Supplementary Material
Social Reinforcement Learning to Combat Fake News Spread

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

A PARAMETER ESTIMATION

We use two real-world datasets, Twitter 2016 and Twitter 2015 [Ma et al., 2017, Liu and Wu, 2018], with 749 and 2051 users in the networks, respectively. We observed that in our data around 75% of the news last for 40 hours, and thus we take $T = 40$ hours, with 40 stages of length $\Delta T = 1$ hour each. Also we observed that true news decays faster than fake news, and thus chose $\omega^F < \omega^T$. Specifically, we set $\omega^F = 0.6$ and $\omega^T = 1$ for Twitter 2016, and $\omega^F = 0.75$ and $\omega^T = 1$ for Twitter 2015. To decide ω, we performed a grid search and chose the one with least error (Eq. 2 in the main paper). We consider ω as a pre-specified hyper-parameter as in [Farajtabar et al., 2017], and did not estimate from data due to increased computational cost.

We infer parameters (μ, χ, Φ) using Maximum Likelihood Estimation as in [Zhou et al., 2013, Farajtabar et al., 2017, Farajtabar et al., 2015] using data from first 10 stages. Suppose we have l previously observed sequences $\mathcal{I} = \{o_q\}_{q=1}^{l}$, where each o is a sequence of events $e^o = \{(t_j^o, i_j^o, c_j^o)\}_{j=1}^{n_o}$ observed during the first 10 stages, and n_o is the number of events in o. Since we assume that the diffusion of fake ($c_j^o = F$) and true ($c_j^o = T$) news is independent, we separate the events corresponding to fake and true news diffusion, and learn the respective parameters separately. We provide a generic expression for likelihood, with exponential kernel for MHP:

$$L(\Theta) = \sum_{o \in \mathcal{I}} [\sum_{j=1}^{n_o} \log \lambda_{i_j^o}(t_j^o) - \sum_{i=1}^{N} \int_0^{T_{PE}} \lambda_i(t)dt)] \quad (1)$$

where T_{PE} corresponds to the time-stamp of the data used to estimate parameters (10 in our case). We minimize with L_1 regularization to avoid over-fitting,

$$\min_{\mu,\Phi} -L(\Phi, \mu) + \zeta_1 ||\mu||_1 + \zeta_2 ||\Phi||_1 \quad (2)$$

where $||\Phi||_1 = \sum_{i,j=1}^{N} \Phi_{ij}$, is used to enforce sparsity of the matrix Φ. To efficiently solve our optimization problem, we divide it into easily solvable sub-problems, based on the approach of Alternating Direction Method of Multipliers (ADMM) [Engelhard et al., 2018]. See [Zhou et al., 2013] for more details on efficient optimization.

B MODEL CHOICE

B.1 CONTROLS AT REGULAR SPANNED TIME-INTERVALS

We consider modeling controls at regular spanned time intervals due to the following reasons. Regulating policy at different time stamps helps model the dynamic behavior of people, for eg, a user is active for certain time period time and becomes inactive afterwards. Also, in our problem, we impose a budget constraint on the total amount of intensities allocated to users, and thus regulating their distribution is important.

B.2 DISCOUNTED REWARDS SETTING

In order for our policy to have long-term impact, we consider both immediate and future rewards. But, as a post ages, its influence decreases [Farajtabar et al., 2015]. In social networks, the feeds are chronologically sorted [Upadhyay et al., 2018], and thus, a user sees most recent posts from her peers than older. This indicates that reward from recent stages is more important than that from later stages in time. Since, our reward is based on number of exposures to a post, we use discounted rewards setting (as used in previous work (eg. [Farajtabar et al., 2017], [Zheng et al., 2018]), that is easier to realize using controls at regular spanned time stamps explained above.

C BOUNDED CONFIDENCE MODEL

According to Bounded Confidence Model (BCM), each user i has an opinion $x_i \in [0, 1]$. Two adjacent users i and j interact iff their opinions are close enough, i.e., $|x_i - x_j| \leq \epsilon \in [0, 0.5]$, resulting in a change in their opinions as $x_i = x_i + \mu(x_j - x_i)$ and $x_j = x_j + \mu(x_i - x_j)$. BCM assumes that the two interacting users must be close enough in their opinions (hold same set of beliefs), and the exchange increases one user's opinion and decreases another's. However, these assumptions do not hold in our case, since an interaction between users with similar ideology can increase the bias instead of reducing it. Moreover, we have a one-way interaction between a user and her followers instead of both ways. Therefore, we proposed a model to update the bias as follows.

