



HAL
open science

Offline A/B testing for recommender systems

Alexandre Gilotte, Clément Calauzènes, Thomas Nedelec, Alexandre Abraham, Simon Dollé

► **To cite this version:**

Alexandre Gilotte, Clément Calauzènes, Thomas Nedelec, Alexandre Abraham, Simon Dollé. Offline A/B testing for recommender systems. WSDM '18 - The 11th ACM International Conference on Web Search and Data Mining, Feb 2018, Los Angeles, United States. pp.198-206, 10.1145/3159652.3159687 . hal-02457457

HAL Id: hal-02457457

<https://hal.science/hal-02457457>

Submitted on 26 Jan 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Offline A/B testing for Recommender Systems

Alexandre Gilotte, Clément Calauzènes, Thomas Nedelec, Alexandre Abraham, Simon Dollé

Criteo Research
f.name@criteo.com

ABSTRACT

Online A/B testing evaluates the impact of a new technology by running it in a real production environment and testing its performance on a subset of the users of the platform. It is a well-known practice to run a preliminary offline evaluation on historical data to iterate faster on new ideas, and to detect poor policies in order to avoid losing money or breaking the system. For such offline evaluations, we are interested in methods that can compute offline an estimate of the potential uplift of performance generated by a new technology. Offline performance can be measured using estimators known as *counterfactual* or *off-policy* estimators. Traditional counterfactual estimators, such as *capped importance sampling* or *normalised importance sampling*, exhibit unsatisfying bias-variance compromises when experimenting on personalized product recommendation systems. To overcome this issue, we model the bias incurred by these estimators rather than bound it in the worst case, which leads us to propose a new counterfactual estimator. We provide a benchmark of the different estimators showing their correlation with business metrics observed by running online A/B tests on a large-scale commercial recommender system.

CCS CONCEPTS

• **Computing methodologies** → **Learning from implicit feedback**; • **Information systems** → *Evaluation of retrieval results*;

KEYWORDS

counterfactual estimation, off-policy evaluation, recommender system, importance sampling.

ACM Reference Format:

Alexandre Gilotte, Clément Calauzènes, Thomas Nedelec, Alexandre Abraham, Simon Dollé. 2018. Offline A/B testing for Recommender Systems. In *Proceedings of 11th ACM International Conf. on Web Search and Data Mining (WSDM 2018)*. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3159652.3159687>

1 INTRODUCTION

Personalized product recommendation has become a central part of most online marketing systems. Having efficient and reliable methods to evaluate recommender systems is critical in accelerating the pace of improvement of these marketing platforms.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

WSDM 2018, February 5–9, 2018, Marina Del Rey, CA, USA

© 2018 Copyright held by the owner/author(s). Publication rights licensed to Association for Computing Machinery.

ACM ISBN 978-1-4503-5581-0/18/02...\$15.00

<https://doi.org/10.1145/3159652.3159687>

Online A/B tests became ubiquitous in tech companies in order to make informed decisions on the rollout of a new technology such as a recommender system. Each new software implementation is tested by comparing its performance with the previous production version through randomised experiments. In practice, to compare two technologies, a pool of units (e.g. users, displays or servers) of the platform is split in two populations and each of them is exposed to one of the tested technologies. The careful choice of the unit reflects independence assumptions under which the test is run. At the end of the experiments, business metrics such as the generated revenue, the number of clicks or the time spent on the platform are compared to make a decision on the future of the new technology.

However, online A/B tests take time and cost money. Indeed, to gather a sufficient amount of data to reach statistical sufficiency and be able to study periodic behaviours (the signal can be different from one day to the other), an A/B test is usually implemented over several weeks. On top of this, prototypes need to be brought to production standard to be tested. These reasons prevent companies from iterating quickly on new ideas.

To solve these pitfalls, people historically relied on offline experiments based on some rank-based metrics, such as NDCG [13], MAP [1] or Precision@K. Such evaluations suffer from very heavy assumptions, such as independence between products or the fact that the feedback (e.g. click) can be translated into a supervised task [10, 16, 22]. To overcome these limitations, some estimators were introduced [2, 15] to estimate offline – i.e. using randomised historical data gathered under only one policy – a business comparison between two systems. In the following, we shall call the procedure of comparing offline two systems based on some business metric defining the outcome an *offline A/B test*.

This setting is called *counterfactual reasoning* or *off-policy evaluation* (OPE) (see [2] for a comprehensive study). Several estimators such as *Basic Importance Sampling* (BIS, [8, 11]), *Capped Importance Sampling* (CIS, [2]), *Normalised Importance Sampling* (NIS, [21]) and *Doubly Robust* (DR, [6]) have been introduced to compute the expected reward of the tested technology π_t based on logs collected on the current technology in production π_p .

All these estimators achieve different trade-offs between bias and variance. After explaining why BIS (Section 3) and DR (Section 4.1) suffer from high variance in the recommendation setting, we shall focus on *capped importance sampling* (Section 4.2). [2] proposed clipping the importance weights, which leads to a biased estimator with lower variance. However, the control of the bias is very loose in the general case and [2] only advocates modifying the source probability (the current system) to further explore whether a light clipping could be sufficient.

The main caveat of such biased estimates is that a low bias is present only under unrealistic conditions (see Section 4.2). Our main contribution is to propose two variants of *capped importance*

sampling that achieve low bias under much more realistic conditions (Section 5.3 and Section 5.4) and show their practical interest on real personalised recommendation systems. In Section 6, we compare metrics observed during real online A/B tests with the pre-computed values of the different counterfactual estimators.

2 SETTING AND NOTATION

We consider recommender systems in the context of online product recommendation. The task consists of displaying a set of products to a user on some e-commerce websites or on some advertising banners. These subsets of products should be personalised based on the interests of the user. This task is formalised as a ranking task, and not only a top-K retrieval, because the different product slots are not equivalent and exhibit different performance [5]. The system outputs a ranked list of products and then maps better products to better positions.

A recommendation policy is designed as a distribution over the top-K rankings. Good examples of distributions are the Bradley-Terry Luce (BTL) model [3], the Plackett-Luce model [4, 7, 20] or more generally Thurstonian models [26, 28].

In the following, we will represent random variables with capital letters such as Y and realisation of random variables with lower-case letters such as y . Given a display x represented by a set of contextual features as well as a set of eligible products, the recommender system outputs a probability distribution $\pi(A|X)$ where a is a top- K ranking on the eligible products and K is the number of items that will be displayed. Taking action a in state x generates a reward $r \in [0, r_{\max}]$ that could be interpreted as a click or a purchase.

3 ONLINE AND OFFLINE A/B TESTING

In *online* A/B tests, the objective is to compare two systems *prod* and *test* to ultimately take a decision on which one performs better than the other based on their respective business value. We consider a set of n units x that are randomly assigned to either *prod* or *test* and seek to measure the average difference in value, based on the reward signal $r \in [0, r_{\max}]$ that could be the number of clicks or the generated revenue. The choice of the units, which could be internet users, recommendation opportunities or servers, heavily depends on the independence assumptions made in order to reach a statistically significant decision in a timely manner. This assumption is called the isolation assumption [2].

