Learning from eXtreme Bandit Feedback

Romain Lopez[1], **Inderjit S. Dhillon**[2, 3], **Michael I. Jordan**[1, 2]

[1] Department of Electrical Engineering and Computer Sciences, University of California, Berkeley
[2] Amazon.com
[3] Department of Computer Science, The University of Texas at Austin
{romain_lopez, jordan}@cs.berkeley.edu
inderjit@cs.utexas.edu

Abstract

We study the problem of batch learning from bandit feedback in the setting of extremely large action spaces. Learning from extreme bandit feedback is ubiquitous in recommendation systems, in which billions of decisions are made over sets consisting of millions of choices in a single day, yielding massive observational data. In these large-scale real-world applications, supervised learning frameworks such as eXtreme Multi-label Classification (XMC) are widely used despite the fact that they incur significant biases due to the mismatch between bandit feedback and supervised labels. Such biases can be mitigated by importance sampling techniques, but these techniques suffer from impractical variance when dealing with a large number of actions. In this paper, we introduce a *selective importance sampling estimator* (sIS) that operates in a significantly more favorable bias-variance regime. The sIS estimator is obtained by performing importance sampling on the conditional expectation of the reward with respect to a small subset of actions for each instance (a form of Rao-Blackwellization). We employ this estimator in a novel algorithmic procedure—named Policy Optimization for eXtreme Models (POXM)—for learning from bandit feedback on XMC tasks. In POXM, the selected actions for the sIS estimator are the top-p actions of the logging policy, where p is adjusted from the data and is significantly smaller than the size of the action space. We use a supervised-to-bandit conversion on three XMC datasets to benchmark our POXM method against three competing methods: BanditNet, a previously applied partial matching pruning strategy, and a supervised learning baseline. Whereas BanditNet sometimes improves marginally over the logging policy, our experiments show that POXM systematically and significantly improves over all baselines.

Introduction

In the classical supervised learning paradigm, it is assumed that every data point is accompanied by a label. Such labels provide a very strong notion of feedback, where the learner is able to assess not only the loss associated with the action that they have chosen but can also assess losses of actions that they did not choose. A useful weakening of this paradigm involves considering so-called “bandit feedback,” where the training data simply provides evaluations of selected actions without delineating the correct action. Bandit feedback is often viewed as the province of reinforcement learning, but it is also possible to combine bandit feedback with supervised learning by considering a batch setting in which each data point is accompanied by an evaluation and there is no temporal component. This is the Batch Learning from Bandit Feedback (BLBF) problem (Swaminathan and Joachims 2015a).

Of particular interest is the off-policy setting where the training data is provided by a *logging policy*, which differs from the learner’s policy and differs from the optimal policy. Such problems arise in many real-world problems, including supply chains, online markets, and recommendation systems (Rahul, Dahiya, and Singh 2019), where abundant data is available in a logged format but not in a classical supervised learning format.

Another difficulty with the classical notion of a “label” is that real-world problems often involve huge action spaces. This is the case, for example, in real-world recommendation systems where there may be billions of products and hundreds of millions of consumers. Not only is the cardinality of the action space challenging both from a computational point of view and a statistical point of view, but even the semantics of the labels can become obscure—it can be difficult to place an item conceptually in one and only category. Such challenges have motivated the development of eXtreme multi-label classification (XMC) and eXtreme Regression (XR) (Bhatia et al. 2016) methods, which focus on computational scalability issues and target settings involving millions of labels. These methods have had real-world applications in domains such as e-commerce (Agrawal et al. 2013) and dynamic search advertising (Prabhu et al. 2018, 2020).

We assert that the issues of bandit feedback and extreme-scale action spaces are related. Indeed, it is when action spaces are large that it is particularly likely that feedback will only be partial. Moreover, large action spaces tend to support multiple tasks and grow in size and scope over time, making it likely that available data will be in the form of a logging policy and not a single target input-output mapping.

We also note that the standard methodology for accommodating the difference between the logging policy and an optimal policy needs to be considered carefully in the setting of large action spaces. Indeed, the standard methodology is some form of importance sampling (Swaminathan and Joachims 2015a), and importance sampling estimators can run aground when their variance is too high (see, e.g., Lefortier et al. (2016)). Such variance is likely to be particularly virulent in large action spaces. Some examples of the XMC framework do treat labels as subject to random variation (Jain, Prabhu, and Varma 2016), but only with the goal of improving the prediction of rare labels; they do not tackle the broader problem of learning from logging policies in extreme-scale action spaces. It is precisely this broader problem that is our focus in the current paper.