D UPDATING POLITICAL BIAS

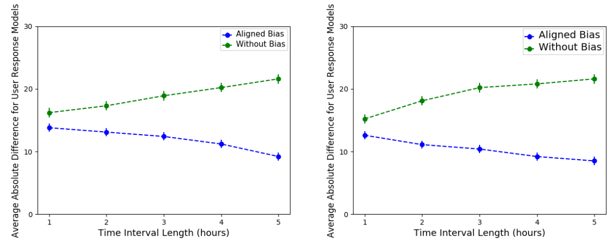
The political bias is updated based on the cumulative effect of interactions during the interval $[\tau_k, \tau_{k+1})$. Algorithm 1 describes the updation of political bias. The function ρ in lines 11 and 14 helps to maintain the bias values in [0, 1] as $\rho(x) = 0$ if $x \in [0, 0.5]$, and $\rho(x) = 1$ if $x \in (0.5, 1]$.

Algorithm 1 Update Political Bias

1: **Input:** $\mathcal{I}_k, \{\mathcal{A}_i\}_{i=1}^N, \{b_{i,k-1}\}_{i=1}^N$
2: /*Initialize bias values at stage k with those at stage $k-1$*/
3: **for** $i = 1 \ldots N$ **do**
4: $\quad b_{i,k}^D = b_{i,k-1}^D, \;\; b_{i,k}^R = b_{i,k-1}^R$
5: **end for**
6: /* Update bias based on interactions during stage k*/
7: **for** $e = (t, i, c) \in \mathcal{I}_k$ **do**
8: $\quad$ Let i be the user corresponding to the event e.
9: $\quad$ **for** each $j \in \mathcal{A}_i$ **do**
10: $\quad\quad$ **if** $b_{i,k-1}^D > b_{i,k-1}^R$ **then**
11: $\quad\quad\quad b_{j,k}^D = b_{j,k}^D + 0.5[|b_{i,k}^D - b_{j,k}^D| - \rho(|b_{i,k}^D - b_{j,k}^D|)]$
12: $\quad\quad\quad b_{j,k}^R = 1 - b_{j,k}^D$
13: $\quad\quad$ **else**
14: $\quad\quad\quad b_{j,k}^R = b_{j,k}^R + 0.5[|b_{i,k}^R - b_{j,k}^R| - \rho(|b_{i,k}^R - b_{j,k}^R|)]$
15: $\quad\quad\quad b_{j,k}^D = 1 - b_{j,k}^R$
16: $\quad\quad$ **end if**
17: $\quad$ **end for**
18: **end for**
19: **return** $\{b_{i,k}\}_{i=1}^N$

E EVALUATION OF PROPOSED USER RESPONSE PROCESSES

To test whether bias helps to model the response process better, we compare it with an alternative model *Without Bias User Response*, described below that doesn't consider political bias.



(a) Twitter 2016 (b) Twitter 2015

Figure 1: Difference in expected and observed number of likes

$$\psi_{u,i}(t) = \chi_i + \sum_{j \in A_{\cdot i}}^{N} \int_0^t \omega A_{ji} \, e^{-\omega t} \, dW_{uj}(s) \qquad (3)$$

To evaluate how well the proposed user response processes, *Aligned Bias User Response* (details in the Main Paper) and *Without Bias User Response* capture "like" events in the network, we use a similar setting as in Sec. 3.2.4 in the main paper.We compare the number of likes generated from the above models (after estimating the parameters) to those observed in the real data. The average absolute difference as a function of time interval length is shown in Figure 1. We can see that the Aligned Bias User Response model is a better than the other model that does not take into account bias.

F EXPECTED EVENTS IN FUTURE

F.1 EFFICIENT COMPUTATION OF INTENSITY

The diffusion of news is modeled as MHP, a non-Markovian process. However, since we map the problem to a Markov Decision Process, we need to include the effect of history from previous events. Let $\mathcal{H}_i^F$ be the effect of intensity due to all events in previous k stages, for user i, on the future $t > \tau_k$, for fake news diffusion. $\mathcal{H}_i^F = \omega^F \sum_{j=1}^{N} \int_0^{\tau_k} \Phi_{ij} \, e^{-\omega^F(t-s)} \, dF_j(s)$. As observed in previous work ([Farajtabar et al., 2016], [Simma and Jordan, 2012]), exponential kernel allows to efficiently compute the intensity, by defining $y_{i,k}^F = \lambda_i^F(\tau_k) - \mu_i^F$, so that, $\mathcal{H}_i^F = y_{i,k}^F e^{-\omega^F(t-\tau_k)}$ Hence, using $y_{i,k}^F$, we can efficiently compute the intensity at $t > \tau_k$, without having to sum over all previous k stages. Similarly, we define $y_{i,k}^T = \lambda_i^T(\tau_k) - a_i^{k-1} - \mu_i^T$ for the true news diffusion.

To incorporate the effect of past events, much of the recent work uses complete trajectories of users (eg.