We will note the current production policy π_p and the test policy π_t . To compare π_p and π_t , we estimate the average difference of value $\Delta\mathcal{R}$ which is called average treatment effect and defined as

$$\Delta\mathcal{R}(\pi_p, \pi_t) = \mathbb{E}_{\pi_t}[R] - \mathbb{E}_{\pi_p}[R]$$

where $\mathbb{E}_{\pi_p}[R] = \mathbb{E}[R|A]\pi_p(A|X)\mathbb{P}(X)$. During an online A/B test, units are randomly split in two populations \mathcal{P}_t and \mathcal{P}_p , such that we can estimate $\Delta\mathcal{R}$ using

$$\Delta\mathcal{R}(\pi_p, \pi_t) = \mathbb{E}[R|X \in \mathcal{P}_t] - \mathbb{E}[R|X \in \mathcal{P}_p].$$

$\Delta\mathcal{R}$ is estimated by Monte-Carlo using the two datasets collected during the test $\mathcal{S}_p = \{(x_i, a_i, r_i) : i \in \mathcal{P}_p\}$ and $\mathcal{S}_t = \{(x_i, a_i, r_i) : i \in \mathcal{P}_t\}$. We build the empirical estimator $\Delta\hat{\mathcal{R}}$ to take the decision by performing a statistical test :

$$\Delta\hat{\mathcal{R}}(\pi_p, \pi_t) = \hat{\mathcal{R}}(\mathcal{S}_t) - \hat{\mathcal{R}}(\mathcal{S}_p)$$

where $\hat{\mathcal{R}}(\mathcal{S})$ is the empirical average of rewards over \mathcal{S} gathered during the *online* AB test.

To perform an *offline* A/B test, we have only one set of n historical i.i.d. samples $\mathcal{S}_n = \{(x_i, a_i, r_i) : i \in [n]\}$ collected using a production recommender system π_p (also known as the *behaviour policy* in the RL community or *logging policy*). The goal is to compare the performance of a new technology, a test policy denoted π_t , to our current system π_p ¹. We can directly estimate $\mathbb{E}_{\pi_p}[R]$ using $\hat{\mathcal{R}}(\mathcal{S}_n)$, but for $\mathbb{E}_{\pi_t}[R]$ we cannot use a direct estimation since we do not have any data gathered under π_t . One of the main tools to estimate the expected reward under the target policy using rewards gathered under the behaviour policy is *importance sampling* or *inverse propensity score* as introduced by Hammersley and Handscomb [8] which leads to the following Monte-Carlo estimator:

$$\hat{\mathcal{R}}^{\text{IS}}(\pi_t) = \frac{1}{n} \sum_{(x, a, r) \in \mathcal{S}_n} w(a, x)r \quad \text{where } w(a, x) = \frac{\pi_t(a|x)}{\pi_p(a|x)}$$

The main advantage of such an estimator that it is unbiased, while its main pitfall is usually its high variance, which depends on how different π_t is from π_p (this variance is unbounded). All the difficulty here resides in the size of the action space. The number of top- K rankings over candidate sets of size M is extremely high ($K! \binom{M}{K}$), resulting in high variance of the importance weights W .

Several variants of importance sampling have been proposed, with different purposes, to tackle the high variance that can arise from the use of importance sampling. They use some classical variance reduction techniques, such as difference control variate or ratio control variate. However these approaches consist of using unbiased or consistent estimators and turn out to still lead to estimators suffering from high variance. An approach to trade off variance for bias is to clip the importance weights as proposed in [2]. In the next section, we detail several classic methods used in importance sampling for counterfactual reasoning.

4 REDUCING ESTIMATORS VARIANCE

4.1 Control Variates

Control variates are a popular variance reduction method in statistics. It consists of finding a second random variable with known expectation and which is correlated with the variable to estimate in order to reduce the variance of its estimation.

Doubly robust estimator. The easiest case is when we dispose of external knowledge like a reward model. We can use this as a control variate to improve our current estimator [6]. We assume we can access a model $\bar{r}(a, x)$ that estimates the expected reward of each action a at context x . We define the *doubly robust* estimator:

$$\hat{\mathcal{R}}^{\text{DR}}(\pi_t) = \sum_{(x, a, r) \in \mathcal{S}_n} \left((r - \bar{r}(a, x)) w(a, x) + \mathbb{E}_{\pi_t}[\bar{r}(A, X)|X = x] \right).$$

This estimator is unbiased and, if the predicted reward $\bar{r}(A, X)$ is well correlated with the actual reward R , it has lower variance than IS (see [19, §8.9] for instance).

¹We need a stochastic policy for π_p that puts a non-zero probability on any ranking a to prevent spurious correlations from biasing the estimators we exhibit in the following sections (see [2] for details).

However, this estimator has several drawbacks in the setting of recommendation systems. First, having an accurate model of the reward given the action is challenging when the number of possible actions is very large. To overcome this problem, Williams [27] – in the context of reinforcement learning – proposed to use a model \bar{r} not dependent on the action a . It has a higher variance than the initial DR model but avoid the marginalization over all the actions.

However, this approach does not solve the second and biggest drawback of this method: when the reward has a high variance even conditionally to X and A , the predicted reward cannot have a strong correlation with it. For instance, when the computed metric is the number of clicks – i.e. the reward R follows a Bernoulli with parameter close to 0 – the actual expected reward per action (typically around 10^{-3} in display advertising for instance) hardly correlates with R (0 or 1). In this particular case, DR is very close to IS: the use of the click model does not help to reduce the variance. Even if no good model is available, another control variate can be implemented to decrease the variance of IS.

Normalised importance sampling. We know that $\mathbb{E}_{\pi_p}[W] = 1$. Using the empirical average $\frac{1}{n} \sum_{(x,a,r) \in \mathcal{S}_n} w(a,x)$ as a global ratio control variate, we have the normalized importance sampling (NIS) estimator [21, 24]:

$$\hat{\mathcal{R}}^{\text{NIS}}(\pi_t) = \frac{1}{\sum_{(x,a,r) \in \mathcal{S}_n} w(a,x)} \sum_{(x,a,r) \in \mathcal{S}_n} w(a,x)r$$

The normalizing constant is equal to the sum of the importance weights and is equal in expectation to n , the number of examples in the dataset. It is a biased estimate of the expected reward but with lower variance than the *basic importance sampling* estimator. It is a consistent estimator of $\mathbb{E}_{\pi_t}[r]$ and the bias decreases in $1/n$. Thus NIS, with a certain amount of data is very close to BIS and the variance is not decreased.

The main problem of such methods aimed at reducing the variance without introducing any bias (at least asymptotically) is that if we do not dispose of a strong external knowledge, we do not reduce the variance that much.

