]. To address these challenges, we propose a *Dynamic Cluster-based Policy Learning* (DCPL) approach for social RL. Our work makes the following major contributions: (1) We utilize agent correlations to cluster agents and reduce model dimensionality (i.e., number of parameters) from $\sim N^2$ to $\sim \mathcal{G}^2$, $\mathcal{G} \ll N$, and thus computational cost, while still facilitating joint learning to capture agent dependencies. (2) We design difference reward based clustering features to help in effective credit assignment for resource allocation. We use these features (via clusters) to learn agents' effectiveness early on, to better explore the action space without increasing the state space, and thus policies converge faster. (3) We revise the clusters over time, which facilitates learning of policies that are responsive to change in agent behavior. We combine this with a method to easily derive personalized agent-level actions from cluster-level policies by exploiting the similarity and variability in agents' behavior. To the best of our knowledge, our DCPL approach is the first to consider policy learning while dynamically clustering users in MARL for social networks.

Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), B. An, N. Yorke-Smith, A. El Fallah Seghrouchni, G. Sukthankar (eds.), May 9–13, 2020, Auckland, New Zealand. © 2020 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

2 CLUSTER-BASED POLICY LEARNING

We consider an application of fake news mitigation [1, 14–16] to demonstrate the utility of our proposed approach (see [9] for detailed problem description). The objective is to combat fake news by increasing the spread of true news [5, 8]. We consider a social network with N users (agents) who interact via different network activities such as tweets, retweets, likes. The (re)tweets are labeled fake (F) or true (T). Our data contains a temporal stream of events with the time horizon divided into K stages, each of time-interval ΔT, where stage $k \in [1, K]$ corresponds to the time-interval $[\tau_k, \tau_{k+1})$. Fig. 1 illustrates the different components of our system.

The objective is to learn a *policy* π that maps the network *state* (over N users) to *actions* (over N users). In a real-world social network, users interact with each other, which leads to *dependencies* between their actions [9]. Thus, to learn the policy for a single user, we need to consider the actions of all N users, resulting in at least N^2 parameters to learn N user policies with N actions per policy. However, in contrast with dense networks where all agents interact (i.e., $\Omega(N^2)$ interactions), social networks are typically *sparse* (i.e., $O(N)$ interactions) since agents only interact with a constant number of other agents. In this case, since there are insufficient observations to capture $\sim N^2$ agent dependencies it is difficult to learn accurate policies. This is particularly challenging for large N due to the curse of dimensionality, which further increases variance.

To address these challenges, we propose to cluster users into $\mathcal{G}$ clusters and learn a policy π_G that maps the state representation of $\mathcal{G}$ clusters to $\mathcal{G}$ actions, where $\mathcal{G} \ll N$. We then combine this with a method to derive user-level actions from the cluster actions. We consider multi-stage policy learning, in which the policy is updated after regular time-intervals to capture dynamic user behavior. Since we want clusters to reflect the effects of applying the updated policy, we need to dynamically update cluster memberships while learning the policy. To ensure that the clusters are aligned across multiple stages, we propose a *weighted centroid* clustering approach.

Our DCPL approach starts by clustering the users and learning a set of Multivariate Hawkes Processes (MHP) [11] from the training data. The MHP models [8] characterize user activities in the network, and are used to simulate additional training data to obtain user states for learning the policy. We represent the user state $\mathbf{s}_U$ as the number of events for different network activities (e.g., sharing true news) and compute the state features of the clusters $\mathbf{s}_G$ by averaging the state features of their associated members. We define reward R as the correlation between exposures to fake and true news [5, 8]. To increase the spread of true news, we define user actions $\mathbf{a}_U$ as interventions to be applied to the intensity function of users' true news activity at each stage of the diffusion process.

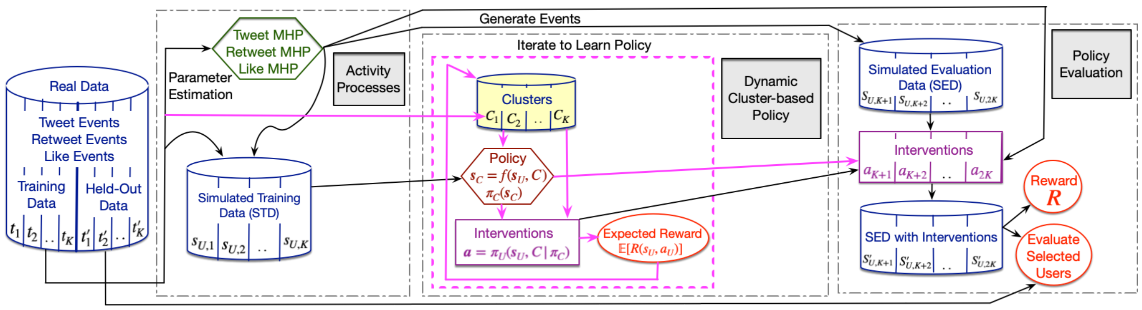

Figure 1: Overview of Cluster-Based Policy Learning and Evaluation

Given the network state, our goal is then to learn a cluster-level policy π_G to determine the actions for users, such that the total expected discounted reward is maximized.

