Offline Contextual Bandit with Counterfactual Sample Identification

Alexandre Gilotte
a.gilotte@criteo.com
Criteo AI Lab
Paris, France

Otmane Sakhi
o.sakhi@criteo.com
Criteo AI Lab
Paris, France

Imad Aouali
i.aouali@criteo.com
Criteo AI Lab and CREST-ENSAE
Paris, France

Benjamin Heymann
b.heymann@criteo.com
Criteo AI Lab, Fairplay joint team
Paris, France

ABSTRACT

In production systems, contextual bandit approaches often rely on direct reward models that take both action and context as input. However, these models can suffer from confounding, making it difficult to isolate the effect of the action from that of the context. We present *Counterfactual Sample Identification*, a new approach that re-frames the problem: rather than predicting reward, it learns to recognize which action led to a successful (binary) outcome by comparing it to a counterfactual action sampled from the logging policy under the same context. The method is theoretically grounded and consistently outperforms direct models in both synthetic experiments and real-world deployments.

KEYWORDS

Offline contextual bandit, off-policy learning, confounding variables

ACM Reference Format:
Alexandre Gilotte, Otmane Sakhi, Imad Aouali, and Benjamin Heymann. 2025. Offline Contextual Bandit with Counterfactual Sample Identification. In *Proceedings of Recsys '25: CONSEQUENCES Workshop*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Contextual Bandits [12] serve as an important intermediate framework between multi-armed bandits and full reinforcement learning (RL) [21]. Like RL, they enable decision-making based on rich, high-dimensional state (or context) information. However, they simplify the problem by assuming that contexts are independent and identically distributed (*i.i.d.*). Because many real life systems, such as some recommender systems [1], are well approximated by contextual bandits, there is value in designing practical algorithms for the Contextual Bandit model. A commonly used framework is the **offline Contextual Bandit** [22], in which a dataset has been collected by an initial policy, and the goal is to learn from this dataset a policy with the best expected reward.

IPS based methods. The best performing methods for offline contextual bandits are usually building an estimator of the expected reward of a policy by *Inverse Propensity Scoring* (IPS) [5, 6, 9], and searching for a parametrized policy directly maximizing this criteria. While simple, IPS implementations may suffer from high variance, many improvements have been proposed to make it more stable [2, 4, 10, 11, 13, 14, 16, 17, 19, 20, 22, 23].

Direct Method. In practice, a simple, commonly used algorithm for contextual bandit consists in fitting a model of the expected reward, as a function of the context x and the chosen action a. This is done by applying a supervised learning algorithm on the available dataset [3, 18]. Then, from this model, a greedy policy is built by returning the action which maximize the model-estimated expected value. This method is sometimes referred as *Direct Method* (DM), or *Q-learning* in the RL settings [21]. While it seems that a well-tuned modern IPS algorithm would usually get better results, the Direct Method seems to be still widely used in practical settings. There are several reasons for this popularity: (a) It is relatively simpler, only requiring a "classical" supervised learning algorithm. (b) It might be more stable than IPS, or at least some IPS variants, because of the potentially high variance of the IPS estimator. (c) It may be useful to tune the level of exploration, required for learning the next iterations of the policy, independently from the learning of the policy. While ϵ-greedy exploration is one approach, practitioners often favor increased exploration of actions that the direct model estimates as near-optimal. However, since IPS methods directly output policies, it makes it less straightforward to tune this level of exploration independently. (d) Contextual bandits are only approximations of real systems, and residual sequence effects (e.g. the action of recommending an item to a user might slightly change the state and reward later in the sequence) make the *i.i.d.* assumption not strictly true. Even in the case when these effects are small, they break the unbiasedness assumption which was a big selling point from IPS. It is unclear how IPS - or other methods - perform in such cases.[1]

Recsys '25: CONSEQUENCES Workshop, September, 2025, Prague

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

[1] we believe that further research would be useful here. The same question applies to the method we propose in this paper.

