Recogym: A Reinforcement Learning Environment for the Problem of Product Recommendation in Online Advertising

David Rohde  
Criteo AI Labs  
Paris  
d.rohde@criteo.com

Stephen Bonner†  
Criteo AI Labs  
Paris  
st.bonner@criteo.com

Travis Dunlop†  
Universitat Pompeu Fabra  
Barcelona  
dunloptravis@gmail.com

Flavian Vasile  
Criteo AI Labs  
Paris  
f.vasile@criteo.com

Alexandros Karatzoglou  
Telefonica Research  
Barcelona  
aver:bektas@gmail.com

1 INTRODUCTION

Recommender systems are becoming more and more prevalent in all walks of modern life. They are particularly valuable when a user would otherwise be forced to do exhaustive searches in spaces with huge numbers of available options (either in terms of knowing what are the best products to buy, people to befriend, jobs to apply for) and that ultimately leads to user unhappiness (see The Paradox of Choice [19]). The theory of learning optimal recommendation policies is also evolving quite rapidly, and the limitations of the classical supervised approach to learning what to recommend are becoming apparent. Practitioners are increasingly reporting poor correlations between offline metrics and measured online performance.

In the context of online performance advertising, recommendation appears as the problem of choosing what item to advertise to the user given her current state and the current context in which the ad is presented (display medium, geography, time of day, etc.). By the nature of the problem, the user has two types of interactions with the items: first, the interactions on the e-commerce website, which we denote as organic sessions and secondly, the interactions on the publisher websites, through ads, which we denote as bandit sessions, inspired by the bandits literature which is represented heavily in the computational advertising field.

The problem of learning what product/item ads to show to each user is addressed differently in the recommendation literature and computational advertising literature. In the recommendation literature, this would be either solved as a missing link prediction task or as a next item prediction task; if time is important. The associated metrics for these approaches are classification/regression metrics such as area under the curve, mean squared error and negative log likelihood and respectively, ranking metrics such as precision@k and negative discounted cumulative gain.

Modern approaches to computational advertising employ randomized policies that ensure that exploration of actions using strategies such as epsilon-greedy or Thompson sampling. By adding randomization to the recommendation policy, we are able to not only compute the probability of an ad under the current policy, but also under any other recommendation policy using Inverse Propensity Scoring (IPS) [18]. This type of counterfactual estimation is explored...
in a series of papers [14, 5] [9] and to date, one item recommendation dataset with inverse propensity scores has been released, see [13].

However, these approaches suffer from problems when the recommendation policy to be evaluated is far from the logging policy, which results in very large inverse propensity scores for some of the items and due to capping, to bias in the estimation of the performance of the new policy.

To address the issues of IPS-based and other classical offline evaluation approaches, we propose RecoGym¹, a full reinforcement learning (RL) environment simulator for recommendation that allows, at evaluation time, the simulation of users’ reaction to arbitrary recommendation policies, without incurring the issue of exploding variance that is present in the case of datasets with IPS weights.

Since we are releasing this as an OpenAI Gym environment [4], we hope that this will allow both recommendation and RL practitioners to expand the type of machine learning approaches for recommendation and bring RL to other real-world applications, outside of games and self-driving cars.

In the first version of our simulator, we are releasing an environment that has the following key properties:

- The environment takes into account both the organic and the bandit user-item interactions.
- It contains a parameterized level of correlation between the two types of user behavior.
- It allows to parameterize the dimension of the hidden space in which the users and items are clustered.
- It allows to parameterize the level of relative impact of the user’s past exposure level to ads on the click-through rate of an ad display at a given time.

The rest of the paper is organized as follows: in Section 2 we review related work; in Section 3 we introduce the reinforcement learning problem. In Section 4 we provide some basic sanity checks that we expect a successful model of user reaction to in-ad recommendations to pass, and in Section 5 we conclude with an overview of contributions and planned next steps for future releases.

2 RELATED WORK

The relevant literature is currently split into two distinct parts: the classical recommendation literature, that considers modeling the organic user behavior and the computational literature that models the bandit user behavior.

The recommendation literature on modeling organic user behavior is vast: often, organic recommendation is framed as a user-item matrix completion task with matrix factorization as a common approach [6, 11]. Time-aware organic models use sequential models such as recurrent neural networks in order to predict next item viewed; see [16] for a recent survey.

