Learning from Logged Bandit Feedback of Multiple Loggers

Yi Su YS756@CORNELL.EDU
Cornell Universtity, Dept. of Statistical Science, Ithaca, NY, USA

Aman Agarwal AA2398@CORNELL.EDU
Cornell Universtity, Dept. of Computer Science, Ithaca, NY, USA

Thorsten Joachims TJ@CS.CORNELL.EDU
Cornell Universtity, Dept. of Computer Science, Ithaca, NY, USA

Abstract

We explore the problem of Batch Learning from Bandit Feedback (BLBF) in a setting where the training data comes from multiple logging policies. This setting is ubiquitous in search, recommendation, and ad placement, since these systems are regularly updated by deploying new policies in production or in A/B testing. Unfortunately, we find that naively using this heterogeneous log data can lead to situations where adding more training data decreases learning accuracy. To overcome this problem, we propose a learning algorithm that incorporates a weighted estimator, providing optimal variance within its class. Empirical results confirm that this leads to substantial improvements in learning accuracy.

1. Introduction

Learning improved policies from logged interaction data is a core problem in many applications, including personalized medicine, search engines, recommendation systems, and ad-placement (Beygelzimer and Langford, 2009; Bottou et al., 2013; Athey and Wager, 2017; Lefortier et al., 2016). A key challenge with such log data is that we only observe the outcomes of the deployed action, but not the potential outcomes of all possible actions the system could have taken (aka contextual bandit feedback). For example, in an ad placement system, we only observe a user's click-feedback for the ad that was presented by the system, but not for the alternative ads that could have been presented instead. The Counterfactual Risk Minimization (CRM) principle (Swaminathan and Joachims, 2015a) has been proposed to learn in this setting, called batch learning from bandit feedback (BLBF). The key idea is to control for differences in variance between target policies when performing Empirical Risk Minimization (ERM) based on a counterfactual risk estimator, especially the inverse-propensity-score (IPS) weighting estimator (Rosenbaum and Rubin, 1983).

This paper makes the case that naively applying CRM to the setting where the log data comes from multiple policies can be highly sub-optimal. Our work centers on the idea that, to effectively learn from all the information, the log data from different policies must be weighted differently. To this end, we derive a weighted counterfactual learning principle for BLBF under multiple loggers and then design an algorithm for it.

Our work builds upon recent results for effective counterfactual evaluation with multiple loggers (Agarwal et al., 2017), where a weighted inverse propensity score (WIPS) estimator is proposed by re-weighting data from different policies by their "divergence" from the target policy. The WIPS estimator is optimal in the sense that this particular choice of weights will give the smallest variance among all weighted estimators. A major challenge in extending this approach to learning is that the weights depend on the *target* policy, and thus need to be updated within the learning algorithm along with the policy being learned (in other words, we have a moving target). To overcome this issue, we formulate the learning problem as a bi-level optimization problem (Colson et al., 2007; Bennett et al., 2008), updating the weights for different loggers and finding the optimal policy simultaneously. Luckily, the lower level of the bi-level optimization problem has a closed form solution, which lets us effectively escape the compu-

CausalML Workshop at the International Conference on Machine Learning (ICML), Stockholm, Sweden, 2018.

tational inefficiency of bi-level optimization problems. Another challenge is that the optimal weights depend on the divergences between the target policy and the historical policies, which requires that we have access to full information about all the input-action-loss tuples. Since this is impossible in the BLBF setting, we exploit that the empirical version of this divergence is a consistent estimator and can be a good surrogate for the true divergence (Agarwal et al., 2017). However, for relatively peaked policy distributions, we observed that the empirical divergence estimator converges very slowly, and thus using it in the weight calculation brings inaccurate weight estimation leading to a poor learning outcome. Motivated by this, we propose a better empirical divergence estimator using control variates (Owen, 2013). To assess effectiveness, we evaluate our learning algorithm on several multi-label classification problems and show that it can provides substantially improved generalization error over naively using the CRM principle without consideration for multiple logging policies.