4.2 Capping weights

Capped importance sampling. Capping weights is another way to control the variance of the IS estimator. Two forms of capping were introduced: max capping and zero capping. For some capping value $c > 0$, these estimators are respectively defined as:

$$\hat{\mathcal{R}}^{\text{maxCIS}}(\pi_t, c) = \frac{1}{n} \sum_{(x,a,r) \in \mathcal{S}_n} \min(w(a,x), c)r$$

and

$$\hat{\mathcal{R}}^{\text{zeroCIS}}(\pi_t, c) = \frac{1}{n} \sum_{(x,a,r) \in \mathcal{S}_n} \mathbf{1}_{w(a,x) < c} w(a,x)r$$

In the following, we denote the capped weights by $\bar{w}(a,x)$, for zero capping $\bar{w}(a,x) = \mathbf{1}_{w(a,x) < c} w(a,x)$ and for max capping $\bar{w}(a,x) = \min(w(a,x), c)$. All calculations presented in the following – except when explicitly specified otherwise – are valid for both zero capping and max capping. Intuitively, the behaviour of these two estimators is the same since the capped importance weights are

very big compare to the capping parameter: in Fig. 1, we provide some empirical results on weights encountered when evaluating recommendation system policies. Both capping methods show very similar results and we only report max capping results in the experiments.

However, capping comes at the cost of introducing a bias: when introducing capping on the IS estimator, we only account for a sub-part of $\mathbb{E}_{\pi_t}[R]$:

$$\begin{aligned} \mathbb{E}_{\pi_t}[R] &= \mathbb{E}_{\pi_t} \left[R \frac{\bar{W}}{W} \right] + \mathbb{E}_{\pi_t} \left[R \frac{W - \bar{W}}{W} \right] \\ &= \underbrace{\mathbb{E}_{\pi_p}[\hat{\mathcal{R}}^{\text{CIS}}(\pi_t, c)]}_{\mathcal{R}^{\text{CIS}}(\pi_t, c)} + \underbrace{\mathbb{E}_{\pi_t} \left[R \frac{W - \bar{W}}{W} \mathbf{1}_{W > c} \right]}_{\mathcal{B}^{\text{CIS}}(\pi_t, c)} \mathbb{P}_{\pi_t}(W > c) \end{aligned}$$

One of the main issues of only estimating $\mathcal{R}^{\text{CIS}}(\pi_t, c)$ is that the bias term $\mathcal{B}^{\text{CIS}}(\pi_t, c)$ becomes low only if $\mathbb{E}_{\pi_t}(R|W > c)$ is low. It means that $\mathbb{E}[R|A, X]$ has to be low for all a such that $w(a,x) > c$ – i.e. for all actions that π_t chooses much more often than π_p . As our test policy π_t is usually a trial for improving the current system π_p , it is not really satisfying to have an estimator with a low bias only if π_t performs poorly on actions it chooses more often than the current system. More formally, as we want to take a statistically significant decision, we need to build a confidence interval around $\hat{\mathcal{R}}^{\text{CIS}}(\pi_t, c)$. We can bound $\mathcal{R}^{\text{CIS}}(\pi_t, c)$ using any concentration bound (e.g. an empirical Bernstein bound [2, 17]). However the bias term can only be controlled in the worst case: $0 \leq \mathcal{B}^{\text{CIS}}(\pi_t, c) \leq r_{\text{max}}(1 - \mathbb{P}(W \leq c))$. As explained right before, this inequality only gets tight when r_{max} is lower on the capped volume than elsewhere.

4.3 No good practical trade-off for CIS

In practice, no capping parameter for CIS yields confidence interval small enough to decide whether π_t is a better policy than π_p . The capping parameter used in CIS introduces a bias-variance tradeoff: Increasing its value decreases the bias and increases the variance of the estimator. However, we demonstrated in the previous section that a volume of capped weights too high will lead to an invalid bias-variance tradeoff. Based on experimental results, we show that this critical limit is often reached for recommendation systems used in production.

We consider an A/B test that was implemented in production and look at the importance weights used to compute the performance of π_t based on logs gathered with π_p . Figure 1 shows their distribution according to the test policy. Figure 2 shows the variance and the upper bound on the bias (see section 4.2 for definition) depending on the capping parameter. We clearly see the tradeoff between the two measures when the capping parameter changes. In both cases, we can determine the values of the capping parameter for which the variance and bias are lower to the uplift that we want to measure offline (usually, we consider a 1% uplift). Figure 2 shows that no value of the capping parameter satisfies both criteria: a good variance is achieved below 10^2 whereas a good value for bias requires a capping parameter above 10^{23} .

This problem led us to design new estimators that model the bias introduced by capping and achieve better bias-variance tradeoff.

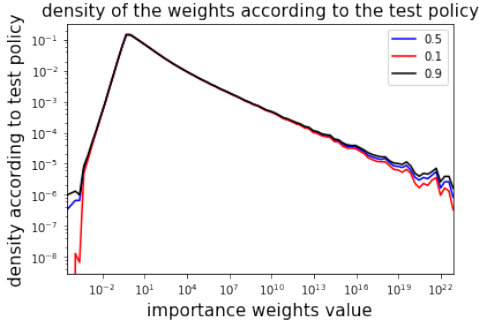


Figure 1: Distribution of the importance sampling weights when sampled according to the test policy with 80% confidence interval. (0.1 corresponds to 10th centile and 0.9 to 90th centile)

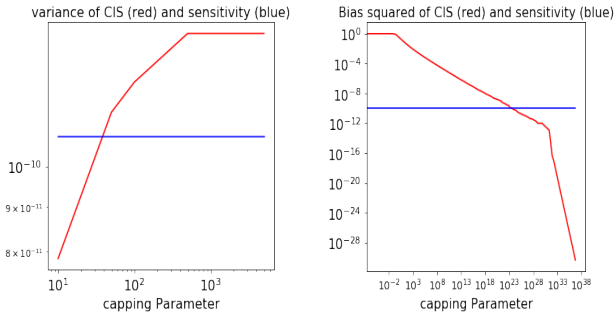


Figure 2: Variance of CIS, upper bound on the bias of CIS and sensitivity depending on the capping parameter.

5 MODELLING THE BIAS

As discussed in the previous section, CIS can not reach a tradeoff with low variance and provable low bias. In the following, we present different estimators that model the bias at different scale. We show that the well-known Normalised Capped Importance Sampling provides a model at the global level. Then, we present a new estimator that models the bias at a contextual state level.

