Recap: Designing a more Efficient Estimator for Off-policy Evaluation in Bandits with Large Action Spaces

Ajinkya More
amore@netflix.com
Netflix

Linas Baltrunas
lbaltrunas@netflix.com
Netflix

Nikos Vlassis
nvlassis@netflix.com
Netflix

Justin Basilico
jbasilico@netflix.com
Netflix

ABSTRACT

Contextual bandits are a common modeling tool in web-scale personalization systems such as entertainment recommendations and ad ranking. Usually, new algorithm variants are evaluated online via A/B testing. However, to reduce the number of variants tested online, it's crucial to have reliable metrics to test policies offline, using data obtained from production policies – a technique known as off-policy evaluation. One of the popular approaches for off-policy evaluation is the replay method using Inverse Propensity Scoring (IPS). This method is known to be unbiased but also suffers from high variance. Additionally, the method only makes use of those examples where the logging policy and target policy take the same action. In this paper, we present a new metric – *Recap*, that makes use of all the available logged data. *Recap* trades off bias for reduced variance for a more efficient off-policy evaluation.

KEYWORDS

Off-policy evaluation, Bandits, Personalization, Recommendations, IPS, Reinforcement Learning, Bias-Variance tradeoff, Replay

1 INTRODUCTION

Recommender Systems help people tame information overload by suggesting relevant items. At Netflix, recommendations are the main way members discover new movies and series to watch, accounting for over 80% [3] of what people discover. However, training and evaluating such systems that are driven by machine learning algorithms is a complex task. One of the open challenges in building such systems is to be able to asses the system's performance in an offline fashion by using collected data. Being able to reliably evaluate models offline allows for a faster innovation cycle, surfaces software bugs early and enables a more efficient use of precious A/B testing resources.

A typical modeling choice for large scale recommender systems is contextual bandits [1, 5, 6]. The explore/exploit nature of these methods call for specialized evaluation protocols. Using Off-policy evaluation techniques it is possible to estimate the quality of a target policy (or ranker) even if the logging policy that produced the data is not the same. This makes it an attractive tool for testing multiple strategies without deploying them to production. Popular approaches for off-policy evaluation are the Replay method [5], inverse propensity scoring (IPS) [2], self normalizing IPS (SNIPS) [7, 8] and Doubly Robust [2].

Replay and IPS are known to be unbiased but also suffer from high variance. In deterministic cases, these methods only make use of those examples where the logging policy and target policy take the same action. Therefore, the variance of these estimators increases as the number of arms increases and exact match becomes rarer. For large action spaces (>1K actions) the variance can be so high that these methods are rendered ineffective.

Our main contribution for this paper is introducing a class of low variance off-policy estimators for rank-based deterministic policies called *Recap*. Instead of requiring an exact match between logging and target policies to select the same action, *Recap* approximates the original deterministic policy with a new policy. This allows us to make use of all the available logged data in bandit policies. Effectively, we create a more efficient estimator by trading off bias for reduced variance. Using experimental evaluation we show that it has lower variance and lower risk than Replay, SNIPS and IPS. Moreover, we also show that this method has high on-line to off-line metric correlation for a Netflix production recommender system.

2 METRIC DEFINITION

Let $\mathcal{A} = \{1, 2, ..., K\}$ be the set of actions. In trial $t \in \{1, 2, ..., n\}$, the environment chooses a context x_t and in response the agent chooses an arm $a_t \in \mathcal{A}$. The environment then reveals a reward $r_{a,t} \in [0, 1]$. We assume the target policy is mediated by a scorer or a ranker (which is often the case), that assigns a real valued score $s(x, a)$ to each arm and selects the arm $\text{argmax}_{a \in \mathcal{A}} s(x, a)$. Many popular bandit policies like Thompson Sampling or UCB, naturally produce such scores. We want to estimate the expected reward $E[r_a]$ of a **deterministic** target policy τ given data from a logging policy π with the selected actions and the associated probability of selecting the particular logged action. Let us denote the probability of the logging policy taking action a for context x as $\pi(x, a)$ and the logged action at trial t as a_π. In this case, the replay-based IPS metric, can be defined as