2. Method

To recap the formulation of the BLBF problem, we have contexts $x \in \mathcal{X}$ drawn i.i.d from a fixed but unknown distribution $Pr(\mathcal{X})$. For each particular context, the system draws the action $y \in \mathcal{Y}$ based on the policy $\pi(\mathcal{Y}|x)$ and receives feedback $\delta(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$. Here, we use δ to denote the loss induced by the action, the lower the better. The learning algorithm considers a hypothesis space $\mathcal{F}$ of stochastic polices $\pi(\mathcal{Y}|x)$, where each is a probability distribution over the action space $\mathcal{Y}$. The risk (i.e. expected loss) for policy $\pi \in \mathcal{F}$ is defined as:

$$U(\pi) = \mathbb{E}_{y\sim\pi(\mathcal{Y}|x),x\sim Pr(\mathcal{X})}[\delta(x, y)] \tag{1}$$

The goal of learning we consider in this paper is to find an policy $\pi \in \mathcal{F}$ that has low risk, using the logged data from multiple logging policies $\pi_1, \pi_2, \cdots, \pi_m$. The log data we get from each historical policy π_i is:

$$\mathcal{D}_i = \{(x_1^i, y_1^i, \delta_1^i, p_1^i), \cdots, (x_{n_i}^i, y_{n_i}^i, \delta_{n_i}^i, p_{n_i}^i)\} \tag{2}$$

with $\mathcal{D} = \bigcup_{i=1}^m \mathcal{D}_i$. Here n_i is the number of data points collected from historical policy π_i, with each data point $x_j^i \overset{\text{i.i.d}}{\sim} Pr(\mathcal{X})$, and $y_j^i \sim \pi(\mathcal{Y}|x_j^i)$, where $j \in \{1, 2, \ldots, n_i\}$. The loss and propensity score for each point is $\delta_j^i \equiv \delta(x_j^i, y_j^i)$, and $p_j^i = \pi_i(y_j^i|x_j^i)$.

2.1. Weighted IPS estimator

The weighted IPS (WIPS) estimator (Agarwal et al., 2017) for evaluation introduces the following "divergence" as a measure of mismatch between target policy π and logging policy π_i.

$$\sigma_\delta^2(\pi||\pi_i) \equiv \mathrm{Var}_{x\sim Pr(\mathcal{X}),y\sim\pi_i(\mathcal{Y}|x)}\left[\frac{\delta(x, y)\pi(y|x)}{\pi_i(y|x)}\right] \tag{3}$$

For simplicity, let us define the re-weighted loss for each data point and their mean loss as

$$U_j^i(\pi) = \delta_j^i \frac{\pi(y_j^i|x_j^i)}{\pi_i(y_j^i|x_j^i)}, \tag{4}$$

$$U^i(\pi) = \frac{1}{n_i}\sum_{i=1}^{n_i} U_j^i(\pi). \tag{5}$$

Then the WIPS estimator can be written as

$$\hat{R}(\pi) = \sum_{i=1}^m \mathbf{p}^T \mathbf{U}(\pi), \tag{6}$$

where $\mathbf{p} = \{p_i\}_{i=1}^m$, $\mathbf{U}(\pi) = \{U^i(\pi)\}_{i=1}^m$ and the weight vector $\mathbf{p}$ is defined by $\mathbf{p} = \arg\min_{p\in\Delta_m} \mathrm{Var}_{\mathcal{D}}(\hat{R}(\pi))$ and has the closed form solution

$$p_i = \frac{n_i}{\sigma_\delta^2(\pi||\pi_i)\sum_{j=1}^m \frac{n_j}{\sigma_\delta^2(\pi||\pi_j)}}. \tag{7}$$

It was shown in (Agarwal et al., 2017) that these weights achieve minimal variance.