When the Direct Method may fail to produce a good policy. One typical case when DM may dramatically fail is when a variable which explained the action chosen in the past data is missing from the direct reward model; possibly leading to an instance of Simpson's paradox [15]. While in practice the past policy is usually another instance of the same model, there are still frequent cases when this may happen, e.g. if an A/B test adds or removes some features from the model. Another issue is that even with all relevant features available to the model, it may be non trivial to entangle the causal effect of an action from the effect of a context. This is especially the case because actions tend to be strongly correlated to contexts: both because some actions might be available only in some contexts, and because the past policy correlates actions and contexts. Because of these correlations, regularization methods typically used on supervised learning may introduce some kind of confounding in the final model [8]. These issues can be made more acute on systems in which the context is much more predictive of the reward than the action. This is typical in online advertising recommender systems, where the probability of a click on an ad may vary by 2 orders of magnitude with the inventory and user state, while between a "good" and "average" recommendation the same probability of click would typically change by a factor less than 2.

Ranking models. In recommendation settings, where several products are displayed together in the same context, another popular variant is the use of a *ranking loss* explicitly comparing the reward of products from the same context. This avoids modeling the effect of the context, and usually outperform *pointwise* DM.

Contribution. In this paper we propose a new method to learn a policy from offline data **which is halfway between IPS and DM**, Counterfactual sample identification (CSI). Like IPS, it requires the knowledge of the logging policy, and uses it to lessen the effect of confounding from context features. But like DM, it only fits a supervised model with regular supervised learning methods. It can be interpreted as an adaptation of ranking models to the pure contextual bandit setting, while also adapting the idea of Retrospective Estimation [7] to a non-binary action space.

2 COUNTERFACTUAL SAMPLE IDENTIFICATION (CSI)

2.1 Notations and context

The system receives a *context* X, sampled from an unknown distribution. Each context comes with a finite set $\mathcal{A}(x)$ of available *actions*, and the system must select one action a in this set. A reward Y, that depends on x and a, is then received. We restrict to binary rewards, that is, Y is a Bernoulli variable whose parameter depends on x and a.

A policy $\pi(\cdot \mid x)$ is a conditional probability distribution over actions. It maps each context x to a distribution on $\mathcal{A}(x)$. The offline contextual bandit model assumes that we have a dataset made of i.i.d. samples $\mathcal{D} = (x_i, a_i, y_i)_{i\in[n]}$, where actions were sampled from a known policy π_0, and the goal is to choose a policy π_θ in the parametrized family $\{\pi_\theta, \theta \in \Theta\}$ that maximizes the expected reward $\mathbb{E}_x\mathbb{E}_{a\sim\pi_\theta(\cdot|x)}[Y]$.

2.2 Method presentation

We present here 'Counterfactual Sample Identification' (CSI), a method to address the confounding effect from context variables. We first process the dataset as follows: (a) we keep only the positive data point from $\mathcal{D}$; (b) for each positive example $(x, a, y = 1)$, we sample a "counterfactual" action a' from the logging policy $a' \sim \pi_0(\cdot \mid x)$; and produce two examples $(x, a, z = 1)$ and $(x, a, z = 0)$ (c) we combine this two examples $(x_i, a', z = 0)$ and $(x_i, a, z = 1)$ into a new log. Here Z is an auxiliary variable indicating wether the sample contains the true action a or the resample a'. Said otherwise, we *build*:

$$\hat{\mathcal{D}} = \{(x, a', z = 0) : (x, a, 1) \in \mathcal{D}\} \cup \{(x, a, z = 1) : (x, a, 1) \in \mathcal{D}\},$$

where a' is sampled from $\pi_0(\cdot \mid x)$. We then *train* a supervised learning algorithm on $\hat{\mathcal{D}}$ using z as target and (x, a) as features. Let $f(x, a)$ be the resulting model. The greedy CSI policy consists in playing $a^* := \arg\max_{a\in\mathcal{A}(x)} f(x, a)$. [2]

Intuitively, the CSI model learns to recognize if the action is the "true" action which led to a reward $y_i = 1$, or just a resample from π_0 independent from Y. This idea is formalized in Lemma 2.1, which relates the learned probability $f(x, a)$ and the expected reward $\mathbb{P}(Y = 1|, A = a, X = x)$ of action a in context x.