5.1 Global bias model

Normalised Capped Importance Sampling (NCIS). A common practice in the literature (e.g. see experiments in [24]) is to use the *normalised capped importance sampling* estimator that is defined as:

$$\hat{\mathcal{R}}^{\text{NCIS}}(\pi_t, c) = \frac{\frac{1}{n} \sum_{(x, a, r) \in \mathcal{S}_n} \bar{w}(a, x) r}{\frac{1}{n} \sum_{(x, a, r) \in \mathcal{S}_n} \bar{w}(a, x)} \quad (1)$$

It involves in re-adjusting the expected reward proportionally to the probability mass capped. In the following, we show how this estimator models the bias introduced by capping. Asymptotically, we

can compute the bias of this estimator (the proof is in the appendix):

$$\mathbb{P} \left(\lim_{n \rightarrow \infty} \hat{\mathcal{R}}^{\text{NCIS}}(\pi_t, c, \mathcal{S}_n) = \frac{\mathbb{E}_{\pi_t} \left[\frac{\bar{W}R}{W} \right]}{\mathbb{E}_{\pi_t} \left[\frac{\bar{W}}{W} \right]} \right) = 1$$

with

$$\mathcal{R}^{\text{NCIS}}(\pi_t, c) \triangleq \frac{\mathbb{E}_{\pi_t} \left[\frac{\bar{W}R}{W} \right]}{\mathbb{E}_{\pi_t} \left[\frac{\bar{W}}{W} \right]} = \mathcal{R}^{\text{CIS}}(\pi_t, c) + \mathcal{R}^{\text{CIS}}(\pi_t, c) \frac{1 - \mathbb{E}_{\pi_t} \left[\frac{\bar{W}}{W} \right]}{\mathbb{E}_{\pi_t} \left[\frac{\bar{W}}{W} \right]} \quad (2)$$

Intuitively, if we reuse the expression of $\mathcal{B}^{\text{CIS}}(\pi_t, c)$, we see that NCIS, instead of setting $\mathcal{B}^{\text{CIS}}(\pi_t, c)$ to zero like CIS, approximates the performance on the capped volume by the performance on the non-capped volume.

In the case of zero capping, the approximation made by NCIS is even more intuitive: NCIS makes the assumption that overall,

$$\mathbb{E}_{\pi_t} [R|W > c] \approx \mathbb{E}_{\pi_t} [R|W < c] .$$

and approximates $\mathcal{B}^{\text{CIS}}(\pi_t, c)$ as:

$$\mathcal{B}^{\text{CIS}}[\pi_t, c] \approx \mathbb{E}_{\pi_t} [R|W < c] \mathbb{P}_{\pi_t}(W > c)$$

Those expectations are both on the action A and on the context X .

This approximation is exact at least in the trivial case when the reward R is independent from both the context X and the action A , and thus of the weight W . We can expect the approximation to be reasonable when the noise level is high because in this case the dependency of R on X and A is low. However we shall see in the next section why this assumption may be poor in practice.

5.2 Need for a local bias modelling

The NCIS estimator compensates the capping with a proportional rescaling. However, this rescaling is performed globally, while the underlying data may contain many sub-groups with different average rewards. For instance, the MovieLens dataset [9] exhibits a small group of frequent user and a large majority of occasional users. In the context of e-commerce, [12] reports that Prime members spend in average 4.6 times more than non-prime members on Amazon™. When operating with such different groups of customers, it is quite common to introduce changes in the recommender system that do not equally affect the different groups. For instance, one could decide to favour good deal products for registered customers, which impact both groups differently. As the new policy impact differs from registered to non-registered customers, the capping may be stronger for one of the groups (formalised by the quantity \bar{W}/W), thus a blind global rescaling introduces a large bias (see Table 1 for a toy counter-example).

More formally, from (2), we can directly deduce that the asymptotic bias of NCIS can be written as

$$\mathcal{B}^{\text{NCIS}}(\pi_t, c) \triangleq \mathbb{E}_{\pi_t} [R] - \mathcal{R}^{\text{NCIS}}(\pi_t, c) = - \text{Cov}_{\pi_t} \left(R, \frac{\bar{W}}{W} \right) / \mathbb{E}_{\pi_t} \left[\frac{\bar{W}}{W} \right] ,$$

which makes NCIS consistent when W and R are independent. However, as we just discussed, the reward R and the capping \bar{W}/W may be correlated through a third confounding variable such as the

Table 1: Counter-example for NCIS. The test policy π_t proposes more good deal products to registered customers, improving the (already good) performance on them. The effect is neutral on unknown (low performing) customers. For π_t , NCIS estimates a reward of 1.8 while the (true) expected reward is 2.1. As the reward of π_p is 1.9, we would wrongly conclude that π_t is worse than π_p .

	registered customers	unknown customers
Proportions	0.1	0.9
Performance of π_p	10	1
Performance of π_t	12	1
$\mathbb{E}(\bar{W}/W)$	0.7	1

type of customer. This information is contained in the context X , which leads us to decompose the bias conditionally to X ,

$$\mathcal{B}^{\text{NCIS}}(\pi_t, c) = -\frac{\text{Cov}_{\pi_t} \left(\mathbb{E}[R|X], \mathbb{E} \left[\frac{\bar{W}}{W} | X \right] \right) + \mathbb{E}_{\pi_t} \left[\text{Cov} \left(R, \frac{\bar{W}}{W} | X \right) \right]}{\mathbb{E}_{\pi_t} \left[\frac{\bar{W}}{W} \right]}$$

To improve over NCIS, we can assume the first term of the numerator to be dominant. Indeed, in practice the recommendation (the action A) itself is usually an unsolicited attempt to influence a user with a pre-existing intent (the context X). Thus, one can realistically expect the reward R to be more correlated to the initial intent than to the recommendation. Following this idea, we propose next several ways to build "local" versions of the NCIS estimator – by normalising conditionally to the context – to get rid of the first term of the bias (the biggest) and obtain estimators with a much smaller bias.

5.3 Piecewise constant model

The following estimator is the first step toward a model of the bias at a finer scale. The simplest way to build a local version of NCIS is to use stratification. In other words, to make a piecewise version of NCIS. The goal is to find a partition \mathcal{G} of X in order to use the decomposition of the expectation over this partition,

$$\mathbb{E}_{\pi_t}[R] = \sum_{g \in \mathcal{G}} \mathbb{E}_{\pi_t}[R|X \in g] \mathbb{P}(X \in g)$$

and then estimate the expectation separately on each group of the partition with the NCIS estimator:

$$\hat{\mathcal{R}}^{\text{PieceNCIS}}(\pi_t, c) = \sum_{g \in \mathcal{G}} \alpha_g \hat{\mathcal{R}}_g^{\text{NCIS}}(\pi_t, c)$$

where $\alpha_g = \sum_{(x, a, r) \in \mathcal{S}_n} \mathbf{1}_{x \in g} / n$ estimates $\mathbb{P}(X \in g)$ and $\hat{\mathcal{R}}_g^{\text{NCIS}}(\pi_t, c)$ is the restriction of the NCIS estimator to the group g to estimate $\mathbb{E}_{\pi_t}[R|X \in g]$:

$$\hat{\mathcal{R}}_g^{\text{NCIS}}(\pi_t, c) = \frac{\sum_{(x, a, r) \in \mathcal{S}_n} \mathbf{1}_{x \in g} \bar{w}(a, x) r}{\sum_{(x, a, r) \in \mathcal{S}_n} \mathbf{1}_{x \in g} \bar{w}(a, x)}$$

A desirable constraint on \mathcal{G} is to be independent from the tested policy π_t . When removing the capping – i.e. $c \rightarrow \infty$ – the estimator $\hat{\mathcal{R}}^{\text{PieceNCIS}}(\pi_t, c)$ is consistent.