2.2. Better Divergence Estimator

Unfortunately, we do not have access to the information necessary for computing the divergence $\sigma_\delta^2(\pi||\pi_i)$ and thus the optimal weights. At first thought, we may therefore consider using the empirical divergence estimator

$$\hat{\sigma}_\delta^2(\pi||\pi_i) = \frac{1}{n_i - 1}\sum_{j=1}^{n_i}(U_j^i(\pi) - U^i(\pi))^2. \tag{8}$$

However, this estimator performs poorly when the logging policy π_i is peaked and the true divergence is large, which causes inaccurate weight estimates and strongly degrades the learning outcome. Here, we provide an improved empirical variance estimator, which introduces a multiplicative correction inspired by the idea of control variates (Owen, 2013; Swaminathan and Joachims, 2015b). For simplicity, define the control variate $S^i(\pi)$ and the overall mean of $U_j^i(\pi)$ as

$$S^i(\pi) = \frac{1}{n_i}\sum_{j=1}^{n_i} \frac{\pi(y_j^i|x_j^i)}{\pi_i(y_j^i|x_j^i)}, \tag{9}$$

$$\bar{U}(\pi) = \frac{1}{\sum_{i=1}^m n_i}\sum_{i=1}^m\sum_{j=1}^{n_i} U_j^i(\pi). \tag{10}$$

$$\tag{11}$$

Our self-normalized divergence estimator is then defined as

$$\tilde{\sigma}^2_\delta(\pi||\pi_i) = \frac{1}{n_i - 1}\sum_{j=1}^{n_i}\left(\frac{U^i_j(\pi)}{S^i(\pi)} - \bar{U}(\pi)\right)^2. \quad (12)$$

We can show that this estimator is consistent even though it is not unbiased. Dividing $U^i_j(\pi)$ by $S^i(\pi)$ exploits that each term $\frac{\pi(y^i_j|x^i_j)}{\pi_i(y^i_j|x^i_j)}$ is typically correlated with $U^i_j(\pi)$ and that $\mathbb{E}[S^i(\pi)] = 1$. Using the overall $\bar{U}(\pi)$ instead of the IPS estimate for each specific logger π_i utilizes information from all the data and provides a more informed estimate. Note that we could also add a multiplicative control variate into this term. However, we found that combining all data already results in a sufficiently accurate estimator of $\bar{U}$ and that the self-normalized term does not help much. Dividing by $S^i(\pi)$ in the first term is crucial, though. Empirically the improved estimator performs better than the naive divergence estimator in terms of learning performance, especially when the historical policy is nearly deterministic.

2.3. Weighted Counterfactual Risk Minimization Principle

We now present the Weighted Counterfactual Risk Minimization (WCRM) principle, which is motivated by an empirical Bernstein argument (Maurer and Pontil, 2009) analogous to the CRM principle for a single logger (Swaminathan and Joachims, 2015a). The learning principle minimizes the WIPS estimator and its empirical standard deviation at the same time, for the aim of robust learning.

$$\pi^{WCRM} = \underset{\pi\in\mathcal{F},\mathbf{p}\in\Delta_m}{\arg\min}\ \mathbf{p}^T\mathbf{U}^M(\pi) + \lambda\sqrt{\frac{\hat{\mathrm{Var}}(p_i U^i_j(\pi))}{\sum_{k=1}^m n_k}}$$

subject to

$$\mathbf{p} = \underset{\mathbf{p}\in\Delta_m}{\arg\min}\,\mathrm{Var}_{\mathcal{D}}(\hat{R}(\pi)) \quad (13)$$

Luckily, the lower level problem has a closed-form solution as stated in (7) and we estimated it using the better divergence estimator stated in (11). We omit the proof that the true risk is upper bounded by the weighted CRM objective with high probability (similar to CRM proof for single logger with an additional covering argument for the simplex in which the weights lie). To further control variance, we adjust our weighted loss for each data point to be the clipped version $U^i_j(\pi) = \delta^i_j \min\{M, \frac{\pi(y^i_j|x^i_j)}{\pi_i(y^i_j|x^i_j)}\}$, where M serves as a hyper-parameter for "clipping" (avoid unbounded variance from $\pi_i(y^i_j|x^i_j) \cong 0$).

Following (Lafferty et al., 2001; Swaminathan and Joachims, 2015a) in the choice of hypothesis space $\mathcal{F}$, in the following we consider learning stochastic linear rules $\pi_w \in \mathcal{F}$ (each policy π is parameterized by vector w)

$$\pi_w(y|x) = \frac{\exp(w^T\phi(x,y))}{\mathbb{Z}(x)}. \quad (14)$$

$\phi(x,y)$ is the joint feature map of input x and action y, and $\mathbb{Z}(x)$ is the normalization factor. Besides this, we introduce a "stochastic multiplier" a, which transforms the function vector w to aw, which will be used in our later experiments.