Lemma 2.1. *Let $\sigma : x \to 1/1 + \exp(-x)$ be the sigmoid function. Under a uniform coverage policy π_0, we have the following identity:*

$$\mathbb{P}(Z = 1|A = a, X = x) = \sigma\left(\log\left(\frac{\mathbb{P}(Y = 1|A = a, X = x)}{\mathbb{P}(Y = 1|X = x)}\right)\right).$$

The proof can be found in Appendix A. Note that $\mathbb{P}(Y = 1|X = x)$ depends on the policy π_0, but not on a, this implies that $\mathbb{P}(Z = 1|A = a, X = a)$ orders the actions by their expected reward in context x. The fraction $\frac{\mathbb{P}(Y=1|,A=a,X=x)}{\mathbb{P}(Y=1|X=x)}$ which appears in the log in the equation above can be thought as the *multiplicative advantage* of playing action a in context x - compared to the typical outcome when following π_0. We therefore have

$$\arg\max_{a\in\mathcal{A}(x)} \underbrace{\mathbb{P}(Z = 1|X = x, A = a, Y = 1)}_{\approx f(x,a)} = \arg\max_{a\in\mathcal{A}(x)} \mathbb{P}(Y = 1|X = x, A = a).$$

In a nutshell, instead of modeling the expected reward $\mathbb{P}(Y|x, a)$, our method directly models the multiplicative advantage. It thus avoids the necessity to learn the direct impact of the context on the reward: $P(Z = 1|Y = 1, X = x) = 0.5$ for all x, and thus a feature of x matters in this model only if the relative performances of the actions change with this feature. It is therefore reasonable to expect that CSI performs better on instances where modeling the multiplicative advantage is easier than modeling the reward.

Dependency on π_0. Like the importance-weight based methods, the CSI relies on having a dataset collected from a stochastic policy π_0 that explores well the action space, and it degenerates when this assumption does not hold. To see why, let us assume that for a given context x_i the policy is deterministic. Then with probability 1 both the true action a_i and the resampled action b_i will be identical. There is no point in trying to learn to distinguish them, the model can only learn to output "0.5" for these samples.

[2] Just as in the classical reward model case, we need to adapt it to include some exploration - We do not delve into this topic here as it is not different.

Replacing the sampling of A' by an expectation . The sampling of A' adds a source of noise in the training, which can be avoided: instead of producing one single counterfactual sample $x, a', z = 0$ with a sampled a', we can return one for each possible action a, and weight these samples in the learning by their sampling probability $\pi_0(a' \mid x)$. This does not change the expectation of the loss (we replaced the loss on samples of a' by the expected loss), but reduces the variance; at the cost of a larger training set size. Empirical experiments in Section 3.1 confirm that this improves the quality of the learned policy.

3 EMPIRICAL RESULTS

3.1 On synthetic data

We now compare, in a synthetic environment, several offline learning methods for contextual bandits. Our experiments can be reproduced with a notebook shared in the supplementary material, and the description of the datasets is available in the appendix.

On these synthetic datasets we trained: A direct model of the reward (DM), a CSI model as described in section 2.2 — either sampling the counterfactual action (CSI-sampling) or taking the expectation (CSI-expect)) — and Logarithmic Smoothing, a recent differentiable IPS based method [17].

Table 1 reports the mean normalized reward on 100 environments with different sample size of the collected datasets. With lowest sample counts, DM method performs well. However, as sample counts increase, this method seems to plateau, while the performances of CSI and IPS increase and outperform DM. IPS gives overall the best results, as expected, but with 1M samples CSI-expect is reasonably close. We also note that CSI-expect consistently improves on CSI-sampling, as predicted. Our proposed method is robust and performs well in environments where context effects are important.

Table 1: Click-through rate on synthetic data

Nb Samples	10K	100K	500K
DM	0.76	0.83	0.84
CSI-sampling	0.62	0.82	0.91
CSI-expect	0.71	0.87	0.92
LS-IPS	0.82	0.93	0.96

3.2 On a live production system

Banner design optimization. We tried our method on our production system, selecting the *design* (the size of a grid of products) of ad banners displayed on the web, on large-scale experiments, with several millions positive — *i.e.* clicked — banners per day.

Production baseline. The baseline was a DM (Section 1) using a logistic regression with quadratic kernel.

Exploration. Online, the policy is a mixture of 5% uniform exploration, with a multinomial assigning to each action a probability $\propto \hat{p}(y|x,a)^\alpha$ where α is a temperature parameter.

