Batch Learning from Bandit Feedback through Bias Corrected Reward Imputation

Lequn Wang [1] **Yiwei Bai** [1] **Arjun Bhalla** [1] **Thorsten Joachims** [1]

Abstract

The problem of batch learning from logged contextual bandit feedback (BLBF) is ubiquitous in recommender systems, search, and online retail. Most previous methods for BLBF have followed a "Model the Bias" approach, estimating the expected reward of a policy using inverse propensity score (IPS) weighting. While unbiased, controlling the variance can be challenging. In contrast, we take a "Model the World" approach using the Direct Method (DM), where we learn a reward-regression model and derive a policy from the estimated rewards. While this approach has not been competitive with IPS weighting for mismatched models due to its bias, we show how directly minimizing the bias of the reward-regression model can lead to highly effective policy learning. In particular, we propose Bias Corrected Reward Imputation (BCRI) and formulate the policy learning problem as bi-level optimization, where the upper level maximizes the DM estimate and the lower level fits a weighted reward-regression. We empirically characterize the effectiveness of BCRI compared to conventional reward-regression baselines and an IPS-based method.

1. Introduction

The logs of interactive systems (e.g., recommender systems, ad placement systems, search engines) are an attractive source of training data, as they provide user-centric feedback that is readily available in large quantities. Such log data takes the form of contextual-bandit feedback, where the system receives a context (e.g., a user profile), takes an action (e.g., recommends a movie) and observes the feedback (e.g., click or not). Different from typical supervised learning where the correct label and a loss function provide full-information feedback, batch learning from contextual bandit feedback (BLBF) deals with partial (only observed for selected action) and biased (by the choice of the policy that logged the data) feedback.

Most previous works on BLBF have taken a "Model the Bias" approach (Strehl et al., 2011; Dudík et al., 2011; Bottou et al., 2013; Swaminathan & Joachims, 2015a;b; Joachims et al., 2018; Kallus, 2018; Su et al., 2019). Inverse propensity score (IPS) weighting techniques are leveraged to model the selection bias in the assignment mechanism, thus providing unbiased estimates of the counterfactual risk (expected reward/loss) throughout a class of policies. This enables learning by optimizing the estimated counterfactual risk, potentially subject to variance regularization (Swaminathan & Joachims, 2015a).

An alternate route to policy learning for BLBF is the "Model the World" approach, where a reward-regression model is learned and then used to derive a policy. However, for real-world problems where models are typically misspecified, the bias of this Direct Method (DM) (Dudík et al., 2011) can be substantial such that the learned policies are often far from optimal. In this paper we address this bias problem and we explore how to directly minimize the bias of DM for misspecified models. In particular, we propose Bias Corrected Reward Imputation (BCRI) as a new method for policy learning in BLBF. The key idea is to optimize a weighted regression estimate of the rewards that minimizes bias with respect to a specific target policy π. We show that the expectation of this weighted reward-regression objective minimizes an upper bound on the mean squared error (MSE) of the DM estimator, primarily by reducing its bias. Reflecting that we need a different regression estimate for each target policy π in our policy space, we formulate the BCRI policy-learning problem as a bi-level optimization problem. In the upper level, it searches through the policy space optimizing the DM estimate of the expected reward. In the lower level, it minimizes an estimated upper bound on the MSE of the upper level objective. We propose a simple procedure for optimizing this bi-level problem.

[1]Department of Computer Science, Cornell University, Ithaca, USA. Correspondence to: Lequn Wang <lw633@cornell.edu>, Yiwei Bai <yb263@cornell.edu>, Arjun Bhalla <ab2383@cornell.edu>, Thorsten Joachims <tj@cs.cornell.edu>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

In an empirical evaluation, we compare BCRI with several other reward-regression objectives for the DM estimator, as well as BanditNet (Joachims et al., 2018) as a representative IPS-based method. Empirical results show that BCRI achieves superior performance against the more naive reward-regression baselines. Compared with BanditNet, BCRI enjoys a large performance gain when the action space is large, which we conjecture is due to a gradient saturation problem of the BanditNet objective (which is non-convex). When the action space is small, BCRI performs comparably to BanditNet but still substantially outperforms the other DM baselines.