The literature on offline policy learning in Reinforcement Learning (RL) has also been concerned with correcting for implicit feedback bias (see, e.g., Degris, White, and Sutton (2012)). This line of work differs from ours, however, in that the focus in RL is not on extremely-large action spaces, and RL is often based on simulators rather than logging policies (Chen et al. 2019c; Bai, Guan, and Wang 2019). Closest to our work is the work of Chen et al. (2019b), who propose to use offline policy gradients on a large action space (millions of items). Their method relies, however, on a proprietary action embedding, unavailable to us.

After a brief overview of BLBF and XMC , we present a new form of BLBF that blends bandit feedback with multi-label classification. We introduce a novel assumption, specific to the XMC setting, in which most actions are irrelevant (i.e., incur a null reward) for a particular instance. This motivates a Rao-Blackwellized (Casella and Robert 1996) estimator of the policy value for which only a small set of relevant actions per instance are considered. We refer to this approach as *selective importance sampling* (sIPS). We provide a theoretical analysis of the bias-variance tradeoff of the sIPS estimator compared to naive importance sampling. In practice, the selected actions for the sIPS estimator are the top-p actions from the logging policy, where p can be adjusted from the data. We derive a novel learning method based on the sIPS estimator, which we refer to as *Policy Optimizer for eXtreme Models* (POXM). Finally, we propose a modification of a state-of-the-art neural XMC method AttentionXML (You et al. 2019b) to learn from bandit feedback. Using a supervised-learning-to-bandit conversion (Dudik, Langford, and Li 2011), we benchmark POXM against BanditNet (Joachims, Swaminathan, and de Rijke 2018), a partial matching scheme from Wang et al. (2016) and a supervised learning baseline on three XMC datasets (EUR-Lex, Wiki10-31K and Amazon-670K) (Bhatia et al. 2016). We show that naive application of the state-of-the-art method BanditNet (Joachims, Swaminathan, and de Rijke 2018) sometimes improves over the logging policy, but only marginally. Conversely, POXM provides substantial improvement over the logging policy as well as supervised learning baselines.

Background

eXtreme Multi-label Classification (XMC)

Multi-label classification aims at assigning a relevant subset $\boldsymbol{Y} \subset [L]$ to an instance x, where $[L] := \{1, \dots, L\}$ denotes the set of L possible labels. XMC is a specific case of multi-label classification in which we further assume that all $\boldsymbol{Y}$ are small subsets of a massive collection (i.e., generally $|\boldsymbol{Y}|/L < 0.01$). Naive one-versus-all approaches to multi-label classification usually do not scale to such a large number of labels and adhoc methods are often employed. Furthermore, the marginal distribution of labels across all instances exhibits a long tail, which causes additional statistical challenges.

Algorithmic approaches to XMC include optimized one-versus-all methods (Babbar and Schölkopf 2017, 2019; Yen et al. 2017, 2016), embedding-based methods (Bhatia et al. 2015; Tagami 2017; Guo et al. 2019), probabilistic label tree-based (Prabhu et al. 2018; Jasinska et al. 2016; Khandagale, Xiao, and Babbar 2020; Wydmuch et al. 2018) and deep learning-based methods (You et al. 2019b; Liu et al. 2017; You et al. 2019a; Chang et al. 2019). Each algorithm usually proposes a specific approach to model the text as well as deal with tail labels. For example, Babbar and Schölkopf (2019) uses a robust SVM approach on TF-IDF features. PfastreXML (Jain, Prabhu, and Varma 2016) assumes a particular noise model for the observed labels and proposes to weight the importance of tail labels. AttentionXML (You et al. 2019b) uses a bidirectional-LSTM to embed the raw text as well as a multi-label attention mechanism to help capture the most relevant part of the input text for each label. For datasets with large L, AttentionXML trains one model per layer of a shallow and wide probabilistic latent tree using a small set of candidate labels.

