Is now a good time? Using reinforcement learning to make optimal decisions about when to send push notifications.

Conor O'Brien*, Huasen Wu, Shaodan Zhai, Dalin Guo, Wenzhe Shi, Jonathan J Hunt*

Twitter Inc,

Email: {conoro, jjh}@twitter.com

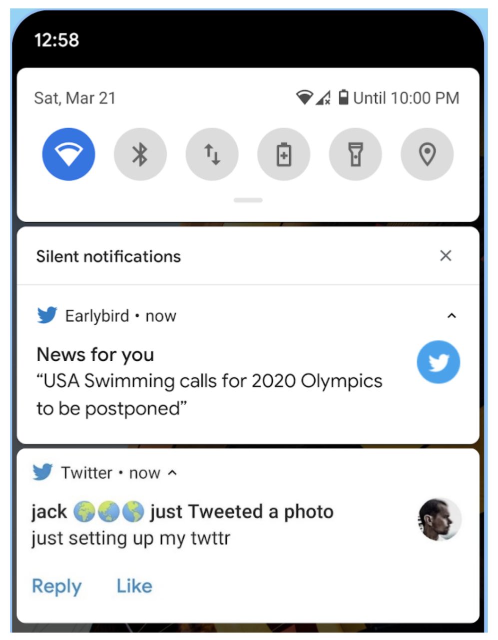
Summary

Push notifications are an important new paradigm for content consumption. Unlike most recommender systems contexts, push notifications require the system to decide when to send the user a notification. Sending too many notifications may create a poor experience from the user, causing them to disable notifications. Here we present a simple model-based reinforcement learning approach to optimizing decision making for notifications. We show in a production experiment on Twitter that it results in improved user retention.

Push notifications present new challenges for information retrieval and ranking. We hope to encourage further research in this topic.

More details on this work available at `https://arxiv.org/abs/2202.08812`

Filtering push notifications



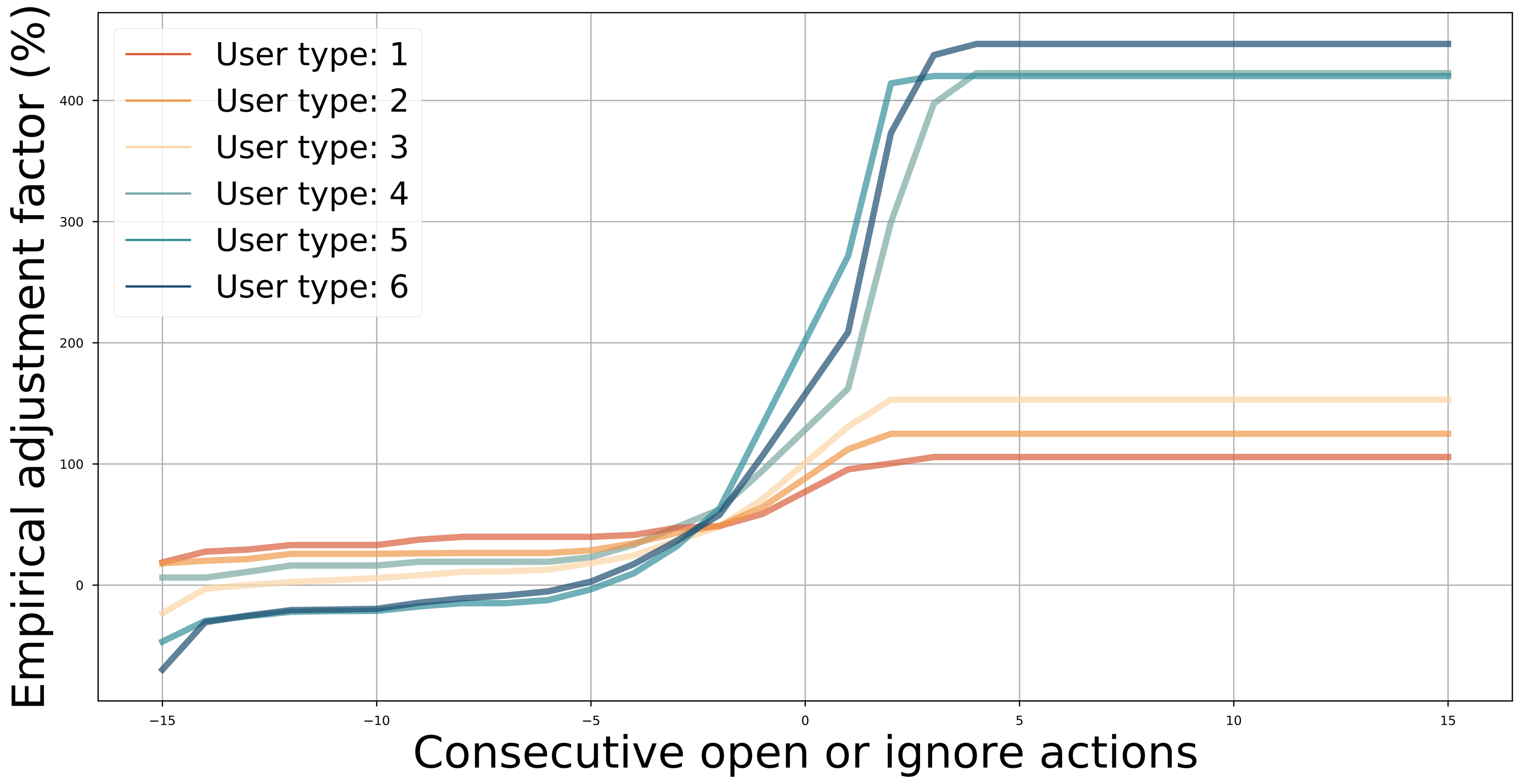
In a recommender system for push notification, at each timepoint, we also have a choice not to send any notification to the user.