Apart from this constraint, the partition can be constructed from any hand-crafted splits on features of x . However, even though easy in practice, this option is not satisfying because the performance of the estimator will strongly depend on a manual choice of the partition. A more agnostic method to build an empirically "good" partition, would be to learn a value function $V(x)$ to predict the expected reward given a context x and build the partition based on the output of this model. For instance, in the experiments presented in Section 6, we use a regular partition of the output of the model in the log-space (base b):

$$\mathcal{G} = \left\{ V^{-1}(I_k) : I_k = [b^k, b^{k+1}], k \in \mathbb{Z} \right\}$$

This approach provides two advantages: 1) given a group, the reward does not depend strongly on x anymore, which was the first objective of stratification, 2) the size of the partition is more controlled than with a hand-crafted one, leading to more samples per group and thus less estimation problems. However, it comes at the cost of needing to fit a value function on a separate set of data.

5.4 Pointwise model

To avoid having to learn and design a value model to perform a stratification, we push the idea further and use the decomposition:

$$\mathbb{E}_{\pi_t}[R] = \sum_{x \in \mathcal{X}} \mathbb{E}_{\pi_t}[R|X = x] \mathbb{P}(X = x)$$

However, building an estimator of $\mathbb{E}_{\pi_t}[R|X = x]$ becomes more challenging. Following the underlying idea of NCIS, we want to make the following approximation:

$$\mathbb{E}_{\pi_t}[R|X = x] \approx \frac{\mathbb{E}_{\pi_t} \left[R \frac{\bar{W}}{W} | X = x \right]}{\mathbb{E}_{\pi_t} \left[\frac{\bar{W}}{W} | X = x \right]}$$

Unfortunately, when conditioning on a value of x , we cannot use a simple ratio estimator such as

$$\frac{\sum_{(x', a, r) \in \mathcal{S}_n} \mathbf{1}_{x' = x} \bar{w}(a, x') r}{\sum_{(x', a, r) \in \mathcal{S}_n} \mathbf{1}_{x' = x} \bar{w}(a, x')}$$

Indeed, the number of samples in the training set exactly matching a given value of x is very low (it can even be 0 if x is continuous), so the bias of the ratio estimator is not negligible anymore. Fortunately, when conditioned on x , we can be much better at the estimation of $\mathbb{E}_{\pi_t} \left[\frac{\bar{W}}{W} | X = x \right]$. In fact, we could even compute it exactly by a simple marginalisation on the actions, but the number of actions is too large in our case to perform this computation in a reasonable time. Moreover, we are not aware of any closed-form for it, even when both policies follow simple models (e.g. Plackett-Luce or Mallows).

However, we can efficiently sample from the policy π_t , and therefore compute a Monte Carlo approximation of this probability. Since we actually want an estimator of the ratio $1/\mathbb{E}_{\pi_t} \left[\frac{\bar{W}}{W} | X = x \right]$, we can use a rejection sampling technique such as Lahiri's [14] or Midzuno-Sen method [18, 23] to get an unbiased estimate denoted $\hat{I}P_c(x)$. In practice to build an estimator of $1/\mathbb{E}_{\pi_t} \left[\frac{\bar{W}}{W} | X = x \right]$, we use the Midzuno-Sen method. We define a random variable U uniformly distributed between 0 and 1. Then, we do successively

- sample u from the uniform distribution and w_1 from π_t until reaching $w_1 < u$
- sample w_2, \dots, w_n from π_t
- return $\frac{n}{\frac{w_1}{w_1 + \dots + \frac{w_n}{w_n}}}$

Through this method, the expectation of $n/(W_1/\bar{W}_1 + \dots + W_n/\bar{W}_n)$ is equal to $1/\mathbb{E}_{\pi_t} \left[\frac{\bar{W}}{W} \mid X = x \right]$. In the following, we will note it $\hat{I}P_c(x)$.

Finally, we can define the following estimator:

$$\hat{\mathcal{R}}^{\text{PointNCIS}}(\pi_t, c) = \frac{1}{n} \sum_{(x, a, r) \in \mathcal{S}_n} \hat{I}P_c(x) \bar{w}(a, x) r$$

We need to notice one pitfall of this method: if the test distribution and the prod distribution are really dissimilar, the expectation $\mathbb{E}_{\pi_t} \left[\frac{\bar{W}}{W} \mid X = x \right]$ may become very low. It means that the effective weight by which we multiply the reward r , $\frac{w}{\mathbb{E}_{\pi_t} \left[\frac{\bar{W}}{W} \mid X = x \right]}$ might be much bigger than the capping value, and the estimator would have a high variance. We can overcome this problem when using max capping by decreasing the capping value when the distributions do not overlap enough (see appendix for justification).

Table 2: Summary table of the different estimators. First column sum up the formulae of the estimators, second one the approximation $\hat{\mathcal{B}}$ of the bias term \mathcal{B} in the general case and the case of zero capping

	$\hat{\mathcal{R}}(\pi_t, c)$	approx $\hat{\mathcal{B}}^{\text{CIS}}(\pi_t, c, x)$
CIS	$\frac{1}{n} \sum_{\mathcal{S}_n} r \bar{w}(a, x)$	0
NCIS	$\frac{\sum_{\mathcal{S}_n} r \bar{w}(a, x)}{\sum_{\mathcal{S}_n} \bar{w}(a, x)}$	$\mathbb{E}_{\pi_t} \left[\frac{R\bar{W}}{W} \right] \frac{1 - \mathbb{E}_{\pi_t} \left[\frac{\bar{W}}{W} \right]}{\mathbb{E}_{\pi_t} \left[\frac{\bar{W}}{W} \right]}$
PieceNCIS	$\sum_{g \in \mathcal{G}} \alpha_g \hat{\mathcal{R}}_g^{\text{NCIS}}(\pi_t, c)$	$\mathbb{E}_{\pi_t} \left[\frac{R\bar{W}}{W} \mid X \in g \right] \frac{1 - \mathbb{E}_{\pi_t} \left[\frac{\bar{W}}{W} \mid X \in g \right]}{\mathbb{E}_{\pi_t} \left[\frac{\bar{W}}{W} \mid X \in g \right]}$
PointNCIS	$\frac{1}{n} \sum_{\mathcal{S}_n} \hat{I}P_c(x) \bar{w}(a, x) r$	$\mathbb{E}_{\pi_t} \left[\frac{R\bar{W}}{W} \mid X = x \right] \frac{1 - \mathbb{E}_{\pi_t} \left[\frac{\bar{W}}{W} \mid X = x \right]}{\mathbb{E}_{\pi_t} \left[\frac{\bar{W}}{W} \mid X = x \right]}$

6 EXPERIMENTS

We use the A/B test history of a commercial recommender system to compare the uplifts $\Delta \hat{\mathcal{R}}$ estimated by our *offline A/B tests* methods with the ground truth ΔR estimated during the *online A/B tests*.