Learning from counterfactual samples. We applied the (CSI), learning a model $\hat{p}(Z = 1|x, a, y = 1)$ with *the same logistic regression*, features and second order interactions as the previous production model. We only re-tuned the L2-regularization of this logistic.

Experiments with a reduced set of features. For privacy reasons, we wanted to be able to learn a policy with only a subset of the context features which were used in the production model. Directly fitting a reward model with only this subset of features was performing poorly. Using the method proposed here allowed to find a policy with the same subset of features whose performances were only slightly worse than the baseline with all features; **thus proving that the down-lift observed when fitting the reward model with less features was due to confounding effects**.

Experiments with the full set of features. We also noted that the proposed method improved on our production system **by more than** 1% **according to an IPS estimate**. We thus tested it online, keeping the same exploration scheme as the baseline's. To ensure that the new model was able to train efficiently when gathering its own data, we split the users in 4 populations ABCD: A used the reference, trained on all available data, B used a CSI, also trained on all data, C and D used models similar to A and B; with training sets restricted to user data from their own population (so 25% of the data). Table 2 shows the online and offline results. **Online, policies learned from CSI consistently over-performed the baseline by 0.5% to 1%.**

Table 2: Experiments on our large-scale "banner design" dataset. Values indicate the percentage of clicks relative to the reference DM model with all features (100%), i.e., a percentage increase or decrease compared to the reference.

Model	**Clicks** IPS estimate	**Clicks** Online A/B test
DM, all features (Reference)	100%	100%
DM, features subset	94%	[92% - 95%]
CSI, features subset	99%	[98.0% - 99.0%]
CSI, all features	101.5%	[100.4%,100.5%]
DM, all features, 25% users	99.5%	[99.5%;99.7%]
CSI, all features, 25% users	100.4%	[100.0%;100.2%]

Towards IPS learning on this system. Since we used IPS for evaluation, and have low enough variance to use it as an estimator, why don't we directly optimize it? We were able to learn an IPS model which confidently increased the production DM baseline; according to IPS estimator on kept out data. However, when deployed online, the observed performances were not aligned with the expectation from IPS. We argue that it might be due to small sequences effects that were interfering with the training of this model, leading to discrepancies between offline and online performances. Also, even if this hypothesis is true, it is not completely clear why IPS was more affected by these effects than other methods. We thus believe that investigating problems that are "almost-contextual-bandit", and the robustness of different contextual bandit algorithms to such settings, is a promising future research area.

REFERENCES

[1] Imad Aouali, Amine Benhalloum, Martin Bompaire, Achraf Ait Sidi Hammou, Sergey Ivanov, Benjamin Heymann, David Rohde, Otmane Sakhi, Flavian Vasile, and Maxime Vono. 2022. Reward Optimizing Recommendation using Deep Learning and Fast Maximum Inner Product Search. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Washington DC, USA) *(KDD '22)*. Association for Computing Machinery, New York, NY, USA, 4772–4773. https://doi.org/10.1145/3534678.3542622

[2] Imad Aouali, Victor-Emmanuel Brunel, David Rohde, and Anna Korba. 2023. Exponential Smoothing for Off-Policy Learning. In *Proceedings of the 40th International Conference on Machine Learning*. PMLR, 984–1017.

[3] Imad Aouali, Victor-Emmanuel Brunel, David Rohde, and Anna Korba. 2024. Bayesian Off-Policy Evaluation and Learning for Large Action Spaces. *arXiv preprint arXiv:2402.14664* (2024).

[4] Imad Aouali, Victor-Emmanuel Brunel, David Rohde, and Anna Korba. 2024. Unified PAC-Bayesian Study of Pessimism for Offline Policy Learning with Regularized Importance Sampling. In *The 40th Conference on Uncertainty in Artificial Intelligence*. https://openreview.net/forum?id=d7W4H0sTXU

[5] Léon Bottou, Jonas Peters, Joaquin Quiñonero-Candela, Denis X Charles, D Max Chickering, Elon Portugaly, Dipankar Ray, Patrice Simard, and Ed Snelson. 2013. Counterfactual Reasoning and Learning Systems: The Example of Computational Advertising. *Journal of Machine Learning Research* 14, 11 (2013).

[6] Miroslav Dudík, John Langford, and Lihong Li. 2011. Doubly Robust Policy Evaluation and Learning. In *Proceedings of the 28th International Conference on International Conference on Machine Learning* (Bellevue, Washington, USA) *(ICML'11)*. 1097–1104.