[Upadhyay et al., 2018]), but our model is unique since MHP helps to include the effect of self and other agents' history and actions implicitly, as has also been adopted in [Farajtabar et al., 2017]. The efficient computation of history and summing it to one scalar per dimension as shown above, is space efficient as we do not need to store the complete trajectory per user.

F.2 EXPECTED NUMBER OF EVENTS

Following [Farajtabar et al., 2017] and [Farajtabar et al., 2016], we obtain,

$$\mathbb{E}[z_k^T] = \Gamma(\mu^T + a^k) + \Upsilon y_k^T$$
$$\mathbb{E}[z_k^F] = \Gamma \mu^F + \Upsilon y_k^F$$

where, y_k^F and y_k^T are as defined above, that capture the effect of history due to past events, and,

$$\Upsilon = (\Phi - \omega I)^{-1}(e^{(\Phi - \omega I)(\Delta)} - I)$$
$$\Gamma = \Upsilon + (\Phi - \omega I)^{-1}(\Upsilon - I(\Delta))/\omega.$$

Thus,

$$\mathbb{E}[R^k(s^k, \mathbf{a}^k)] = \frac{1}{N}(\Gamma(\mu^T + \mathbf{a}^k) + \Upsilon y_k^T)^T\, A^T A\, (\Gamma \mu^F + \Upsilon y_k^F) \tag{4}$$

The linear dependence of expected reward on policy $\mathbf{a}^k$ results in a convex optimization problem. Similarly, we calculate $\mathbb{E}[\nu^k]$. For more details, please refer [Farajtabar et al., 2016].

G POLICY FUNCTION APPROXIMATOR

We describe the details of the Neural Network used for learning optimal policy. The input to the NN, for stage k, is the state $s^k = [z_k^F, z_k^T, \nu^k]$. Hence, the dimensionality of input layer is $3N$, where N is the number of users in the network. Now, $V(s^k) = f(s^k; \phi)$, and $\mathbf{a}^{k+1} = \pi(s^k; \theta)$. We use the same NN to learn both θ and ϕ, however they are independent of each other. We use one hidden layer to learn the policy $\pi(\mathbf{a}^{k+1}|s^k)$, and a separate hidden layer to learn the value function $V(s^k)$. There are 2 different outputs of the NN, a scalar value $V(s^k)$, and an N-dimensional output corresponding to the action a_i^{k+1} for each user i. We used Adam optimizer and learning rate of 0.02. Figure 2 shows the network for two users.

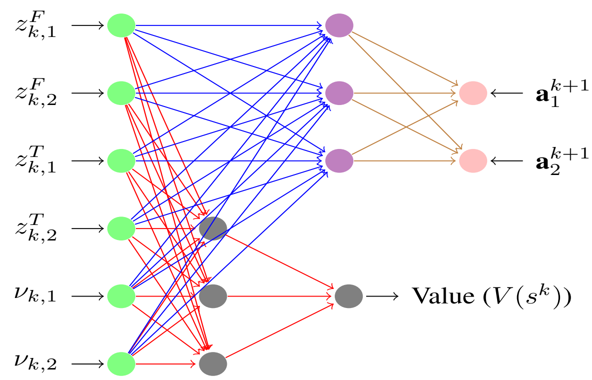


Figure 2: Neural Network Architecture

H EXPERIMENTS ON SEMI-SYNTHETIC DATA

We use subsets of twitter data to study the performance with respect to different network parameters. The results, on Twitter 2016, are shown in Fig 3, where we highlight the region closely representing real-world scenarios. We observe that our method outperforms the baselines by larger margin in all such regions.

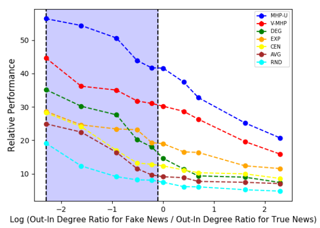
Ratio of out-degree to in-degree Fake news sources have lower out-degree and high in-degree, compared to sources for true news [Shu et al., 2018, Mccord and Chuah, 2011]. Fig 3-a shows that the performance decreases as the ratio of ratio of out-degree to in-degree for sources of fake news to that of true news, increases. This can be due to a decrease in the number of followers for true news sources. This ratio is usually less than 1 in real networks [Yue, 2017].

Average Degree [Bovet and Makse, 2019] shows that the average degree of users in fake news network (users who retweet fake news) is more than that in true news network. Fig 3-b shows that the performance decreases as the ratio of average degree for fake news network to true news network increases, since fewer people are reached out by true news spreaders.

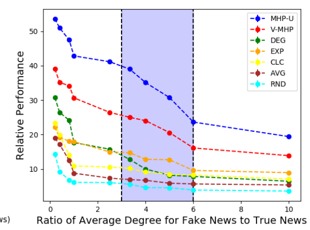
Centrality Closeness centrality for sources of fake news is higher than that for true news [Yue, 2017, Yang et al., 2013]. Fig 3-c shows change in performance with respect to ratio of average closeness centrality of fake news sources to that for true news. The performance is high for ratio 1.