Some works focused on organic datasets are aware of the tension between the task of modeling organic user behavior and the final goal of developing a ranking over potential recommendations. The ranking of organic user behavior is often used as a proxy to ranking bandit behavior - for a prominent example see [17]. In a similar spirit, [3] and [15] use the fact that the scoring phase of a recommendation engine must uniformly evaluate every action in order to adjust the organic model to be more appropriate for making recommendations. There are numerous public datasets of organic user behavior e.g. [8] [2], including some with temporal information[16].

The recommendation literature on modeling bandit user behavior is smaller: contextual bandits is a popular formulation of this type of problem [14], [5]; as is counter factual risk minimization [9]. Another approach is Gaussian process bandit optimization [12], [10]. There are also relatively few datasets for benchmarking such algorithms. [13] is one exception, unfortunately the requirement to use IPS for evaluation means that the measures of algorithmic performance are noisy.

Very importantly, to the best of our knowledge, there are currently no public datasets or indeed no other way to evaluate models that is able to incorporate both organic and bandit information.

3 FORMALIZING RECOMMENDATION AS A GAME

In Figure 1 we show our idealized (and heavily simplified) model of user activity, that alternates between organic e-commerce sessions and bandit publisher sessions and finishes by stopping her online activity for a longer period of time. In this setup, the advertising recommendations can only be shown during the publisher sessions. For simplicity, in this version of the model we assume that the user has a constant probability of conversion once she is back on one of the product shopping pages.

Under this formulation of the environment and attached rewards, the goal of the recommendation agent is to show personalized ads that increase the likelihood of the user to transition back to the e-commerce website, e.g. to increase the click-through rate of the user-ad pairs (shown as a dotted arrow in the figure below).

Figure 1: Markov Chain of the organic and bandit user sessions

3.1 Notation

A table of notation used is given in Table 1. An example of the type of data is given in Table 2.

In the example we see the timeline of user 10 over 6 time steps. All users start with an organic event at time 0 and in this case, user 10 visits product 104, then she transitions at time step 1 to product 52 and at time step 2 to product 71. At this point the user stops...
The modern advances in combining deep neural networks with RL for product and in this case the user responds positively and to create a unified API in which many RL benchmark problems of recommender systems. Our work will focus on designing and ing an RL environment specifically for the training and evaluation algorithms. However, to date, there has been little work on designing data source are becoming key to the creation of state-of-the-art environments in which RL algorithms can learn from a near unlimited many millions of simulated games of Go [20]. As such, simulated example, the recent success of Deep Mind’s AlphaGo was driven to massive quantities of simulated data from which to learn. For has been enabled by the ability of various algorithms to have access to information in order to make good recommendations and obtain a

\[ P \] Number of products
\[ u \] User
\[ t \] Time index (discrete) starting from 0
\[ z_{u,t} \] Event \( t \) for user \( u \) is organic or bandit
\[ v_{u,t} \] Organic product view or undefined if \( z_{u,t} = \text{bandit} \)
\[ v_{u,p,t} \] One hot coding of \( v_{u,t} \) where \( p \) denotes the product
\[ a_{u,t} \] Recommended or action product or undefined if \( z_{u,t} = \text{organic} \)
\[ c_{u,t} \] Click was made on recommendation \( a_{u,t} \) or undefined if \( z_{u,t} = \text{organic} \)

| Symbol | Meaning |
|--------|---------|
| \( P \) | Number of products |
| \( u \) | User |
| \( t \) | Time index (discrete) starting from 0 |
| \( z_{u,t} \) | Event \( t \) for user \( u \) is organic or bandit |
| \( v_{u,t} \) | Organic product view or undefined if \( z_{u,t} = \text{bandit} \) |
| \( v_{u,p,t} \) | One hot coding of \( v_{u,t} \) where \( p \) denotes the product |
| \( a_{u,t} \) | Recommended or action product or undefined if \( z_{u,t} = \text{organic} \) |
| \( c_{u,t} \) | Click was made on recommendation \( a_{u,t} \) or undefined if \( z_{u,t} = \text{organic} \) |

Table 1: Notation

| \( u \) | \( t \) | \( z_{u,t} \) | \( v_{u,t} \) | \( a_{u,t} \) | \( c_{u,t} \) |
|--------|--------|--------|--------|--------|--------|
| 10     | 0      | organic | 104    | NA     | NA     |
| 10     | 1      | organic | 52     | NA     | NA     |
| 10     | 2      | organic | 71     | NA     | NA     |
| 10     | 3      | bandit  | NA     | 42     | 0      |
| 10     | 4      | bandit  | NA     | 52     | 1      |
| 10     | 5      | organic | 52     | NA     | NA     |
|        |        |         |        |        |        |

Table 2: Example Data

In order to formalize these properties we describe three models; one that uses only the organic data, one that uses the bandit and a third that uses both attempts to get the best of both worlds.