3. Experiments

In this section, we empirically examine the generalization performance of the weighted CRM in comparison to the conventional CRM that does not account for multiple loggers. We follow the same supervised $\rightarrow$ bandit transformation as in (Agarwal et al., 2014). We chose the multi-label datasets from LibSVM for the experiments. Each multi-label classification dataset contains input $x \in \mathbb{R}^p$ and target labels $y^* \in \{0,1\}^q$. For any supervised dataset $\mathcal{D} = \{(x_1, y^*_1), \cdots, (x_n, y^*_n)\}$, we collect bandit dataset by simulating $y_i \sim \pi_i(\mathcal{Y}|x_i)$ and report the loss $\delta(y^*_i, y_i)$ associated with this sample by the number of correctly predicted labels compared with the ground truth y^*_i.

For each dataset, we kept aside 20% to train and get the historical logging policy. For simplicity, we use two logging policies in the following experiment. The first logger π_1 (aka bad logger) is trained using a CRF on these 20% of data, and we then set the stochastic multiplier to be {-1,-0.8,-0.5,-0.3,0,0.3,0.5,0.8,1} (-1 looks like inverting the probability of reasonably good policy). The second logger π_2 (aka good logger) is trained on the same data with stochastic multiplier to be 1. We kept another 20% of data as validation set (choosing hyperparameter λ). We optimizate the WCRM and CRM objectives using L-BFGS (Lewis and Overton, 2013) from scikit-learn (Pedregosa et al., 2011). We test performance of the learned policies π^{WCRM} and π^{CRM} by calculating the expected loss per test instance $\hat{R}(\pi) = \frac{1}{n_{test}}\sum_{i=1}^{n_{test}} \mathbb{E}_{y_i\sim\pi^{WCRM}(\mathcal{Y}|x_i)}[\delta(y^*_i, y_i)]$. Results for each experiment are averaged over 10 runs.

The baseline to beat is the naive CRM principle. We report the expected hamming loss of both WCRM and naive CRM, the lower the better. Figure 1 compares the performance of WCRM and naive CRM as the stochastic multiplier of logger 1 ranges from -1 to 1. For both datasets, WCRM maintains good performance even as the quality of logger 1 is degraded, while the naive CRM approach is severly affected.

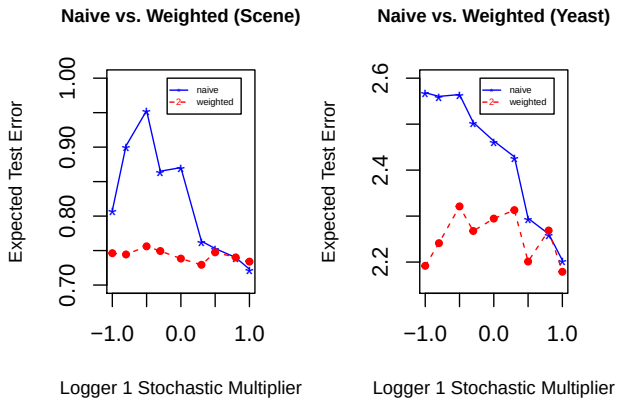

Figure 1. Left: Expected Test Error of the policy learned by naive CRM and WCRM for the Scene dataset (296 features, 4 labels). Right: Expected Test Error of the policy learned by naive CRM and WCRM for the Yeast dataset (104 features, 6 labels).