To cluster users, we design *payoff* and *contribution* features inspired by the idea of *difference rewards* [2]. Payoff measures the change in the expected reward of a user, after applying the policy in the previous stage, which indicates the user's responsive to the past policy. Contribution is defined as the difference in the total expected reward obtained when applying the policy to a user, compared to not applying it to that user. This measures the effectiveness of a user in maximizing the reward, given the actions of other users.

Learning Cluster-Based Policy. Let there be K stages in the Simulated Training Data (STD) (Fig. 1). Given state features at the beginning of stage k (i.e., time τ_k), we use a multi-layer feed-forward neural network to learn a policy function π_G to obtain the actions to be applied during stage k (i.e., time-interval $[\tau_k, \tau_{k+1})$). We determine actions for the clusters, $\mathbf{a}_{G,k} = \pi_G(\mathbf{s}_{G,k})$, which correspond to interventions to be added to intensity for true news diffusion. Then, we compute actions for the users $\mathbf{a}_{U,k}$ by weighting the cluster actions by the user's distance to the centroid. This helps to capture both similarity and variability in user behavior. Next, we compute expected reward from these user actions and optimize the policy.

Update Clusters Dynamically. Using actions $\mathbf{a}_{U,k}$ learnt for stage k, we calculate the clustering features for the next stage, $\mathbf{X}_{k+1}$. Due to application of the policy, $\mathbf{X}_{k+1,i}$ is different from $\mathbf{X}_{k,i}$. Thus, we need to re-compute the centroids $\mathbf{Y}$ and cluster memberships $\mathcal{M}$. Additionally, we want the clusters to be aligned across different stages, so that the policies can be optimized using the neural network (for different clusters across multiple epochs). To achieve this, we define *weighted centroids* that include the effect of centroids in the previous stage. This helps to ensure that the centroids do not shift much and thus, clusters in stage $k+1$ can be aligned with those in stage k. Using $\mathbf{X}_{k+1}$, we obtain the updated centroids $\dot{Y}_{k+1,m} \forall m \in [1, \mathcal{G}]$ for stage $k+1$. We define $\mathcal{G}$ *weighted centroids* $\ddot{Y}_{k+1,m} = \varepsilon_1 \dot{Y}_{k+1,m} + \varepsilon \mathbf{Y}_{k,m} \forall m \in [1, \mathcal{G}]$, $\varepsilon_1 + \varepsilon_2 = 1$. Here ε_1, ε_2 indicate the importance assigned to the centroids from the previous stage and the current stage, respectively. After updating the centroids, we update the membership matrix and repeat until the centroids converge. Additionally, since the change in policy estimates across epochs reduces as optimization gets closer to convergence, the clustering features (which are dependent on these estimates), do not change much for the same stages across such epochs. Thus, we can reuse the learned clusters from stage k' of epoch e' as the clusters for stage k' in epoch $e' + 1$, without re-clustering users for any stage of epoch $e' + 1$. We use an approach similar to simulated annealing and only update the cluster assignments every $\eta_e \in \mathbb{Z}^+$ epochs gradually increasing η_e as the epoch number increases, which helps speed up convergence of policy learning.

Policy Evaluation. To evaluate the estimated policy $\hat{\pi}_G$, we simulate data again from the MHPs. Using the final clusters G_K, we obtain actions from the policy to add to the MHP intensity functions and generate evaluation data to assess empirical reward.

3 EXPERIMENTS

We use real-world datasets, Twitter 2016 and Twitter 2015 [12, 13], with 749 and 2051 users, respectively. We compare our approach to different *clustering* approaches that use reward features to obtain fixed (static) clusters, (eg. KM-R) [3, 18]. We also compare to *non-clustering* baselines (eg. NC-PF, NC-TR), which do not cluster users and/or simply add the clustering features to the state representation. Our DCPLapproach converges faster than the non-clustering methods and achieves a greater performance for all epochs. Clustering based approaches achieve lower variance, which indicates that they overcome sparsity and the curse of dimensionality. As expected, the non-clustering methods have larger variance and noisy estimates due to high dimensionality of state/action space. Additionally, DCPL, which updates cluster assignments dynamically outperforms KM-R, which assumes fixed assignments. In DCPL, the information about agents' payoff and contribution via clusters, helps faciliate exploration over the action space without increasing the state space. This helps the model to learn agents' effectiveness early on and converge faster to a better policy. We conducted additional experiments to study the effect of different network properties, as well as evaluate the learned clusters and the users selected by our model to spread true news.