6.1 Dataset

We have access to a proprietary dataset of 39 *online A/B tests*, representing a total of few hundreds of billions of recommendations. We consider a click-based business metric. Since clicks are relatively rare, the reward signal has a high variance, even conditioned to the context and to the action. For each test, we consider:

- the two policies involved in the A/B test π_p and π_t ,
- for each display (units x whose features represent the context of the display and past interactions with the user), we have access to,

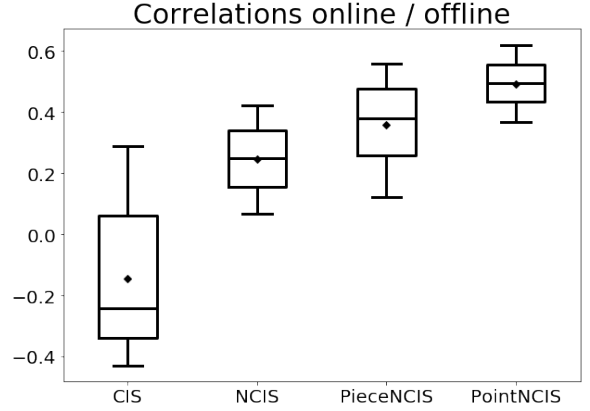


Figure 3: Correlation between online and offline uplifts. Confidence bounds are obtained using bootstraps. Whiskers are 10% and 90% quantiles and the boxes represent quartiles.

- the top- K ranking (action a) chosen along with its probability under the logging policy $-\pi_p(a|x)$ on control population A or $\pi_t(a|x)$ on test population B,
- the set of eligible items,
- the observed reward r .

On each A/B test, we can compute the online estimate of the uplift $\Delta \hat{\mathcal{R}}(\pi_p, \pi_t) = \hat{\mathcal{R}}(\mathcal{S}_t) - \hat{\mathcal{R}}(\mathcal{S}_p)$ – our ground truth – and for any estimator, the offline estimate of the uplift $\hat{\mathcal{R}}^{\text{EST}}(\pi_t, c, \mathcal{S}_p) - \hat{\mathcal{R}}(\mathcal{S}_p)$. We added \mathcal{S}_p in the arguments of the estimator to emphasize the estimator is computed on data collected by the *prod* population.

We could run several *offline A/B tests* based on the data on a single *online A/B test* – e.g. $\Delta \hat{\mathcal{R}}(\pi_p, \pi_t)$ on the data of control population A or $\Delta \hat{\mathcal{R}}(\pi_t, \pi_p)$ on the data of test population B. We only keep one of them, because if we compare them with the same *online A/B test* result, the comparisons wouldn't be independent. In the following, we consider $\Delta \hat{\mathcal{R}}(\pi_p, \pi_t)$ on the data of control population A, which means the control (logging) policy is the production one π_p and the tested one is π_t .

6.2 Estimators

We compare here four estimators presented in the previous sections: CIS, NCIS, PieceNCIS and PointNCIS. We set the capping value to $c = 100$ based on the graph presented in Section 4.3. We discarded non-capped estimators such as IS or NIS due to their very high variance: the confidence intervals on $\Delta \hat{\mathcal{R}}$ would never lead to a positive or negative decision, it would always be neutral. Moreover, we chose to ignore the doubly-robust estimator (DR) for two reasons. Indeed if the estimator is not capped, it suffers the same issue as IS and NIS. When it is capped [25], its performance strongly depends on the reward model such that the optimal policy under such estimator when $c \rightarrow 0$ is the deterministic policy choosing the argmax action on the reward model.

Computation time. The NCIS and PieceNCIS estimators need to be computed on the entire dataset, on both positive and negative

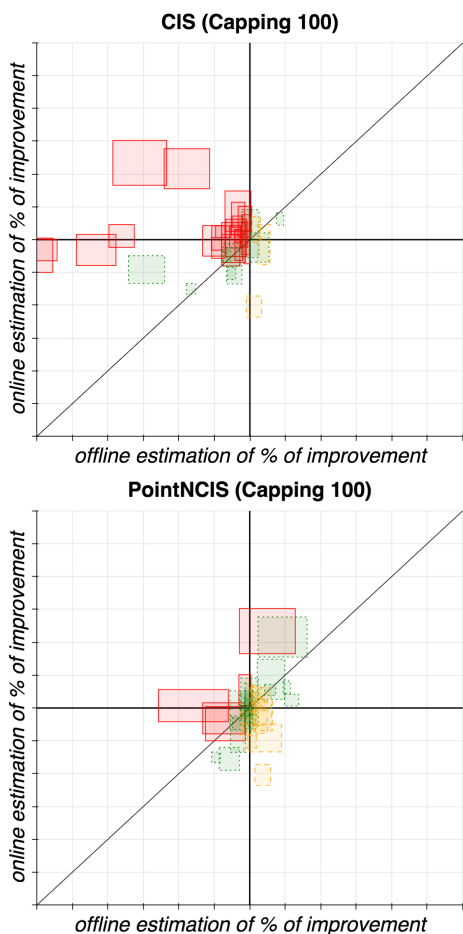


Figure 4: Comparison of online/offline decision. A box is an A/B test. The width (resp. height) of the box is a 90% confidence interval on the offline (resp. online) uplift. The scale is the same for both axis. Green/dotted: right decision. Orange/dashed: false positive). Red/plain: false negative.

examples. Like CIS, PointNCIS just needs to read the positive examples : it leads to a huge gain in computation time and efficiency when the reward is very sparse.

6.3 Correlation in Online / Offline Uplifts

To compare the performance of the different estimators, we first simply compute the correlations between the values of $\Delta\hat{R}$ and ΔR on the series of A/B tests. Results are presented in Fig. 3. As expected, CIS performs quite poorly, due to the strong capping. Compensating the bias only globally with NCIS already proves to be a good improvement. Then, the more local the model on the bias, the better performing the estimator: the piecewise estimator is better than the global and the pointwise is better than the piecewise.

While these first results are already satisfying, it does not reflect the fact that all types of errors are not as bad and that the different estimators do not lead to the same types of errors – which, as we

will see in the following, explains why the correlation online/offline of CIS seems to be negative.

6.4 False positive VS False negative rate

From the point of view of continuously improving a production recommender system, false negatives are much worse than false positives. A false positive – predicted positive by the offline estimator and actually negative or neutral during the A/B test – has a cost limited to the A/B test duration. It slows down the pace of improvement of the system. On the contrary, a false negative is an actual improvement of the system that will never be tested online due to a mistake in the offline estimation. Such an error has a long term cost, as it is an actual improvement that won't be rolled-out.

In Figure 4, we show the comparison between online and offline decisions for our two extreme estimators CIS and PointNCIS. The number of false negatives of CIS is especially interesting: it reflects that CIS always underestimates the reward. It explains why the correlation of CIS is negative in Fig. 3, just because there are more positive decisions than negative in our dataset.

To have a clearer understanding of the quality of the different estimators, we sum up in Table 3 the uplift correlations (same data as in Fig. 3) along with metrics on the decision (positive / neutral / negative) such as the precision and the false negative rate (FNR). We can split the improvement in three. First, only using a global model of the bias – such as NCIS – is enough to drastically reduce the number of false negative errors: the FNR goes from 0.64 (CIS) to 0.33 (NCIS). This is also reflected by an improvement in precision and correlation. Then, another improvement comes from using more local approximations of the bias: precision and correlation also improves from NCIS to PieceNCIS or PointNCIS. When looking more carefully at the changes of decisions between the estimator, we actually noticed several A/B tests that include changes similar to the counter-example presented in Sec. 5.2, which support the need for more local estimators and explain the improvement the more local estimators show. Finally, we can also observe a better correlation of PointNCIS over PieceNCIS. However, as the precision is quite similar, it may only come from a better estimation of the value of the uplift and not from more aligned decisions. In the end, its performance and computation efficiency improvements make PointNCIS a better choice than the other estimators.