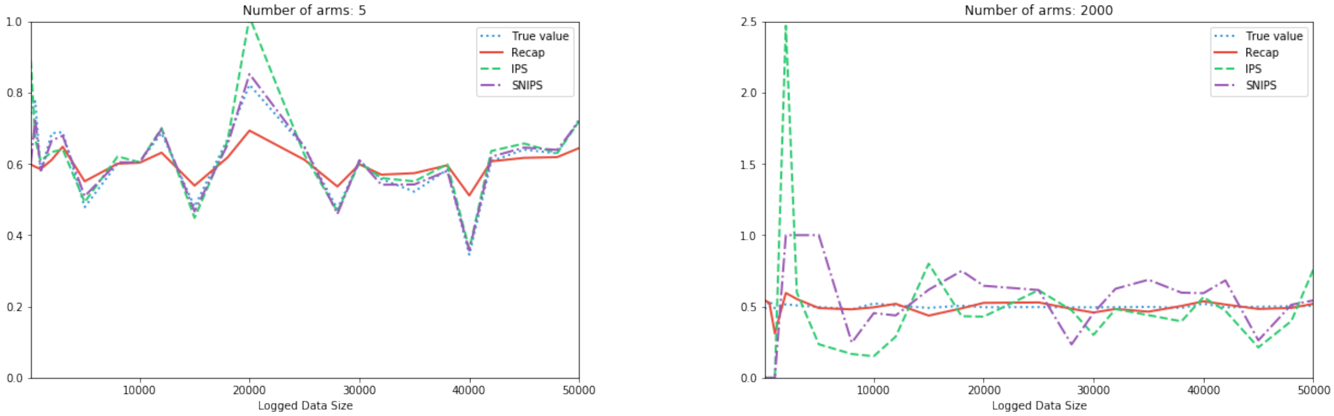
$$\hat{V}^{\tau}_{\text{IPS}} = \frac{\Sigma_{t=0}^{n} r_{a_\pi} \frac{\mathbb{I}(a_\tau = a_\pi)}{\pi(x_t, a_\pi)}}{n} \tag{1}$$

This metric is proven to be unbiased [2]. However, because the propensities $\pi(x, a)$ can get very small, the metric values can have a large variance depending on which actions get logged. A slightly biased version of IPS, which can reduce variance is the so-called self normalizing IPS or SNIPS [7, 8],

$$\hat{V}^{\tau}_{\text{SNIPS}} = \frac{\Sigma_{t=0}^{n} r_{a_\pi} \frac{\mathbb{I}(a_\tau = a_\pi)}{\pi(x_t, a_\pi)}}{\Sigma_{t=0}^{n} \frac{\mathbb{I}(a_\tau = a_\pi)}{\pi(x_t, a_\pi)}} \tag{2}$$

We introduce a new metric called ***Recap***, as follows

$$\hat{V}^{\tau}_{\text{Recap}} = \frac{\Sigma_{t=0}^{n} r_{a_\pi} \frac{RR}{\pi(x_t, a_\pi)}}{\Sigma_{t=0}^{n} \frac{RR}{\pi(x_t, a_\pi)}} \tag{3}$$



(a) Small (5) number of arms

(b) Large (2000) number of arms

Figure 1: Estimates vs data size

where

$$RR = \frac{1}{\Sigma_{a \in \mathcal{A}} \mathbb{I}(s(x,a) \geq s(x, a_\pi))}$$