4 CONCLUSION

This paper outlines a cluster-based policy learning approach for social RL that helps to address challenges due to high-dimensionality and sparsity. Specifically, we cluster users based on their payoff and contribution and aggregate the interactions of similar agents. We learn policies for the clusters, and then use those to obtain actions for individual users. This allows for efficiently learning personalized policies, while still considering all agent dependencies given a large number of users. Since the number of effective policies is greatly reduced, this also lowers the computational complexity. Experiments show that dynamic clustering of users helps the model to quickly learn better policy estimates, allowing it to outperform other static clustering-based and non-clustering alternatives. See [9] for a complete description of algorithms, methodology, and experiments.

REFERENCES

[1] Hunt Allcott and Matthew Gentzkow. 2017. Social media and fake news in the 2016 election. *Journal of Economic Perspectives* 31, 2 (2017), 211–36.

[2] Sam Devlin, Logan Yliniemi, Daniel Kudenko, and Kagan Tumer. 2014. Potential-based difference rewards for multiagent reinforcement learning. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*. International Foundation for Autonomous Agents and Multiagent Systems, 165–172.

[3] Ali el Hassouni, Mark Hoogendoorn, Martijn van Otterlo, and Eduardo Barbaro. 2018. Personalization of health interventions using cluster-based reinforcement learning. In *International Conference on Principles and Practice of Multi-Agent Systems*. Springer, 467–475.

[4] Mehrdad Farajtabar, Yichen Wang, Manuel Gomez Rodriguez, Shuang Li, Hongyuan Zha, and Le Song. 2015. Coevolve: A joint point process model for information diffusion and network co-evolution. In *Advances in Neural Information Processing Systems*. 1954–1962.

[5] Mehrdad Farajtabar, Jiachen Yang, Xiaojing Ye, Huan Xu, Rakshit Trivedi, Elias Khalil, Shuang Li, Le Song, and Hongyuan Zha. 2017. Fake news mitigation via point process based intervention. *arXiv preprint arXiv:1703.07823* (2017).

[6] Mehrdad Farajtabar, Xiaojing Ye, Sahar Harati, Le Song, and Hongyuan Zha. 2016. Multistage campaigning in social networks. In *Advances in Neural Information Processing Systems*. 4718–4726.

[7] Mehrdad Farajtabar, Safoora Yousefi, Long Q Tran, Le Song, and Hongyuan Zha. 2015. A Continuous-time Mutually-Exciting Point Process Framework for Prioritizing Events in Social Media. *arXiv preprint arXiv:1511.04145* (2015).

[8] Mahak Goindani and Jennifer Neville. 2019. Social Reinforcement Learning to Combat Fake News Spread. In *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence*.

[9] Mahak Goindani and Jennifer Neville. 2020. Cluster-Based Social Reinforcement Learning. *arXiv preprint arXiv:2003.00627* (2020).

[10] Mahak Goindani and Jennifer Neville. 2020. Social reinforcement learning. In *2020 AAAI Spring Symposium Series*.

[11] Alan G Hawkes. 1971. Spectra of some self-exciting and mutually exciting point processes. *Biometrika* 58, 1 (1971), 83–90.

[12] Yang Liu and Yi-Fang Brook Wu. 2018. Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[13] Jing Ma, Wei Gao, and Kam-Fai Wong. 2017. Detect rumors in microblog posts using propagation structure via kernel learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vol. 1. 708–717.

[14] Karishma Sharma, Feng Qian, He Jiang, Natali Ruchansky, Ming Zhang, and Yan Liu. 2019. Combating Fake News: A Survey on Identification and Mitigation Techniques. *arXiv preprint arXiv:1901.06437* (2019).

[15] Kai Shu, H Russell Bernard, and Huan Liu. 2019. Studying fake news via network analysis: detection and mitigation. In *Emerging Research Challenges and Opportunities in Computational Social Network Analysis and Mining*. Springer, 43–65.

[16] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. 2017. Fake news detection on social media: A data mining perspective. *ACM SIGKDD Explorations Newsletter* 19, 1 (2017), 22–36.

[17] Utkarsh Upadhyay, Abir De, and Manuel Gomez-Rodriguez. 2018. Deep Reinforcement Learning of Marked Temporal Point Processes. *arXiv preprint arXiv:1805.09360* (2018).

[18] Feiyun Zhu, Jun Guo, Zheng Xu, Peng Liao, Liu Yang, and Junzhou Huang. 2018. Group-driven reinforcement learning for personalized mhealth intervention. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 590–598.