Batch Learning from Bandit Feedback (BLBF)

We assume that the instance x is sampled from a distribution $\mathcal{P}(x)$. The action for this particular instance is a unique label $y \in [L]$, sampled from the logging policy $\rho(y \mid x)$ and a feedback value $r \in \mathbb{R}$ is observed. Repeating this data collection process yields the dataset $[(x_i, y_i, r_i)]_{i=1}^n$. The BLBF problem consists in maximizing the expected reward $V(\pi)$ of a policy π. We use importance sampling (IS) to estimate $V(\pi)$ from data based on the logging policy as follows:

$$\hat{V}_{\text{IS}}(\pi) = \frac{1}{n}\sum_{i=1}^{n}\frac{\pi(y_i \mid x_i)}{\rho(y_i \mid x_i)}r_i. \tag{1}$$

Classically, identifying the optimal policy via this estimator is infeasible without a thorough exploration of the action space by the logging policy (Langford, Strehl, and Wortman 2008). More specifically, the IS estimator $\hat{V}_{\text{IS}}(\pi)$ requires the following basic assumption for there to be any hope of asymptotic optimality:

Assumption 1. *There exists a scalar $\epsilon > 0$ such that for all $x \in \mathbb{R}^d$ and $y \in [L]$, $\rho(y \mid x) > \epsilon$.*

The IS estimator has high variance when π assigns actions that are infrequent in ρ; hence a variety of regularization schemes have been developed, based on risk-upper-bound minimization, to control variance. Examples of upper bounds include empirical Bernstein concentration bounds (Swaminathan and Joachims 2015a) and various divergence-based bounds (Atan, Zame, and Mihaela Van Der Schaar 2018; Wu and Wang 2018; Johansson, Shalit, and Sontag 2016; Lopez et al. 2020). Another common strategy for reducing the variance is to propose a model of the reward function, using as a baseline a doubly robust estimator (Dudik, Langford, and Li 2011; Su et al. 2019).

A recurrent issue with BLBF is that the policy may avoid actions in the training set when the rewards are not scaled properly; this is the phenomenon of *propensity overfitting*. Swaminathan and Joachims (2015b) tackled this problem via the self-normalized importance sampling estimator (SNIS), in which IS estimates are normalized by the average importance weight. SNIS is invariant to translation of the rewards and may be used as a safeguard against propensity overfitting. BanditNet (Joachims, Swaminathan, and de Rijke 2018) made this approach amenable to stochastic optimization by translating the reward distribution:

$$\hat{V}_{\text{BN}}(\pi) = \frac{1}{n}\sum_{i=1}^{n}\frac{\pi(y_i \mid x_i)}{\rho(y_i \mid x_i)}\left[r_i - \lambda\right], \tag{2}$$

and selecting λ over a small grid based on the SNIS estimate of the policy value.

Learning an XMC algorithm from bandit feedback requires offline learning from slates $\boldsymbol{Y}$, where each element of the slate comes from a large action space. Swaminathan et al. (2017) proposes a pseudo-inverse estimator for offline learning from combinatorial bandits. However, such an approach is intractable for large action spaces as it requires inverting a matrix whose size is linear in the number of actions. Another line of work focuses on offline evaluation and learning of semi-bandits for ranking (Li et al. 2018; Joachims, Swaminathan, and Schnabel 2017) but only with a small number of actions. In real-world data, a partial matching strategy between instances and relevant actions is applied in applications to internet marketing for policy evaluation (Wang et al. 2016; Li, Kim, and Zitouni 2015). More recently, Chen et al. (2019b) proposed a top-k off-policy correction method for a real-world recommender system. Their approach deals with millions of actions although it treats label embeddings as given, whereas this problem is in general a hard problem for XMC.

Bandit Feedback and Multi-label Classification

We consider a setting in which the algorithm (e.g., a policy for a recommendation system) observes side information $x \in \mathbb{R}^d$ and is allowed to output a subset $\boldsymbol{Y} \subseteq [L]$ of the L possible labels. Side information is independent at each round and sampled from a distribution $\mathcal{P}(x)$. We assume that the subset $\boldsymbol{Y}$ has fixed size $|\boldsymbol{Y}| = \ell$, which allows us to adopt the slate notation $\boldsymbol{Y} = (y_1, \ldots, y_\ell)$. The algorithm observes noisy feedback for each label, $\boldsymbol{R} = (r_1, \ldots, r_\ell)$, and we further assume that the joint distribution over $\boldsymbol{R}$ decomposes as $\mathcal{P}\left(\boldsymbol{R} \mid x, \boldsymbol{Y}\right) = \prod_{j=1}^{\ell}\mathcal{P}\left(r_j \mid x, y_j\right)$. We will denote the conditional reward distribution as a function: $\delta(x, y) = \mathbb{E}[r \mid x, y]$. In the case of multi-label classification, this feedback can be formed with random variables indicating whether each individual label is inside the true set of labels for each datapoint (Gentile and Orabona 2014). More concretely, feedback may be formed from sale or click information (Chen et al. 2019b,c).