Table 3: Performance of the different estimators. Uncertainty is reported based on 10%/90% confidence intervals obtained by bootstrapping. CI size indicates the average relative width of the confidence interval compared to CIS one.

	Correlation	Precision	FNR	CI size
CIS	-0.15 ± 0.35	0.28 ± 0.10	0.64 ± 0.11	–
NCIS	0.24 ± 0.18	0.47 ± 0.11	0.33 ± 0.11	1.1
PieceNCIS	0.36 ± 0.22	0.53 ± 0.11	0.28 ± 0.08	1.5
PointNCIS	0.49 ± 0.13	0.56 ± 0.13	0.16 ± 0.09	0.8

7 CONCLUSION

Through the paper, we exhibited the different sub-optimality properties of the traditional counterfactual estimators in the setting

of recommender systems. We introduced several new estimators exhibiting a better bias-variance trade-off than the traditional Normalised Capped Importance Sampling estimator. Then, we provided a benchmark of the different offline estimators with experiments conducted on a large commercial recommender system. In the future, we plan to investigate other ways to reduce the variance of the offline estimators. A simple way would be to exploit that several recommendation policies are implemented every week: combining them in order to make the mixture of policies closer to the test policy could help to reduce the variance of the different offline estimators.

A APPENDIX

A.1 Analysis of the bias of the NCIS estimator

In this section, we analyse the bias of the Normalised Capped Importance Sampling estimator in the general case (true for both max and zero capping). We also show how the formula can be simplified in the case of zero capping.

LEMMA A.1 (ASYMPTOTICAL BEHAVIOUR OF $\hat{\mathcal{R}}^{\text{NCIS}}$). *Let $\hat{\mathcal{R}}^{\text{NCIS}}$ the normalised capped importance sampling estimator defined in (1). Then,*

$$\mathbb{P}\left(\lim_{n \rightarrow \infty} \hat{\mathcal{R}}^{\text{NCIS}}(\pi_t, c, \mathcal{S}_n) = \hat{\mathcal{R}}^{\text{CIS}} + \hat{\mathcal{R}}^{\text{CIS}} \frac{1 - \mathbb{E}_{\pi_t} \left[\frac{\bar{W}}{W} \right]}{\mathbb{E}_{\pi_t} \left[\frac{\bar{W}}{W} \right]} \right) = 1$$

PROOF. First, we study the convergence of the numerator: It is the mean of n i.i.d. random variables. Thus, according to the strong law of large numbers, with probability one,

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{(x, a, r) \in \mathcal{S}_n} \bar{w}(a, x) r = \mathbb{E}_{\pi_p} [\bar{W}R] = \mathbb{E}_{\pi_t} \left[\frac{\bar{W}R}{W} \right]$$

The denominator is also the mean of n i.i.d. random variables. Then, with probability one,

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{(x, a, r) \in \mathcal{S}_n} \bar{w}(a, x) = \mathbb{E}_{\pi_p} [\bar{W}] = \mathbb{E}_{\pi_t} \left[\frac{\bar{W}}{W} \right]$$

Hence, we reach

$$\begin{aligned} \lim_{n \rightarrow \infty} \hat{\mathcal{R}}^{\text{NCIS}} &= \frac{\mathbb{E}_{\pi_t} \left[\frac{\bar{W}R}{W} \right]}{\mathbb{E}_{\pi_t} \left[\frac{\bar{W}}{W} \right]} = \mathbb{E}_{\pi_t} \left[\frac{\bar{W}R}{W} \right] + \mathbb{E}_{\pi_t} \left[\frac{\bar{W}R}{W} \right] \frac{1 - \mathbb{E}_{\pi_t} \left[\frac{\bar{W}}{W} \right]}{\mathbb{E}_{\pi_t} \left[\frac{\bar{W}}{W} \right]} \\ &= \hat{\mathcal{R}}^{\text{CIS}} + \hat{\mathcal{R}}^{\text{CIS}} \frac{1 - \mathbb{E}_{\pi_t} \left[\frac{\bar{W}}{W} \right]}{\mathbb{E}_{\pi_t} \left[\frac{\bar{W}}{W} \right]} \end{aligned}$$

□

We observe that NCIS is correcting CIS by approximating the performance on the capped volume by the performance on the non-capped volume. It helps to reduce the bias of CIS. We study now the particular case of zero capping.

LEMMA A.2 (ASYMPTOTICAL BEHAVIOUR OF $\hat{\mathcal{R}}_{\text{zero}}^{\text{NCIS}}$). *Let $\hat{\mathcal{R}}_{\text{zero}}^{\text{NCIS}}$ the capped normalised importance sampling estimator. Then,*

$$\mathbb{P}\left(\lim_{n \rightarrow \infty} \hat{\mathcal{R}}_{\text{zero}}^{\text{NCIS}}(\pi_t, c, \mathcal{S}_n) = \mathbb{E}_{\pi_t} [RI_{W \leq c}] \right) = 1$$

PROOF. Straightforward application of the previous lemma □

This analysis can obviously be extended to PieceNCIS.

A.2 Analysis under varying capping parameter

To prove that we can control the variance of PointNCIS even though the value of $\tilde{w}_{\tilde{c}}(a, x) = \frac{\bar{w}(a, x)}{\mathbb{E}_{\pi_t} \left[\frac{\bar{W}}{W} | X=x \right]}$ may be higher than the capping c , we need to prove that for any value of $c > 1$, we can find a \tilde{c} such that $\tilde{w}_{\tilde{c}}(a, x) \leq c$. The following lemma states that it is possible to ensure this with max-capping.

LEMMA A.3 (MAX-CAPPING). *For any a and $c > 1$, there exists \tilde{c} such that $\tilde{w}_{\tilde{c}}(a, x) \leq c$.*

PROOF. For any \tilde{c} , $\mathbb{E}_{\pi_t} \left[\frac{\bar{W}}{W} | x \right] = \mathbb{E}_{\pi_p} [\min(W, \tilde{c}) | x]$, thus

$$\tilde{w}_{\tilde{c}}(a, x) \leq \frac{\tilde{c}}{\mathbb{E}_{\pi_p} [\min(W, \tilde{c}) | x]} \leq \frac{1}{\mathbb{P}_{\pi_p}(W \geq \tilde{c} | x)} \xrightarrow{\tilde{c} \rightarrow 0} 1$$

as $\mathbb{E}_{\pi_p} [\min(W, \tilde{c}) | x] = \tilde{c} \mathbb{P}_{\pi_p}(W \geq \tilde{c} | x) + \mathbb{E}_{\pi_p} [W | W < \tilde{c} | x]$. □

Unfortunately, we cannot ensure such property for zero-capping, which prevents us from adapting \tilde{c} for PointNCIS.