2. Batch Learning from Logged Bandit Feedback

In this section, we first formally define the problem of BLBF. Then we introduce several reward-regression baselines, the intuition and theoretical basis for our proposed BCRI approach, and finally the formulation of the BCRI policy learning problem.

2.1. Contextual-Bandit Setting and BLBF

In the contextual-bandit setting, a context $x \in \mathcal{X}$ is drawn i.i.d. from some unknown distribution $P(\mathcal{X})$. The deployed policy $\pi_0(y|x)$ then selects an action $y \in \mathcal{Y}$, and the system receives feedback reward $r \sim D(r|x,y)$ for this particular context-action pair. However, we do not observe feedback for any of the other actions. This results in logged contextual-bandit data from logging policy π_0 of the form

$$\mathcal{S} = \{(x_i, y_i, r_i, \pi_0(\cdot|x_i))\}_{i=1}^n, \tag{1}$$

where $r_i := r(x_i, y_i)$ is the observed reward. The policy-evaluation problem is estimating the expected reward R of a new policy π

$$R(\pi) = \mathbb{E}_{x\sim P(x)}\, \mathbb{E}_{y\sim\pi(\cdot|x)}\, E_{r\sim D(\cdot|x,y)}[r] \tag{2}$$

from $\mathcal{S}$. Analogously, the BLBF policy-learning problem lies in using the logged data $\mathcal{S}$ for finding a policy from some function class $\pi^* \in \Pi$ that maximizes the expected reward

$$\pi^* = \operatorname*{argmax}_{\pi\in\Pi} \Big[R(\pi)\Big]. \tag{3}$$

The value of a policy $R(\pi)$ cannot be calculated directly, but counterfactual estimators $\hat{R}(\pi)$ (Horvitz & Thompson, 1952; Strehl et al., 2011; Dudík et al., 2011; Swaminathan & Joachims, 2015b; Thomas & Brunskill, 2016; Wang et al., 2017; Farajtabar et al., 2018; Su et al., 2019) were proposed to estimate $R(\pi)$ from $\mathcal{S}$. These estimators enable Counterfactual Risk Minimization (CRM) for BLBF (Swaminathan & Joachims, 2015a), where the algorithm searches the policy space Π to maximize the counterfactual estimate, possibly subject to various forms of regularization.

Most previous works on BLBF followed a "Model the Bias" approach, where they model the selection bias in the assignment mechanism with importance sampling techniques for estimating the expected reward. One widely used approach is Inverse Propensity Score (IPS) weighting (Horvitz & Thompson, 1952; Strehl et al., 2011)

$$\hat{R}_{IPS}(\pi|\mathcal{S}) = \frac{1}{n}\sum_{i=1}^{n} \frac{\pi(y_i|x_i)}{\pi_0(y_i|x_i)} r_i \tag{4}$$

The IPS estimator provides an unbiased estimate of the expected reward of a policy under the common support condition.

Condition 1 (Common Support)**.** *The logging policy π_0 has full support for the target policy π, which means $\pi(y|x) > 0 \rightarrow \pi_0(y|x) > 0$ for all x and y.*

The challenge in using IPS weighting lies in the variance of the estimate, which can be large when the IPS weights $\frac{\pi(y_i|x_i)}{\pi_0(y_i|x_i)}$ are large. Variance regularization (Swaminathan & Joachims, 2015a) is thus proposed to alleviate the problem.