- Sending the user too many or unwanted notifications may result in a poor user experience and cause them to uninstall the app or disable future notifications.
- We seek a policy $\pi(u, x)$ to decide whether to send a notification x to user u. If the notification is sent, we receive feedback $y \in \{0, 1\}$ indicating if the user opened the notification.
- We define the objective of the policy as the discounted sum of notifications opened:

$$J(\pi) = \mathbb{E}_{u \sim \mathcal{U}} \sum_{i=0}^{\infty} \gamma^i y_i(u, x) \quad (1)$$

User model

We used logged data to understand how user's future behavior is affected by their notifications.



The x-axis indicates how many consecutive positive, $y = 1$, or negative, $y = 0$, responses a user had to notifications immediately prior to their action on the current notification; we call this their "streak." We observed the users in a positive streak correlate with a significant increase in open rate for all user types.

Due to the limitations of the data, we can only observe the correlations, but cannot be sure this relationship is causal. We introduce a hyperparameter κ, for the fraction of the correlation that is causal, and try differing values of κ online.

Optimizing for the future

For every notification, we have an estimate from the ranking model of the likelihood the user will open this notification $\hat{p}(y = 1|u, x)$.

Using the user model we recursively solve the Bellman equations to determine what is the minimum value of $\hat{p}(y = 1|u, x)$ (as a function of user type and streak value) such that the expected value of sending the notification is greater than not sending it.