LEMMA A.4 (ZERO-CAPPING). *There exists $a, c > 1$ such that for any $\tilde{c} > 0$, $\tilde{w}_{\tilde{c}}(a, x) > c$.*

PROOF. We use a counter-example where two actions a_0 and a_1 are taken with probabilities $\pi_p(a_0) = p$ and $\pi_t(a_0) = 1 - \pi_p(a_0)$ where $p > 0.5$. Then, $w(a_0) = \frac{1-p}{p} < w(a_1) = \frac{p}{1-p}$.

Case $\tilde{c} < w(a_0)$: $\tilde{w}_{\tilde{c}}(a, x)$ is undefined.

Case $\tilde{c} \in [w(a_0), w(a_1)]$: $\tilde{w}_{\tilde{c}}(a_1, x) = 0$ and $\tilde{w}_{\tilde{c}}(a_0, x) = \frac{1-p}{p^2}$.

Case $\tilde{c} \geq w(a_1)$: $\tilde{w}_{\tilde{c}}(a_1, x) = \frac{p}{1-p} > 1$ and $\tilde{w}_{\tilde{c}}(a_0, x) = \frac{1-p}{p}$.

So, if $0.5 < p < \frac{-1+\sqrt{5}}{2}$ and $1 < c < \min\left(\frac{p}{1-p}, \frac{1-p}{p^2}\right)$ then there exists an action in each case such that $\tilde{w}_{\tilde{c}} > c$. □

A.3 Additional figures

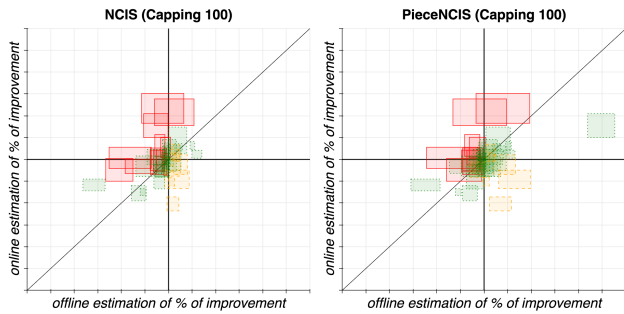


Figure 5: Comparison of online/offline decision. A box is an A/B test. The width (resp. height) of a box is a 90% confidence bound on the offline (resp. online) uplift. The scale is the same for both axis. Green/dotted: right decision. Orange/dashed: false positive. Red/plain: false negative.

REFERENCES

- [1] Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. 1999. *Modern information retrieval*.
- [2] Léon Bottou and Jonas Peters. 2013. Counterfactual reasoning and learning systems: the example of computational advertising. *Proceedings of Journal of Machine Learning Research (JMLR)*.
- [3] Ralph Allan Bradley and Milton E. Terry. 1952. Rank Analysis of Incomplete Block Designs: The Method of Paired Comparisons. *Biometrika* (1952).
- [4] Weiwei Cheng, Eyke Hüllermeier, and Krzysztof J Dembczynski. 2010. Label ranking methods based on the Plackett-Luce model. *Proceedings of the 27th International Conference on Machine Learning (ICML)*.
- [5] Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. 2008. An Experimental Comparison of Click Position-bias Models. *Proceedings of the International Conference on Web Search and Data Mining (WSDM)*.
- [6] Miroslav Dudík, John Langford, and Lihong Li. 2011. Doubly robust policy evaluation and learning. *Proceedings of the 28th International Conference on Machine Learning (ICML)*.
- [7] John Guiver and Edward Snelson. 2009. Bayesian inference for Plackett-Luce ranking models. *Proceedings of the 26th annual International Conference on Machine Learning (ICML)*.
- [8] JM Hammersley and DC Handscomb. 1964. *Monte Carlo Methods*. Chapter.
- [9] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.*
- [10] Jonathan L Herlocker, Joseph A Konstan, Loren G Terveen, and John T Riedl. 2004. Evaluating collaborative filtering recommender systems. *Proceedings of Transactions on Information Systems (TOIS)*.
- [11] Daniel G Horvitz and Donovan J Thompson. 1952. A generalization of sampling without replacement from a finite universe. *Journal of the American statistical Association*.
- [12] Business Insider. 2017. Morgan Stanley puts Amazon Prime subscribers at 65M. <http://www.businessinsider.fr/us/morgan-stanley-puts-amazon-prime-subscribers-at-65m-2017-2/>. (2017).
- [13] Kalervo Järvelin and Jaana Kekäläinen. 2000. IR evaluation methods for retrieving highly relevant documents. *Proceedings of the 23rd annual international conference on Research and development in information retrieval (SIGIR)*.
- [14] Deba B Lahiri. 1951. A method of sample selection providing unbiased ratio estimates. *Bulletin of the International Statistical Institute*.
- [15] Lihong Li, Wei Chu, John Langford, and Xuanhui Wang. 2011. Unbiased off-line evaluation of contextual-bandit-based news article recommendation algorithms. *Proceedings of the International Conference on Web Search and Data Mining (WSDM)*.
- [16] Benjamin M. Marlin, Richard S. Zemel, Sam Roweis, and Malcolm Slaney. 2007. Collaborative Filtering and the Missing at Random Assumption. *Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence (UAI)*.
- [17] Andreas Maurer and Massimiliano Pontil. 2009. Empirical Bernstein bounds and sample variance penalization. *Proceedings of the 22nd Annual Conference on Learning Theory (COLT)*.
- [18] Hiroshi Midzuno. 1951. On the sampling system with probability proportionate to sum of sizes. *Annals of the Institute of Statistical Mathematics*.
- [19] Art Owen. 2010. Monte Carlo theory, methods and examples. (2010). arXiv:arXiv:1012.5461v2
- [20] R. L. Plackett. 1975. The Analysis of Permutations. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*.
- [21] MJD Powell and J Swann. 1966. Weighted uniform sampling: a Monte Carlo technique for reducing variance. *Journal of Applied Mathematics*.
- [22] Bruno Pradel, Nicolas Usunier, and Patrick Gallinari. 2012. Ranking with non-random missing ratings: influence of popularity and positivity on evaluation metrics. *Proceedings of the Sixth Conference on Recommender Systems (RecSys)*.
- [23] Amode Ranjan Sen. 1952. Present status of probability sampling and its use in estimation of farm characteristics. *Econometrica* (1952).
- [24] Adith Swaminathan and Thorsten Joachims. 2015. The Self-Normalized Estimator for Counterfactual Learning. *Proceeding of Neural Information Processing Systems (NIPS)*.
- [25] Philip Thomas and Emma Brunskill. 2016. Data-efficient off-policy policy evaluation for reinforcement learning. *Proceedings of the International Conference on Machine Learning (ICML)*.
- [26] Louis Leon Thurstone. 1927. A Law of Comparative Judgement. *Psychological Review*.
- [27] Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Journal of Machine Learning Research (JMLR)* (1992).
- [28] John I. Yellott. 1977. The relationship between Luce's Choice Axiom, Thurstone's Theory of Comparative Judgment, and the double exponential distribution. *Journal of Mathematical Psychology*.