2.2. Direct Method and Reward Regression

Unlike most prior works, we use a "Model the World" approach for BLBF where a reward-regression model $\hat{\delta}(x,y)$ is learned to estimate $\delta(x,y) := \mathbb{E}_{r\sim D(x,y)}[r|x,y]$. Given $\hat{\delta}(x,y)$, the DM estimator can be used for evaluating any policy π

$$\hat{R}_{DM}(\pi|\hat{\delta},\mathcal{S}) = \frac{1}{n}\sum_{i=1}^{n}\sum_{y\in\mathcal{Y}} \pi(y_i|x_i)\hat{\delta}(x_i,y_i) \tag{5}$$

Furthermore, the policy that maximizes the DM estimate is easily derived from $\hat{\delta}(x,y)$ via

$$\pi(y|x) = \mathbb{1}\{y = \operatorname*{argmax}_{\bar{y}\in\mathcal{Y}} \hat{\delta}(x,\bar{y})\}. \tag{6}$$

Since there is typically low variability in the reward-regression model, the DM estimator often has small variance as discussed in Section 2.3. If we could learn a perfect reward-regression model (i.e., $\forall x,y:\ \hat{\delta}(x,y) = \delta(x,y)$), it is easy to see that the policy derived from the perfect reward-regression model is optimal. However, a perfect reward-regression model rarely exists in practice, since most models for real-world problems are misspecified. Therefore, the DM estimator can be substantially biased when the reward-regression model is misspecified and biased. This raises the question of how to train the reward-regression model to most effectively mitigate bias.

The naive approach to training the reward-regression model uses the standard least squares objective

$$\hat{\mathcal{L}}_{naive}(\hat{\delta}|\mathcal{S}) = \frac{1}{n}\sum_{i=1}^{n}(r_i - \hat{\delta}(x_i,y_i))^2, \tag{7}$$

which is minimized over some class of regressors Δ

$$\hat{\delta}^* = \operatorname*{argmin}_{\hat{\delta}\in\Delta} \hat{\mathcal{L}}_{naive}(\hat{\delta}|\mathcal{S}). \tag{8}$$

For misspecified models, the issue with the naive approach is that it fits well for the action distribution of the logging policy, but it may not provide accurate estimates for the policy we derive from it via (6). This lack of accuracy can mean that the derived policy is far from optimal. If we decompose the rewards into $r(x,y) = \delta(x,y) + \epsilon(x,y)$ with zero-mean noise $\epsilon(x,y)$ independent of $\delta(x,y)$ and $\hat{\delta}(x,y)$, then the expectation of $\hat{\mathcal{L}}_{naive}$ is

$$\mathbb{E}(\hat{\mathcal{L}}_{naive}) = \mathbb{E}_x\, \mathbb{E}_{y\sim\pi_0(\cdot|x)}[(\delta(x,y)-\hat{\delta}(x,y))^2+\epsilon(x,y)^2] \tag{9}$$

If the logging policy is poor, minimizing $\hat{\mathcal{L}}_{naive}$ means that the learned reward-regression model fits well for actions that are far from optimal, and it may be highly biased for high-reward actions.

To overcome the dependency of the bias on the logging policy, one could use importance weighting to shift the reward-regression model to the uniform distribution, which treats all actions equally.

$$\hat{\mathcal{L}}_{unif}(\hat{\delta}|\mathcal{S}) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{|\mathcal{Y}|\pi_0(y_i|x_i)}(r_i - \hat{\delta}(x_i,y_i))^2 \tag{10}$$

With the same assumption as for the naive approach,

$$\mathbb{E}(\hat{\mathcal{L}}_{unif}) = \mathbb{E}_x\, \mathbb{E}_{y\sim unif(\mathcal{Y})}[(\delta(x,y)-\hat{\delta}(x,y))^2+\epsilon(x,y)^2], \tag{11}$$

where $unif(\mathcal{Y})$ refers to the policy that selects actions uniformly. While this makes the expected estimate independent of the logging policy and its selection biases, we are still not explicitly accounting for the policy π that we aim to evaluate.