### 4.2 Sanity Checks for an Agent that Combines Organic and Bandit information

The most successful algorithm will combine organic and bandit information in order to make good recommendations and obtain a low regret. Basic sanity checks for such an algorithm include that in the limit of large amounts of data it performs similarly to a pure bandit algorithm and that in the presence of less bandit data it is able to gain some advantages from the organic data.

In order to formalize these properties we describe three models; one that uses only the organic data, one that uses the bandit and a third that uses both attempts to get the best of both worlds.

Let \( v_{u,p,t} = 1 \) if user \( u \) organically viewed product \( p \) at time \( t \) and \( v_{u,p,t} = 0 \) otherwise. Furthermore define \( u_{p,f,t} \) such that \( u_{p,f,t} \sim \text{Bernoulli}(\sigma(\Lambda_{u,p,f,t})) \). Here \( \sigma(\cdot) \) is the logistic sigmoid and \( \Lambda_{u,p,f,t} \) is the log odds of user \( u \) organically viewing product \( p \) at time \( t \). This allows us to draw a formal link between the organic world and the bandit world in the equation below:
which has a Bernoulli distribution. The fact that
where \( \Phi \) to user
\( a \) the function
\( f \) is precisely in this case where purely organic algorithms are able to
incorporated.
then we may (subject to confirmation in the data) assume that a similar correlation exists
between users and actions in \( \epsilon_{u,a,t} \). In the first version of the simulator this will be implemented by users belonging
to hidden latent variables or clusters.
Now that we have formalized the connection between bandit and
organic we can sketch the relationships we expect to observe as we vary (a) the amount of bandit data available for training and (b) the
amount of noise \( \sigma_\epsilon \). The relationships we expect to see are shown in
Figure 2.
In Figure 2 a) we see that the “Pure Organic” behavior is determined by the amount of noise in \( \sigma_\epsilon \) and is unaffected by the number of bandit events. If \( \sigma_\epsilon \) is sufficiently small that the actions are correctly ordered it will perform well, in contrast if it is large it will perform poorly. The performance does not change as the number of bandit events increase as these are not used. In contrast the “Pure Bandit” algorithm is unable to make predictions without large numbers of bandit events. It performs poorly when this data is scarce and well when it is plentiful. The “Combined” model should achieve a happy medium using information from both sources and therefore able to outperform “Pure Bandit” for moderate numbers of bandit events the two agreeing only in the
limit of (often) unfeasibly large numbers of bandit events.
In Figure 2 a) we see that the “Pure Organic” performance reduces as \( \sigma_\epsilon \) breaks the connection between organic and bandit events. In contrast the combined and pure bandit do not exhibit this issue. As “Pure Bandit” ignores organic information we expect it to be invariant to \( \sigma_\epsilon \) and its performance will be determined by the amount of bandit events. Again we expect a “Combined” model to exhibit better performance so long as bandit events are not too plentiful.

4.3 Baseline Agents
In addition to the reco-gym environment, we also provide a selection of baseline agents to interact with it. These agents include the following:
- **Random** - This agent makes random recommendations and performs no learning from the organic data.
• Logistic - During training, this agent maintains a count of how many times a given user is exposed to a certain recommendation, as well as their response (click/no-click). Here the user is represented by their last organically viewed product, ignoring any information from the users organic sequence. To recommend a product, the logistic agent simply looks up the product with the highest CTR for the current user.

• Supervised-Prod2Vec - Inspired by the prod2vec algorithm [7], we implement a supervised user-product factorization like agent. In this agent, a user is again represented by their last organically seen product only. Clicks are predicted as a sigmoid over a linear transform of the user representation (their last organic product seen) and the recommended product representation.

5 CONCLUSIONS

In this paper we introduce RecoGym, the first Reinforcement Learning environment for recommendation in the context of online advertising. In the first release, the simulator supports the existence of user shopping and user publisher sequences, it allows the experimenter to specify the desired level of correlation between the two, to simulate various intrinsic dimensions for the user - item clusters and to vary the impact of repeated ad exposure on the ad click-through rate. In the paper we additionally cover some basic sanity checks on the behavior of the models trained on the simulator data and propose guidelines. Finally, we hope the release of this simulator will bring more interdisciplinary research between the reinforcement learning and recommender system communities and that subsequently, better alignment between offline and online performance can be achieved.