is simply the reciprocal rank of the logged action according to the scores assigned by the target policy. For example, RR of action a_π is 0.5 if a_π is ranked second within all the actions by the policy τ. It is easy to see that the only difference between SNIPS and recap is using the quantity RR instead of $\mathbb{I}(a_\tau = a_\pi)$. Note that, RR is non-zero for all logged data points while the latter is by definition non-zero only for those data points where the logged action is the same as the action taken by the target policy. The intuition behind using reciprocal rank of the logged action instead of just using matched actions is that when a ranker places the logged action higher, it is assigned a higher "partial credit" (that rapidly decays as rank increases) even if it may not be at rank 1. The idea of "partial credit" has parallels in more general off-policy evaluation metrics for stochastic reinforcement learning policies (not the focus of this paper) [9], where a "partial credit" of $\tau(x_t, a_\pi)$ (the probability of target policy taking action a_π) is used in place of $\mathbb{I}(a_\tau = a_\pi)$.

More generally, for $m \in \mathbb{N}$ (or even $m \in \mathbb{R}^+$), we define,

$$\hat{V}^\tau_{\text{Recap}(m)} = \frac{\Sigma_{t=0}^n r_{a_\pi} \frac{RR^m}{\pi(x_t, a_\pi)}}{\Sigma_{t=0}^n \frac{RR^m}{\pi(x_t, a_\pi)}} \tag{4}$$

so that $\hat{V}^\tau_{\text{Recap}(1)} = \hat{V}^\tau_{\text{Recap}}$. Observe that, as $m \to \infty$, $\hat{V}^\tau_{\text{Recap}(m)} \to \hat{V}^\tau_{\text{SNIPS}}$. Thus, m allows us to smoothly trade off bias for variance. In the following sections we will compare *Recap* to IPS and SNIPS. We don't make a direct comparison to metrics such as doubly robust [2] since the obvious advantage of using *Recap* over such methods is that it obviates the need for any explicit model for rewards. We will also use IPS instead of plain replay as IPS can be seen as a more general form of replay where the logging policy π is not necessarily uniform.

3 SIMULATIONS

In this section we illustrate the properties of *Recap* using simulated data. We use the following set up.

(1) The rewards for arm a are drawn from a Bernoulli distribution with parameter $\theta_a = 1/a$
(2) The arm scores $s(x,a)$ are drawn from a normal distribution $\mathcal{N}(\mu, \sigma)$ where $\sigma = 0.1$ and μ is drawn from a uniform distribution with support $[0, 0.2]$.
(3) The target policy is $\text{argmax}_{a \in \mathcal{A}} s(x,a)$.
(4) At each trial, the logging policy samples the action from a multinomial distribution over the arms.

3.1 Behavior in small vs large action spaces

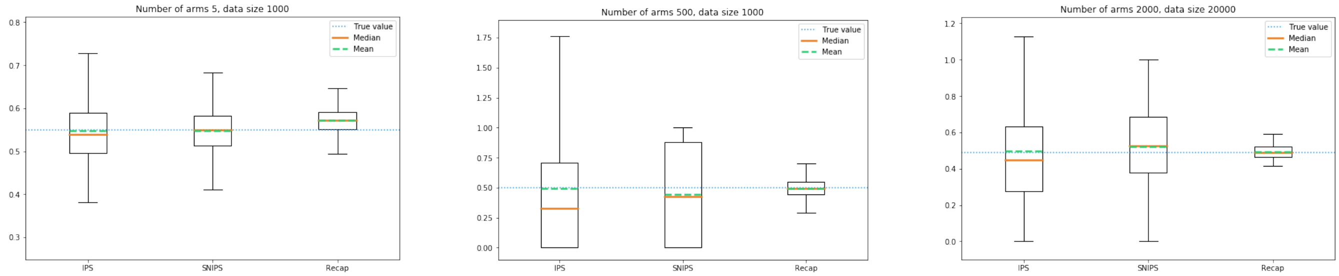
IPS and SNIPS metrics track the true expected reward or true value quite well when the action space is small (see Figure 1a). The bias introduced by *Recap* is quite visible in this case.