2.3. Bias Corrected Reward Imputation (BCRI)

The key idea behind BCRI is to optimize the regression estimates to minimize the bias of the DM estimator. As we will show, this is achieved by the following regression objective, where the losses are importance-weighted towards the target policy π.

$$\hat{\mathcal{L}}_{BCRI}(\hat{\delta}|\pi,\mathcal{S}) = \frac{1}{n}\sum_{i=1}^{n} \frac{\pi(y_i|x_i)}{\pi_0(y_i|x_i)}(r_i - \hat{\delta}(x_i,y_i))^2 \tag{12}$$

This regression objective corrects the selection bias induced by the logging policy π_0 so that it becomes in expectation equivalent to training the reward regressor with on-policy data from π. In this way, and with the same assumption as for $\hat{\mathcal{L}}_{naive}$ and $\hat{\mathcal{L}}_{unif}$, the BCRI-objective explicitly minimizes the bias of the regression estimator w.r.t. π as

$$\mathbb{E}(\hat{\mathcal{L}}_{BCRI}) = \mathbb{E}_x\, \mathbb{E}_{y\sim\pi(\cdot|x)}[(\delta(x,y)-\hat{\delta}(x,y))^2+\epsilon(x,y)^2]. \tag{13}$$

Most importantly, this bias minimization for the regression estimates translates into an upper bound on the mean squared error (MSE) of the DM estimator, as the following result shows. It requires the typical assumption of common support.

Theorem 1 (MSE Bound for DM via BCRI)**.** *For contexts $x_1, x_2, \cdots, x_n$ drawn i.i.d from some distribution $P(\mathcal{X})$, actions $y_i \sim \pi_0(\mathcal{Y}|x_i)$ drawn from the logging policy π_0 under Condition 1, rewards $r_i = \delta(x_i,y_i)+\epsilon_i$ with some underlying true reward function $\delta(x,y)$ and zero-mean noise ϵ_i, then for any bounded reward-regression model class Δ with $\hat{\delta}_{max} := max_{x,y,\hat{\delta}\in\Delta}|\hat{\delta}(x,y)|$ and any policy class π*

$$\begin{aligned} MSE(\hat{R}_{DM}) \leq\, & \mathbb{E}(\hat{\mathcal{L}}_{BCRI}) + \frac{\hat{\delta}^2_{max}}{n} \\ & - \mathbb{E}_x\, \mathbb{E}_{y\sim\pi(\cdot|x)}\, \epsilon^2(x,y) \end{aligned} \tag{14}$$

Proof. We first upper bound the squared bias of the DM estimator as

$$\begin{aligned} & Bias^2(\hat{R}_{DM}) \\ & = \left[\mathbb{E}_x\left[\sum_{y\in\mathcal{Y}} \pi(y|x)(\hat{\delta}(x,y) - \delta(x,y))\right]\right]^2 \\ & \leq \mathbb{E}_x\left[\sum_{y\in\mathcal{Y}} (\sqrt{\pi(y|x)})^2\right] \cdot \\ & \quad \mathbb{E}_x\left[\sum_{y\in\mathcal{Y}} (\sqrt{\pi(y|x)}(\hat{\delta}(x,y) - \delta(x,y)))^2\right] \\ & = \mathbb{E}_x\left[\sum_{y\in\mathcal{Y}} \pi(y|x)(\hat{\delta}(x,y) - \delta(x,y))^2\right] \end{aligned} \tag{15}$$

via an application of the Cauchy-Schwarz inequality. Similarly, we bound the variance of the DM estimator as

$$\begin{aligned} Var(\hat{R}_{DM}) =& \frac{1}{n} V_x\left[\sum_{y\in\mathcal{Y}} \pi(y|x)\hat{\delta}(x,y)\right] \\ =& \frac{1}{n}\, \mathbb{E}_x\left[(\sum_{y\in\mathcal{Y}} \pi(y|x)\hat{\delta}(x,y))^2\right] \\ & - \frac{1}{n}\left[\mathbb{E}_x\left[\sum_{y\in\mathcal{Y}} \pi(y|x)\hat{\delta}(x,y)\right]\right]^2 \\ \leq& \frac{1}{n}\, \mathbb{E}_x\left[\sum_{y\in\mathcal{Y}} \pi(y|x)\hat{\delta}(x,y)^2\right] \\ \leq& \frac{1}{n}\hat{\delta}^2_{max}. \end{aligned} \tag{16}$$

The first inequality is again an application of Cauchy−Schwarz, and we exploit that the second term is non-positive. The second inequality is a direct consequence of how $\hat{\delta}_{max}$ is defined. Combing (13), (15), and (16) proves (14). □