Political Bias [Allcott and Gentzkow, 2017] observed that Democrats (D) are more likely to believe fake news than Republicans (R), and R are more likely to believe true news than D. We say that a user is High D (High R) if $b_{i,0}^D > 0.5$ ($b_{i,0}^R > 0.5$) and Low D (Low R) otherwise. Based on this, we create four groups of people. Group 1: High D fake, High R true news sources. Group 2: High D fake, Low R true news sources. Group 3: Low D fake, High R true news sources. Group 4: Low D fake, Low

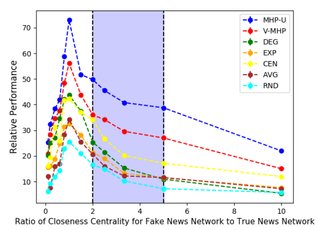
R true news sources. Fig 3-d shows that the performance is least for Group 1, when bias is high for both R and D, indicating that it is difficult to encourage people to spread news that does not align with their ideology.



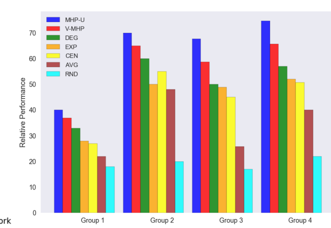
(a) Ratio of Out-to-in Degree



(b) Ratio of Average Degree



(c) Closeness Centrality



(d) Political Bias

Figure 3: Relative Performance (Different Network Properties)

References

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

[Bovet and Makse, 2019] Bovet, A. and Makse, H. A. (2019). Influence of fake news in twitter during the 2016 us presidential election. *Nature communications*, 10(1):7.

[Engelhard et al., 2018] Engelhard, M., Xu, H., Carin, L., Oliver, J. A., Hallyburton, M., and McClernon, F. J. (2018). Predicting smoking events with a time-varying semi-parametric hawkes process model. *arXiv preprint arXiv:1809.01740*.

[Farajtabar et al., 2017] Farajtabar, M., Yang, J., Ye, X., Xu, H., Trivedi, R., Khalil, E., Li, S., Song, L., and Zha, H. (2017). Fake news mitigation via point process based intervention. *arXiv preprint arXiv:1703.07823*.

[Farajtabar et al., 2016] Farajtabar, M., Ye, X., Harati, S., Song, L., and Zha, H. (2016). Multistage campaigning in social networks. In *Advances in Neural Information Processing Systems*, pages 4718–4726.

[Farajtabar et al., 2015] Farajtabar, M., Yousefi, S., Tran, L. Q., Song, L., and Zha, H. (2015). A continuous-time mutually-exciting point process framework for prioritizing events in social media. *arXiv preprint arXiv:1511.04145*.

[Liu and Wu, 2018] Liu, Y. and Wu, Y.-F. B. (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*.

[Ma et al., 2017] Ma, J., Gao, W., and Wong, K.-F. (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)*, volume 1, pages 708–717.

[Mccord and Chuah, 2011] Mccord, M. and Chuah, M. (2011). Spam detection on twitter using traditional classifiers. In *international conference on Autonomic and trusted computing*, pages 175–186. Springer.

[Shu et al., 2018] Shu, K., Wang, S., and Liu, H. (2018). Understanding user profiles on social media for fake news detection. In *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 430–435. IEEE.

[Simma and Jordan, 2012] Simma, A. and Jordan, M. I. (2012). Modeling events with cascades of poisson processes. *arXiv preprint arXiv:1203.3516*.

[Upadhyay et al., 2018] Upadhyay, U., De, A., and Gomez-Rodriguez, M. (2018). Deep reinforcement learning of marked temporal point processes. *arXiv preprint arXiv:1805.09360*.

[Yang et al., 2013] Yang, C., Harkreader, R., and Gu, G. (2013). Empirical evaluation and new design for fighting evolving twitter spammers. *IEEE Transactions on Information Forensics and Security*, 8(8):1280–1293.

[Yue, 2017] Yue, J. (2017). *Analyzing and Detecting Social Spammers with Robust Features*. PhD thesis, McGill University Libraries.

[Zheng et al., 2018] Zheng, G., Zhang, F., Zheng, Z., Xiang, Y., Yuan, N. J., Xie, X., and Li, Z. (2018). Drn: A deep reinforcement learning framework for news recommendation. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, pages 167–176. International World Wide Web Conferences Steering Committee.

[Zhou et al., 2013] Zhou, K., Zha, H., and Song, L. (2013). Learning social infectivity in sparse low-rank networks using multi-dimensional hawkes processes. In *Artificial Intelligence and Statistics*, pages 641–649.