However, as the action space gets larger, the variance of these metrics is quite high since the probability of the target policy action matching the logged action reduces (see Figure 1b). This is the more common scenario for web scale systems. Recommendation systems like Netflix or Spotify typically need to choose among thousands of videos or music titles after a candidate selection step that filters down from a much larger catalog. Ad serving platforms like Google and Facebook deal with millions of ads. In this regime, *Recap* metrics can be more efficient than IPS or SNIPS. This difference is even more pronounced when the logged data size is small compared to the number of arms. In this case, there might be no matches for the target policy actions with the logged action causing IPS and SNIPS to collapse (as can be seen from some zero values for these metrics in Figure 1b). The *Recap* metric provides a much better approximation of the true reward in this situation.

3.2 Variance of the estimators

Both IPS and SNIPS have a larger variance than *Recap* in both small or large action spaces, while SNIPS has a lower variance than IPS (see Figure 2).

Further, when the data size is of the same order of magnitude as the number of arms, the target policy and logging policy may never match and IPS and SNIPS estimators evaluate to zero. For example, for our simulation dataset, we see that when the number



(a) Variance: Small (5) number of arms

(b) Variance: Medium (500) number of arms

(c) Variance: Large (2000) number of arms

Figure 2: Estimator variance for different arm sizes

of arms is 500 and the data size is 1000, the 25th percentile of IPS and SNIPS values is 0 as can be seen in Figure 2b, while the *Recap* estimate has a much lower variance around the true value.

3.3 Estimator risk

Figure 3 shows that, the risk [10] or mean squared error (squared bias plus variance) is smaller for *Recap* and also grows slower compared to IPS or SNIPS as the number of arms increase.

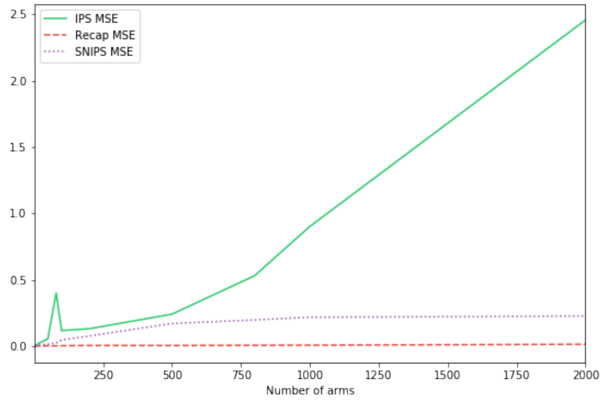


Figure 3: Mean squared error for the estimators vs number of arms

4 NETFLIX USE CASE

At Netflix, we conduct a lot of A/B tests to improve our machine learning models for continuously enhancing member experience. However, given that A/B test cycles tend to be long and expensive, it is desirable to have robust offline metrics in order to reduce the candidate set tested online as well as to identify potential issues early in the development process.

Most popular evaluation protocols for ranking approaches in Recommender Systems are derived from the field of Information Retrieval and statistical Machine Learning. However, in many Netflix scenarios measuring offline performance using metrics such as RMSE, AUC, MRR or Precision does not translate well to the online performance of the system. This is especially true when

(1) the data collected for ML model training comes from the production system itself (see position bias [4])
(2) the optimization objective does not fully capture the overall success metric of the task

To mitigate this problem, we define a weighted variant of the *Recap* metric

$$\hat{V}^{\tau}_{\text{Weighted Recap}} = \frac{\Sigma_{t=0}^{n} w_t r_{a_\pi} \frac{RR}{\pi(x_t, a_\pi)}}{\Sigma_{t=0}^{n} w_t \frac{RR}{\pi(x_t, a_\pi)}} \qquad (5)$$

where w_t is the per sample weight.

We propose to use $\hat{V}^{\tau}_{\text{Weighted Recap}}$ to evaluate an item ranker that produces a list of recommendations. First, we collect explore data by injecting random recommendations in the production system. More precisely, we use non uniform ϵ-greedy policy to generate a recommendation list of length K. At each rank in the item list, we select an item with the highest utility as predicted by a production model but with probability ϵ we choose a random item. We log the full rank together with the propensities of the random selection. Note that the target policy can be any ML model that produces a rank for the entire item catalog. As we have only one random item in the list we can use it as a random event in *Recap* off-policy evaluation, where the reward is 1 if that random item was played and 0 otherwise.