Note that the second term in the right-hand side of (14) decays quickly with the number of contexts that are sampled. The third term results from the noise in the rewards and it is unavoidable. While $\mathbb{E}(|\mathcal{Y}|\hat{\mathcal{L}}_{naive})$ is also an upper bound since $\pi(y|x) \leq 1$, it can be substantially less tight for mismatched models with non-zero bias. This explains our empirical findings in Section 3, where we find BCRI to perform better than training with the uniform objective (10). However, note that the bound does not account for the variability of $\hat{\mathcal{L}}_{BCRI}$, which we plan to explore in future work.

2.4. BCRI Policy Learning as Bi-level Optimization

The previous section showed that BCRI can lead to improved DM estimates, and we now explore how BCRI can be used inside of a policy-learning algorithm. Here we are faced with a "Chicken and Egg" problem, where we fit the reward regressor based on the target policy while at the same time deriving the target policy from the regressor. We formalize this as the following bi-level optimization problem. The upper level maximizes the expected reward of the policy according to the DM estimate, while the lower level provides the BCRI regression estimate.

$$\begin{aligned} \pi^{BCRI} &= \underset{\pi \in \Pi}{\operatorname{argmax}} \left[\hat{R}_{DM}(\pi|\hat{\delta}, \mathcal{S}) \right] \\ &\text{s. t. } \hat{\delta} = \underset{\hat{\delta} \in \Delta}{\operatorname{argmin}} \left[\hat{\mathcal{L}}_{BCRI}(\hat{\delta}|\pi, \mathcal{S}) \right] \end{aligned} \tag{17}$$

Note that the upper level is maximized in closed form via (6) for any fixed $\hat{\delta}$, and that the lower level has a convex loss function for any fixed π. We thus use the following simple strategy to find an approximate solution, although we anticipate that this can be substantially improved. In particular, we perform batch stochastic gradient descent for $\hat{\delta}$ on $\hat{\mathcal{L}}_{BCRI}(\hat{\delta}|\pi, \mathcal{S})$ in the lower level given the current π from the upper level. After each gradient step, we then update the policy π via (6) in the upper level and repeat until we have reached a fixed point. Note that this bears some resemblance with temporal-difference reinforcement learning (Sutton & Barto, 2018), where we sample examples for a gradient descent update — but with the difference that here we "sample off-policy" from the collected bandit feedback and use importance weighting to correct the bias.

As already mentioned, the variance of $\hat{\mathcal{L}}_{BCRI}$ can become large as the learned target policy deviates from the logging policy. To alleviate this problem, we restrict the policy class to softmax policies with temperature parameter T

$$\pi(y|x) = \frac{e^{\hat{\delta}(x,y)/T}}{\sum_{\bar{y} \in \mathcal{Y}} e^{\hat{\delta}(x,\bar{y})/T}} \tag{18}$$

during optimization. Compared to the argmax policy in (6), such stochastic policies typically have less extreme importance weights and thus lower variance. We empirically evaluate the effectiveness of softmax policies in Section 3, but conjecture that other methods of variance regularization may further improve performance.

3. Experiments

We empirically examine the performance of BCRI on bandit feedback derived form multi-class classification. This setting is widely used in the off-policy evaluation and learning literature (Dudík et al., 2011; Su et al., 2019). Specifically, we use real world multi-class classification datasets from which we sample synthetic bandit data. This provides the ground truth for the performance evaluation and enables varying different aspects of the contextual bandit setting for analyzing the performance in different situations.

3.1. Data Setup

For bandit data generation, we follow (Dudík et al., 2011; Su et al., 2019) using the standard supervised $\rightarrow$ bandit conversion for several multiclass classification datasets from the UCI repository (Asuncion & Newman, 2007). Formally, for a supervised dataset $\{(x_i, y_i^*)\}_{i=1}^m$ where x_i is the feature of the i^{th} example and y_i^* is its corresponding label, we use a small amount of the dataset to train a logging policy π_0. The bandit data is sampled according to the following process, context and label $x, y^* \sim unif(\{(x_1, y_1^*), ...(x_m, y_m^*)\})$, action $y \sim \pi_0(\cdot|x)$, reward $r(x, y) = \mathbb{1}\{y = y^*\}$. Repeating the process n times, we can get a set of bandit data $\mathcal{S} = \{x_i, y_i, r(x_i, y_i), \pi_0(\cdot|x_i)\}_{i=1}^n$ which is used for learning a new policy.