[7] Dmitri Goldenberg, Javier Albert, Lucas Bernardi, and Pablo Estevez. 2020. Free Lunch! Retrospective Uplift Modeling for Dynamic Promotions Recommendation within ROI Constraints. In *Fourteenth ACM Conference on Recommender Systems (RecSys '20)*. ACM, 486–491. https://doi.org/10.1145/3383313.3412215

[8] P. Richard Hahn, Carlos M. Carvalho, Jingyu He, and David Puelz. 2016. Regularization and confounding in linear regression for treatment effect estimation. arXiv:1602.02176 [stat.ME] https://arxiv.org/abs/1602.02176

[9] Daniel G Horvitz and Donovan J Thompson. 1952. A generalization of sampling without replacement from a finite universe. *Journal of the American statistical Association* 47, 260 (1952), 663–685.

[10] Olivier Jeunen and Bart Goethals. 2021. Pessimistic reward models for off-policy learning in recommendation. In *Fifteenth ACM Conference on Recommender Systems*. 63–74.

[11] Ilja Kuzborskij, Claire Vernade, Andras Gyorgy, and Csaba Szepesvári. 2021. Confident off-policy evaluation and selection through self-normalized importance weighting. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 640–648.

[12] Tor Lattimore and Csaba Szepesvari. 2019. *Bandit Algorithms*. Cambridge University Press.

[13] Ben London and Ted Sandler. 2019. Bayesian counterfactual risk minimization. In *International Conference on Machine Learning*. PMLR, 4125–4133.

[14] Alberto Maria Metelli, Alessio Russo, and Marcello Restelli. 2021. Subgaussian and differentiable importance sampling for off-policy evaluation and learning. *Advances in Neural Information Processing Systems* 34 (2021), 8119–8132.

[15] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. 2017. *Elements of causal inference: foundations and learning algorithms*. The MIT press.

[16] Otmane Sakhi, Pierre Alquier, and Nicolas Chopin. 2023. PAC-Bayesian Offline Contextual Bandits with Guarantees. In *International Conference on Machine Learning*. PMLR, 29777–29799.

[17] Otmane Sakhi, Imad Aouali, Pierre Alquier, and Nicolas Chopin. 2024. Logarithmic Smoothing for Pessimistic Off-Policy Evaluation, Selection and Learning. In *Advances in Neural Information Processing Systems*, A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (Eds.), Vol. 37. Curran Associates, Inc., 80706–80755. https://proceedings.neurips.cc/paper_files/paper/2024/file/9379ea6ba7a61a402c7750833848b99f-Paper-Conference.pdf

[18] Otmane Sakhi, Stephen Bonner, David Rohde, and Flavian Vasile. 2020. BLOB: A Probabilistic model for recommendation that combines organic and bandit signals. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 783–793.

[19] Yi Su, Maria Dimakopoulou, Akshay Krishnamurthy, and Miroslav Dudík. 2020. Doubly robust off-policy evaluation with shrinkage. In *International Conference on Machine Learning*. PMLR, 9167–9176.

[20] Yi Su, Lequn Wang, Michele Santacatterina, and Thorsten Joachims. 2019. Cab: Continuous adaptive blending for policy evaluation and learning. In *International Conference on Machine Learning*. PMLR, 6005–6014.

[21] Richard Sutton and Andrew Barto. 1998. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA.

[22] Adith Swaminathan and Thorsten Joachims. 2015. Batch learning from logged bandit feedback through counterfactual risk minimization. *The Journal of Machine Learning Research* 16, 1 (2015), 1731–1755.

[23] Yinglun Zhu, Dylan J Foster, John Langford, and Paul Mineiro. 2022. Contextual bandits with large action spaces: Made practical. In *International Conference on Machine Learning*. PMLR, 27428–27453.