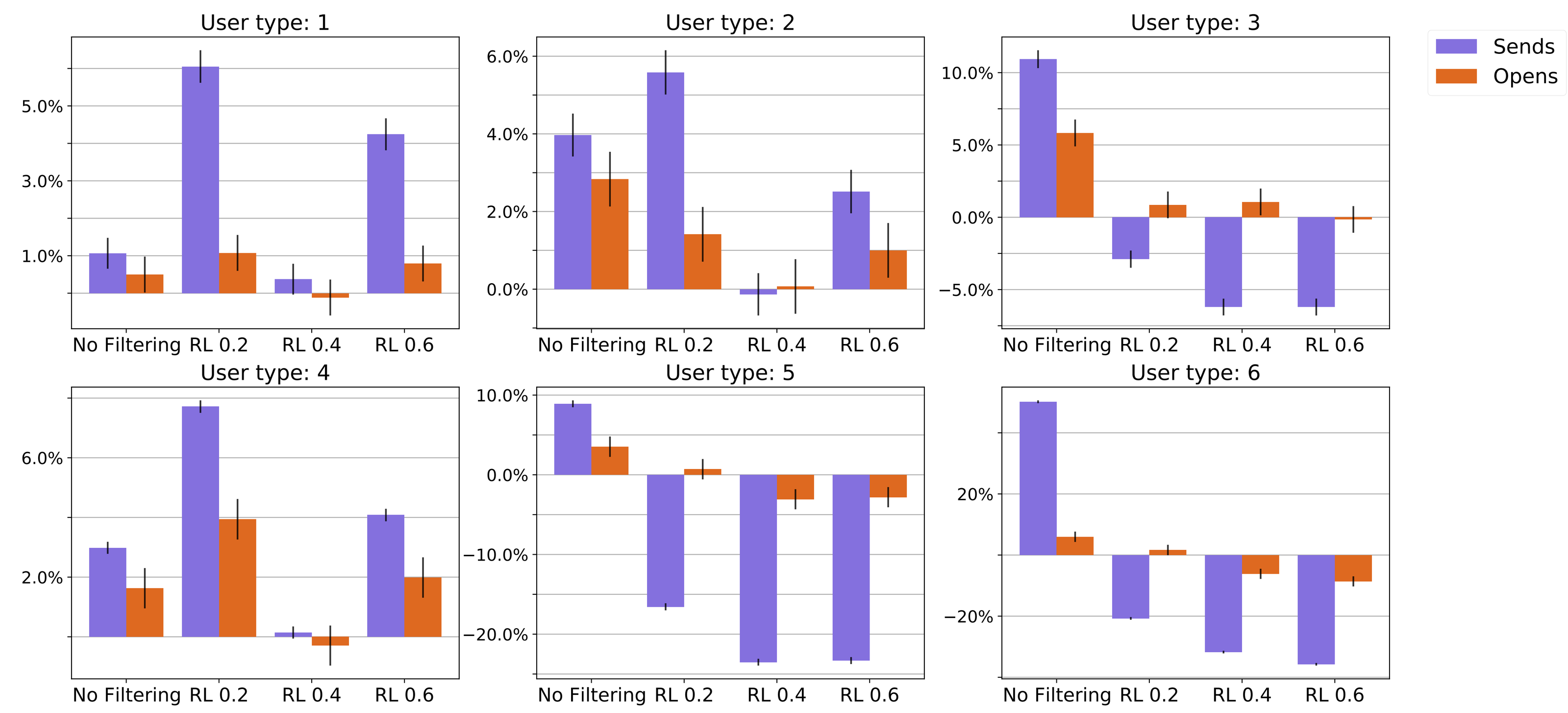
Results

We compared the RL policy (for different values of κ) in a large-scale production experiment.

The baselines were "no filtering" (all notifications are sent) and a heuristic (percentile) that simply filtered notifications if their score was below a certain percentile.

Treatment		**Metrics**			
Filtering Policy	*Send Limit*	*Total Sends*	*Open Rate*	*DAU*	*Reachability*
Percentile	-	-	-	-	-
No Filtering	0	+16.78%**	-11.97%**	+0.41%**	+0.08%
No Filtering	+1	+22.10%**	-14.40%**	+0.46%**	+0.00%
No Filtering	+2	+24.95%**	-15.52%**	+0.50%**	-0.08%
RL $\kappa = 0.2$	+2	-5.79%**	+7.96%**	+0.20%*	+0.09%
RL $\kappa = 0.4$	0	-13.08%**	+14.48%**	-0.34%**	+0.10%
RL $\kappa = 0.6$	+1	-12.65%**	+14.72%**	-0.22%**	+0.08%
RL $\kappa = 0.2$ tuned	+2.	+17.95%**.	-13.22%**	+2.3%**.	-0.05%

We found that an RL based policy for optimizing sends was able to substantially increase long-term user engagement (DAU).



We analysed the responses per user type. We found for user types 5 and 6 our RL model did not seem to be performing well, potentially due to less data for these user states.

By tuning the experiment to focus only on user types where the model performs well we significant increased performance (RL $\kappa = 0.2$ tuned).

Next steps

- Improve the causal reasoning of our approach.
- Incorporate additional features in decision making.