For each dataset, we split the dataset into train,validation and test sets randomly. 2% of the train set is used to learn a logging policy using a multi-class logistic regression and the rest of the train set is used to simulate bandit feedback. To ensure the common support assumption, during the deployment of the logging policy, 2% of the actions are randomly selected while the other actions are selected according to the trained logging policy. To test the performance of different methods with data at different scale, different amounts of the feedback are simulated. Then different methods are used to train models using the bandit data. The validation set is used to conduct hyper-parameter selection and we report the expected reward on the test set.

3.2. Experiment Setup

In the experiments, we compare the following methods, including the Hardmax and the Softmax version of BCRI and several baselines.

- BCRI (Hardmax): BCRI with policy π as the argmax of the reward-regression model during optimization.
- BCRI (Softmax): BCRI with policy π as the softmax

Table 1. Expected reward on test set for the Letter and SatImage datasets at different training-sample sizes.

Dataset	Letter	Letter	Letter	SatImage	SatImage	SatImage
#of train contexts	9600	19200	48000	400	800	2000
Naive DM	0.502	0.496	0.540	0.746	0.751	0.776
Uniform DM	0.552	0.553	0.583	0.753	0.761	0.799
BanditNet	0.360	0.399	0.431	**0.790**	**0.817**	0.803
BCRI(Hardmax)	0.513	0.608	0.652	0.762	0.800	0.795
BCRI(Softmax)	**0.622**	**0.651**	**0.666**	0.771	0.802	**0.809**

of the reward-regression model during optimization.

- Naive: DM with the naive regression objective (7).
- Unif: DM with the uniform regression objective (10).
- BanditNet: IPS-based BanditNet (Joachims et al., 2018).

To ensure a fair comparison, we use the same feature map and linear model for both policy-based and regression based methods. For the policy-based methods, following (Swaminathan & Joachims, 2015b; Su et al., 2019), the function class of the policy is $\mathcal{F} := \{\pi_w : w \in R^p\}$ with π_w as the stochastic linear rules defined by

$$\pi_w(y|x) = \frac{\exp(w^T \phi(x, y))}{\mathbb{Z}(x)} \tag{19}$$

where $\mathbb{Z}(x) = \sum_{y' \in \mathcal{Y}} exp(w \cdot \phi(x, y'))$ is the partition function and $\phi(x, y)$ denotes the joint feature map between context x and action y. The hypothesis space of the reward-regression model is in $\mathcal{G} := \{\hat{\delta}_w : w \in R^p\}$ where $\hat{\delta}_w(x, y) = w^T \phi(x, y)$. Following (Swaminathan & Joachims, 2015a), assuming $\vec{y}$ is the 0-1 vector representation of the label y, the feature map we used is $\phi(x, y) = x \otimes \vec{y}$ where $\otimes$ denotes outer product.

For all the methods, we add L2 regularization for the parameters w. We grid-search learning rate, L2 regularization parameter and the batch size. Stochastic gradient descent is conducted with a momentum parameter set at 0.9 for optimization. All the methods are trained 1500 epochs to ensure convergence. For BCRI with softmax policy update, we also grid-search the temperature T. To reduce the variance of different methods with importance weights, we adopt the widely used clipping technique to replace $\frac{\pi(y_i|x_i)}{\pi_0(y_i|x_i)}$ or $\frac{1}{|\mathcal{Y}|\pi_0(y_i|x_i)}$ in the objective with $min\{\frac{\pi(y_i|x_i)}{\pi_0(y_i|x_i)}, M\}$ or $min\{\frac{1}{|\mathcal{Y}|\pi_0(y_i|x_i)}, M\}$ and also grid-search the clipping parameter M. For BanditNet, we also grid-search the Lagrange multiplier hyperparameter.