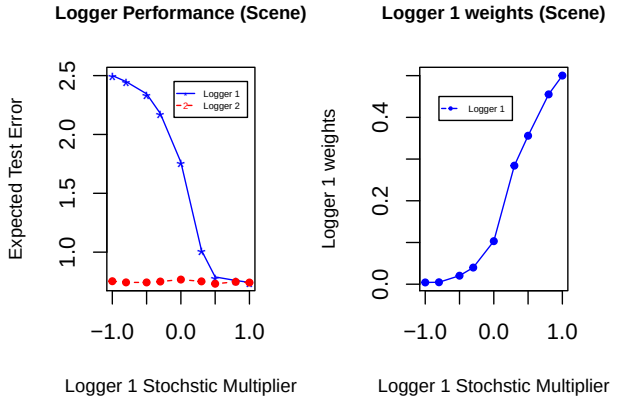

Figure 2. Left: Expected Test Error of the CRM policies learned by only using data generated from Logger 1 or Logger 2 respectively (Scene Dataset). Right: Weight of logger 1 as chosen by the WCRM evaluated at the learned policy (Scene Dataset).

Nevertheless, the performance of WCRM slightly improves towards the right-hand side of the plots as expected, since it has access to more high-quality data for stochastic multipliers close to 1. Further note that WCRM and naive CRM perform similarly when the stochastic multiplier for logger 1 approaches 1, since at this point both loggers are nearly identical and the naive CRM applies.

To give further insight into the performance gains of the WCRM, the left-hand plot in Figure 2 shows the performance of CRM when training only on data from logger 1 or logger 2 respectively. This is shown to sanity check that logger 1 indeed provides poor data for learning a good policy in the BLBF setting. The right-hand plot of Figure 2 shows the final weight p_1 for logger 1 chosen by the WCRM and evaluated at the π^{WCRM} that it learned. As expected, the WCRM puts more weight on logger 1 (i.e. $p_1 \approx 0.5$) when its data quality is good, and it down-weights logger 1 (i.e. $p_1 \approx 0.0$) when its data quality would degrade the variance of the learning objective.

Acknowledgments

This work was supported by NSF awards IIS-1615706 and IIS-1513692, and through gifts from Criteo and Amazon. This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE-1650441. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

Agarwal, A., Basu, S., Schnabel, T., and Joachims, T. (2017). Effective evaluation using logged bandit feedback from multiple loggers. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*.

Agarwal, A., Hsu, D. J., Kale, S., Langford, J., Li, L., and Schapire, R. E. (2014). Taming the monster: A fast and simple algorithm for contextual bandits. In *31st International Conference on Machine Learning*, pages 1638–1646.

Athey, S. and Wager, S. (2017). Efficient policy learning. *arXiv preprint arXiv:1702.02896*.

Bennett, K. P., Kunapuli, G., Hu, J., and Pang, J.-S. (2008). Bilevel optimization and machine learning. *In IEEE World Congress on Computational Intelligence, Springer*, pages 25–47.

Beygelzimer, A. and Langford, J. (2009). The offset tree for learning with partial labels. In *15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 129–138.

Bottou, L., Peters, J., Candela, J. Q., Charles, D. X., Chickering, M., Portugaly, E., Ray, D., Simard, P. Y., and Snelson., E. (2013). Counterfactual reasoning and learning systems: the example of compu-

tational advertising. *Journal of Machine Learning Research*, 14(1):3207–3260.

Colson, B., Marcotte, P., and Savard, G. (2007). An overview of bilevel optimization. *Annals of Operations Research*, pages 235–256.

Lafferty, J. D., McCallum, A., and Pereira, F. C. N. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning*, pages 282–289.

Lefortier, D., Swaminathan, A., Gu, X., Joachims, T., and de Rijke, M. (2016). Large-scale validation of counterfactual learning methods: A test-bed. In *NIPS 2016 What-If Workshop*.

Lewis, A. S. and Overton, M. L. (2013). Nonsmooth optimization via quasi-newton methods. *Mathematical Programming*, 141(1-2):135–163.

Maurer, A. and Pontil, M. (2009). Empirical bernstein bounds and sample-variance penalization. In *22nd Conference on Learning Theory*.

Owen, A. (2013). *Monte Carlo theory, methods and examples*.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830.

Rosenbaum, P. R. and Rubin, D. B. (1983). The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55.

Swaminathan, A. and Joachims, T. (2015a). Counterfactual risk minimization: Learning from logged bandit feedback. In *International Conference on Machine Learning (ICML)*, pages 814–823.

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