We found a strong correlation between $\hat{V}^{\tau}_{\text{Weighted Recap}}$ and online A/B test metrics across a range of experiments (see Figure 4). We currently use it to evaluate performance of various experimental rankers.

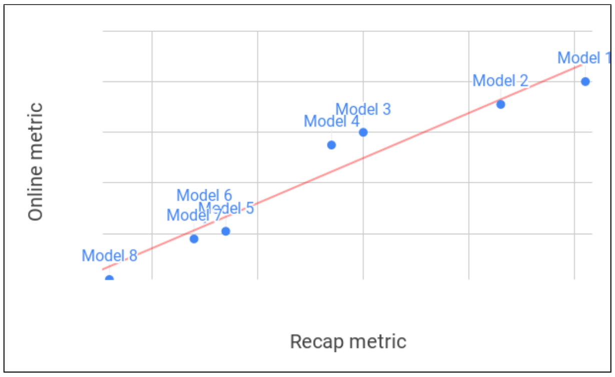


Figure 4: Comparison of online vs *Recap* metric for Netflix recommendations

5 CONCLUSION

In this paper, we introduced *Recap*, a new metric for efficient off-policy evaluation of bandit policies. We showed using simulated data that *Recap* has several desirable properties that are pertinent for large scale contextual bandit systems. We compared *Recap* to IPS and SNIPS and showed that it has a comparatively smaller variance and lower mean squared error, especially in large action spaces. Finally we described how using a weighted version of *Recap* has enabled us to get an early pulse on the online performance of our models at Netflix. In the future, we would like to study the theoretical properties of these estimators and determine more precisely the regimes in which these metrics would be preferable to alternatives.

REFERENCES

[1] Ashok Chandrashekar, Fernando Amat, Justin Basilico, and Tony Jebara. 2017. Artwork personalization at Netflix. *Medium. com* (2017).

[2] Miroslav Dudík, John Langford, and Lihong Li. 2011. Doubly robust policy evaluation and learning. *arXiv preprint arXiv:1103.4601* (2011).

[3] Carlos A Gomez-Uribe and Neil Hunt. 2016. The netflix recommender system: Algorithms, business value, and innovation. *ACM Transactions on Management Information Systems (TMIS)* 6, 4 (2016), 13.

[4] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, Filip Radlinski, and Geri Gay. 2007. Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. *ACM Transactions on Information Systems (TOIS)* 25, 2 (2007), 7.

[5] 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, 661–670.

[6] James McInerney, Benjamin Lacker, Samantha Hansen, Karl Higley, Hugues Bouchard, Alois Gruson, and Rishabh Mehrotra. 2018. Explore, exploit, and explain: personalizing explainable recommendations with bandits. In *Proceedings of the 12th ACM Conference on Recommender Systems*. ACM, 31–39.

[7] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. 2016. Recommendations as treatments: Debiasing learning and evaluation. *arXiv preprint arXiv:1602.05352* (2016).

[8] Adith Swaminathan and Thorsten Joachims. 2015. The self-normalized estimator for counterfactual learning. In *advances in neural information processing systems*. 3231–3239.

[9] Adith Swaminathan, Akshay Krishnamurthy, Alekh Agarwal, Miro Dudik, John Langford, Damien Jose, and Imed Zitouni. 2017. Off-policy evaluation for slate recommendation. In *Advances in Neural Information Processing Systems*. 3632–3642.

[10] Nikos Vlassis, Aurelien Bibaut, Maria Dimakopoulou, and Tony Jebara. 2019. On the Design of Estimators for Bandit Off-Policy Evaluation. In *International Conference on Machine Learning*. 6468–6476.