3.3. Empirical Results

The test set performance of all methods for both datasets at three training-sample sizes is shown in Table 3. First, BCRI(Softmax) consistently outperforms the Naive DM and the Uniform DM baselines across all datasets and sample sizes. This empirically confirms the theoretical motivation for BCRI. Among the baselines, Uniform DM performs better than Naive DM. Second, BCRI(Softmax) also substantially outperforms BanditNet on the Letter Recognition dataset. Upon inspection, we conjecture that the reason lies in BanditNet getting stuck in bad local optima due to the large action space (26 classes) that leads to gradient-saturation when optimizing the non-convex objective. On the SatImage dataset, BCRI(Softmax) performs comparably to BanditNet. Third, comparing the softmax and the hardmax versions of BCRI, we find that the softmax policies perform better than the hardmax policies, which we conjecture is due to the improved variance control in the BCRI regression objective. In future work, we thus plan to explore methods to more directly control variance to further improve estimation and learning performance.

4. Conclusion

In this work, we propose BCRI as a method for minimizing the bias of the reward regressor w.r.t. the target policy, thus optimizing the MSE of the DM estimator for the practically important case of misspecified models. We formulate the BCRI policy-learning problem as a bi-level optimization problem and provide a strategy for optimizing the solution. We empirically find that BCRI is consistently better than other reward-regression baselines across different datasets at different scales. Compared to BanditNet, BCRI can provide substantial performance gains when the action space is large, and the convexity of the reward-regression objective in BCRI appears to make its optimization more robust.

In future work, we plan to incorporate the variance of the regression estimates into our theoretical framework and into the training objective. Furthermore, we will explore other strategies for solving the bi-level optimization problem, and evaluate high-capacity deep models.

References

Asuncion, A. and Newman, D. Uci machine learning repository, 2007.

Bottou, L., Peters, J., Quiñonero-Candela, J., Charles, D. X., Chickering, D. M., Portugaly, E., Ray, D., Simard, P., and Snelson, E. Counterfactual reasoning and learning systems: The example of computational advertising. *The Journal of Machine Learning Research*, 14(1):3207–3260, 2013.

Dudík, M., Langford, J., and Li, L. Doubly robust policy evaluation and learning. In *International Conference on Machine Learning (ICML)*, 2011.

Farajtabar, M., Chow, Y., and Ghavamzadeh, M. More robust doubly robust off-policy evaluation. In *International Conference on Machine Learning(ICML)*, 2018.

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

Joachims, T., Swaminathan, A., and de Rijke, M. Deep learning with logged bandit feedback. In *International Conference on Learning Representations (ICLR)*, 2018.

Kallus, N. Balanced policy evaluation and learning. In *Advances in Neural Information Processing Systems(NeurIPS)*, 2018.

Strehl, A., Langford, J., Li, L., and Kakade, S. M. Learning from logged implicit exploration data. In *Advances in Neural Information Processing Systems (NIPS)*, 2011.

Su, Y., Wang, L., Michele, S., and Joachims, T. Cab: Continuous adaptive blending estimator for policy evaluation and learning. In *International Conference on Machine Learning(ICML)*, 2019.

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018.

Swaminathan, A. and Joachims, T. Batch learning from logged bandit feedback through counterfactual risk minimization. *Journal of Machine Learning Research (JMLR)*, 16:1731–1755, Sep 2015a. Special Issue in Memory of Alexey Chervonenkis.

Swaminathan, A. and Joachims, T. The self-normalized estimator for counterfactual learning. In *Neural Information Processing Systems (NIPS)*, 2015b.

Thomas, P. and Brunskill, E. Data-efficient off-policy policy evaluation for reinforcement learning. In *International Conference on Machine Learning(ICML)*, 2016.

Wang, Y.-X., Agarwal, A., and Dudik, M. Optimal and adaptive off-policy evaluation in contextual bandits. In *International Conference on Machine Learning(ICML)*, 2017.