A PROOF OF THE LEMMA 2.1

We suppose that for each context x, π_0 covers the possible action space $\mathcal{A}(x)$, that is for each context x:

$$\forall a \in \mathcal{A}(x), \quad \pi_0(a|x) > 0\,.$$

Now, to clarify the definition of random variables, we note: A the action as sampled online, A' the resampled action, and B the action observed in the log example. That is:

$$B := \left(A \text{ if } Z = 1 \text{ else } A'\right).$$

For a context x, and any $a \in \mathcal{A}(x)$, we have:

$$\begin{aligned}\mathbb{P}(Z{=}1|X{=}x, B{=}a, Y{=}1) &= \frac{\mathbb{P}(Z{=}1, B{=}a, Y{=}1|X{=}x)}{\mathbb{P}(Z{=}0, B{=}a, Y{=}1|x) + \mathbb{P}(Z{=}1, B{=}a, Y{=}1|X{=}x)} \\ &= \frac{1}{1 + \frac{\mathbb{P}(Z{=}0, B{=}a, Y{=}1 \mid X{=}x)}{\mathbb{P}(Z{=}1, B{=}a, Y{=}1 \mid X{=}x)}}\end{aligned}$$

By definition of B, when $Z = 1$:

$$\begin{aligned}\mathbb{P}(Z = 1, B = a, Y = 1|X = x) &= \mathbb{P}(Z = 1, A = a, Y = 1|X = x) \\ &= 0.5 \times \mathbb{P}(Y = 1|X = x, A = a)\mathbb{P}(A = a|X = x)\end{aligned}$$

and when Z=0, noting Y ⊥ A':

$$\begin{aligned}\mathbb{P}(Z = 0, B = a, Y = 1|X = x) &= \mathbb{P}(Z = 0, A' = a, Y = 1|X = x) \\ &= 0.5 \times \mathbb{P}(Y = 1|X = x)\mathbb{P}(A' = a|X = x) \\ &= 0.5 \times \mathbb{P}(Y = 1|X = x)\mathbb{P}(A = a|X = x)\end{aligned}$$

Then:

$$\begin{aligned}\mathbb{P}(Z = 1|X = x, B = a, Y = 1) &= \frac{1}{1 + \frac{\mathbb{P}(Y=1|X=x)}{\mathbb{P}(Y=1|X=x, A=a)}} \\ &= \sigma\left(\log\left(\frac{\mathbb{P}(Y = 1|X = x, A = a)}{\mathbb{P}(Y = 1|X = x)}\right)\right)\end{aligned}$$

B DETAILS ON THE EXPERIMENTS

Experiments of Section 3.1 were run on a synthetic dataset where we have an oracle for the exact reward function and distribution of context, allowing exact evaluation of the expected reward of a policy.

Choice of the synthetic environment. We wanted an environment where the effect of context is more important than the effect of the actions, to ensure that either modeling correctly or removing the effect of context is important. Here we note that offline experiments proposed in the literature usually do not share this property: many experiments have been run on contextual bandit made from multi-label dataset, and for each context exactly one (or a few) actions lead to a reward of 1. In such settings, the best possible reward does not depend (much) on the context, we thus did not re-use these benchmarks.

Description if the synthetic environment . We run experiments on a context space $\mathcal{X} := \{0,1\}^7$ of 128 contexts described by 7 binary features; and an action space $\mathcal{A} := \{0,1\}^5$ of 32 actions. For each run of our experiments, we started by sampling a contextual bandit environment defined as: i) a distribution on $\mathcal{X}$ and a reward model on $\mathcal{X}x\mathcal{A}$. These were defined by logistic models with random coefficients. The set of features used here to define the oracle was slightly richer than the set of features used when fitting the models or policy, to introduce some level of model miss-specification. This gives us an environment, and an oracle to estimate the expected reward of any policy.

Dataset generation . From this contextual bandit, we sampled a first dataset from an uniform policy. We then fit a first model (either directly fitting the reward, or using the method of this paper), and collected a second dataset following the 5%-epsilon-greedy policy defined by this model. We then ran different algorithm on this second dataset; and compared the results of their greedy policy. Experiment results in the main section are average on runs where the dataset was collected following a direct model, or a CSI. In practice, we did not observe noticeable differences between the two sets of runs, but wanted to control that using an algo or another here did not change significantly the results.

Model details. Both direct model and CSI in these experiments were fitted with a scikit-learn logistic regression using features from context, action and context-action interactions: $(x, a, x \times a^T)$. The IPS model searched a policy in the same parameter space. For each trained model, we then evaluated the greedy policy inferred from the model; normalized so that the best possible policy in this environment gets a reward of 1 and the worst possible gets 0.