REFERENCES

[1] Marc G. Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. 2013. The arcade learning environment: An evaluation platform for general agents. Journal of Artificial Intelligence Research 47 (2013), 253–279.

[2] James Bennett, Stan Lanning, et al. 2007. The Netflix prize. In Proceedings of KDD cup and workshop, Vol. 2007. ACM, New York, NY, USA, 35.

[3] Stephen Bonner and Flavian Vasile. 2017. Causal Embeddings for Recommendation. In The Twelfth ACM Conference on Recommender Systems (RecSys 18), Vol. 1706. ACM, Vancouver, 9.

[4] Greg Bocekman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. OpenAI Gym.

[5] Miroslav Dudík, Daniel Hsu, Satyen Kale, Nikos Karampatziakis, John Langford, Lev Reyzin, and Tong Zhang. 2011. Efficient optimal learning for contextual bandits. In Uncertainty in Artificial Intelligence Proceedings of the Twenty Seventh Conference, Vol. 27. AUAI Press, Barcelona, 7.

[6] Mihajlo Grbovic, Vladan Radosavljevic, Nemanja Djuric, Narayan Bhamidipati, Jaikot Savla, Varun Bhagwan, and Doug Sharp. 2015. E-commerce in Your Inbox: Product Recommendations at Scale. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD ’15). ACM, New York, NY, USA, 1809–1818. https://doi.org/10.1145/2783258.2788627

[7] Mihajlo Grbovic, Vladan Radosavljevic, Nemanja Djuric, Narayan Bhamidipati, Jaikot Savla, Varun Bhagwan, and Doug Sharp. 2015. E-commerce in your inbox: Product recommendations at scale. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 1809–1818.

[8] F. Maxwell Harper and Joseph A Konstan. 2016. The Movielens datasets: History and context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4 (2016), 19.

[9] Thorsten Joachims, Artem Grotov, Adith Swaminathan, and Maarten de Rijke. 2018. Deep Learning with Logged Bandit Feedback. In Proceedings of the International Conference on Learning Representations (ICLR). CBLS, Vancouver, 12.

[10] Jaya Kawale, Hung H Bui, Branislav Kveton, Long Tran-Thanh, and Sanjay Chawla. 2015. Efficient Thompson Sampling for Online Matrix-Factorization Recommendation. In Advances in Neural Information Processing Systems 28, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett (Eds.). Curran Associates, Inc., Montreal, 1297–1305.

[11] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. Computer 42, 8 (Aug. 2009), 30–37. https://doi.org/10.1109/MC.2009.263

[12] Andreas Krause and Cheng S. Ong. 2011. Contextual Gaussian Process Bandit Optimization. In Advances in Neural Information Processing Systems 24. J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger (Eds.). Curran Associates, Inc., Granada, 2447–2455. http://papers.nips.cc/paper/4487-contextual-gaussian-process-bandit-optimization.pdf

[13] D. Lefortier, A. Swaminathan, Xiaotao Gu, T. Joachims, and M. de Rijke. 2016. Large-scale Validation of Counterfactual Learning Methods: A Test-Bed. http://arxiv.org/abs/1612.00367

[14] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. 2010. A contextual-bandit approach to personalized news article recommendation. In Proceedings of the 19th international conference on World wide web. ACM, New York, NY, USA, 661–670.

[15] Dawn Liang, Laurent Charlin, and David M Blei. 2016. Causal Inference for Recommendation.

[16] Massimino Quadranra, Paolo Cremonesi, and Dietmar Jannach. 2018. Sequence-aware recommender systems.

[17] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In Proceedings of the Twentieth Conference on Uncertainty in Artificial Intelligence (UAI ’09). AUAI Press, Arlington, Virginia, United States, 452–461.

[18] PR Rosenblum and DM Rubin. 1983. The central role of the propensity score in observational studies for causal effect. Biometrika 70, 1 (1983), 41–55.

[19] Barry Schwartz. 2004. The paradox of choice: Why less is more. Harper Perennial, New York.

[20] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. 2017. Mastering the game of Go without human knowledge. Nature 550, 7676 (2017), 354.