We are interested in optimizing a policy $\pi(\boldsymbol{Y} \mid x)$ from offline data. Accessible data is sampled according to an existing algorithm, the logging policy $\rho(\boldsymbol{Y} \mid x)$. We assume that both joint distributions over the slate decompose into an auto-regressive process. For example, for π we assume:

$$\pi(\boldsymbol{Y} \mid x) = \prod_{j=1}^{\ell}\pi(y_j \mid x, y_{1:j-1}). \tag{3}$$

Introducing this decomposition does not result in any loss of generality, as long as the action order is identifiable (otherwise, one would need to consider all possible orderings (Kool, van Hoof, and Welling 2020b)). This is a reasonable hypothesis because the order of the actions may also be logged as supplementary information. We now define the value of a policy π as:

$$V(\pi) = \mathbb{E}_{\mathcal{P}(x)}\mathbb{E}_{\pi(\boldsymbol{Y}\mid x)}\mathbb{E}\left[\boldsymbol{1}^\intercal\boldsymbol{R} \mid x, \boldsymbol{Y}\right]. \tag{4}$$

In our setting, the reward decomposes as a sum of independent contributions of each individual action. The reward may in principle be generalized to be rank dependent, or to consider interactions between items (Gentile and Orabona 2014), but this is beyond the scope of this work.

A general approach for offline policy learning is to estimate $V(\pi)$ from logged data using importance sampling (Swaminathan and Joachims 2015a). As emphasized in Swaminathan et al. (2017), the combinatorial size of the action space $\Omega(L^\ell)$ may yield an impractical variance for importance sampling. This is particularly the case for XMC, where typical values of L are minimally in the thousands. A natural strategy to improve over the IS estimator on the slate $\boldsymbol{Y}$ is to exploit the additive reward decomposition in Eq. (4). Along with the factorization of the policy in Eq. (3), we may reformulate the policy value as:

$$V(\pi) = \mathbb{E}_{\mathcal{P}(x)}\sum_{j=1}^{\ell}\mathbb{E}_{\pi(y_{1:j}\mid x)}\delta(x, y_j). \tag{5}$$

The benefit of this new decomposition is that instead of performing importance sampling on $\boldsymbol{Y}$, we can now use ℓ IS estimators, each with a better bias-variance tradeoff. Unbiased estimation of $V(\pi)$ in Eq. (5) via importance sampling still requires Assumption 1. The logging policy must therefore explore a large action space. However, most actions are unrelated to a given context and deploying an online logging policy that satisfies Assumption 1 may yield a poor customer experience.

Learning from eXtreme Bandit Feedback

We now explore alternative assumptions for the logging policy that may be more suitable to the setting of very large action spaces. We formalize the notion that most actions are irrelevant using the following assumption:

Assumption 2. (Sparse feedback condition). *The individual feedback random variable r takes values in the bounded interval $[\nabla, \Delta]$. For all $x \in \mathbb{R}^d$, the label set $[L]$ can be partitioned as $[L] = \Psi(x) \coprod \Psi^0(x)$ such that for all actions y of $\Psi^0(x)$, the expected reward is minimal: $\delta(x, y) = \nabla$.*

We refer to the function Ψ as an *action selector*, as it maps a context to a set of relevant actions. Throughout the manuscript, we use the notation Λ^0 to refer to the pointwise set complement of any action selector Λ. Intuitively, we are interested in the case where $|\Psi(x)| \ll L$ for all x. Assumption 2 is implicitly used in online marketing applications of offline policy evaluation, formulated as a partial matching between actions and instances (Wang et al. 2016; Li, Kim, and Zitouni 2015). Notably, this assumption can be assimilated to a mixed-bandit feedback setting, where we observe feedback for all of $\Psi^0(x)$ but only one selected action inside of $\Psi(x)$. Under Assumption 2, the IS estimator will be unbiased for all logging policies that satisfy the following relaxed assumption:

Assumption 3. (Ψ-overlap condition). *There exists a scalar $\epsilon > 0$ such that for all $x \in \mathbb{R}^d$ and $y \in \Psi(x)$, $\rho(y \mid x) > \epsilon$.*

Batch learning from bandit feedback may be possible under this assumption, as long as the logging policy explores a set of actions large enough to cover the actions from Ψ but small enough to avoid exploring too many suboptimal actions. Furthermore, Assumption 2 also reveals the existence of $\Psi^0(x)$, a sufficient statistic for estimating the reward on the irrelevant actions. Making appeal to Rao-Blackwellization (Casella and Robert 1996), we can incorporate this information to estimate each of the ℓ terms of Eq. (5) (e.g., in the case $\ell = 1$ and $\nabla = 0$):

$$V(\pi) = \mathbb{E}_{\mathcal{P}(x)}\left[\pi\left(\Psi(x) \mid x\right) \cdot \mathbb{E}_{\pi(y|x)}\left[\delta(x, y) \mid y \in \Psi(x)\right]\right]. \tag{6}$$

The decomposition in Eq. (6) suggests that when the action selector Ψ is known, one can estimate $V(\pi)$ via importance sampling for the conditional expectation of the rewards with respect to the event $\{y \in \Psi(x)\}$. Intuitively, this means that one can modify the importance sampling scheme to only include a relevant subset of labels and ignore all the others. Without loss of generality, we assume that $\nabla = 0$ in the remainder of this manuscript.

In practice, the oracle action selector Ψ is unknown and needs to be estimated from the data. It may be hard to infer the smallest Ψ such that Assumption 2 is satisfied. Conversely, a trivial action selector including all actions is valid (it does include all relevant actions) but is ultimately unpractical. As a flexible compromise, we will replace Ψ in Eq. (6) by any action selector Φ and study the bias-variance tradeoff of the resulting plugin estimator.

Let ρ be a logging policy with a large enough support to satisfy Assumption 3. Let Φ be an action selector such that $\Phi(x) \subset \operatorname{supp} \rho(\cdot \mid x)$ almost surely in x, where supp denotes the support of a probability distribution. The role of Φ is to prune out actions to maintain an optimal bias-variance tradeoff. In the case $\ell = 1$, the Φ-selective importance sampling (sIS) estimator $\hat{V}^{\Phi}_{\mathrm{sIS}}(\pi)$ for action selection Φ can be written as:

$$\hat{V}^{\Phi}_{\mathrm{sIS}}(\pi) = \frac{1}{n}\sum_{i=1}^{n} \frac{\pi(y_i \mid x_i, y \in \Phi(x))}{\rho(y_i \mid x_i)} r_i. \tag{7}$$

Its bias and variance depends on how different the policy π is from the logging policy ρ (as in classical BLBF) but also on the degree of overlap of Φ with Ψ:

Theorem 1 (Bias-variance tradeoff of selective importance sampling)**.** *Let $\boldsymbol{R}$ and ρ satisfy Assumptions 2 and 3. Let Φ be an action selector such that $\Phi(x) \subset \operatorname{supp} \rho(\cdot \mid x)$ almost surely in x. The bias of the sIS estimator is:*

$$\left|\mathbb{E}\hat{V}^{\Phi}_{sIS}(\pi) - V(\pi)\right| \leq \Delta\kappa(\pi, \Psi, \Phi), \tag{8}$$

where $\kappa(\pi, \Psi, \Phi) = \mathbb{E}_{\mathcal{P}(x)}\pi\left(\Psi(x) \cap \Phi^0(x) \mid x\right)$ quantifies the overlap between the oracle action selector Ψ and the proposed action selector Φ, weighted by the policy π. The performance of the two estimators can be compared as follows:

$$\begin{aligned} MSE\left[\hat{V}^{\Phi}_{sIS}(\pi)\right] \leq{}& MSE\left[\hat{V}_{IS}(\pi)\right] + 2\Delta^2\kappa(\pi, \Psi, \Phi) \\ &- \frac{\sigma^2}{n}\mathbb{E}_{\mathcal{P}(x)}\frac{\pi^2\left(\Phi^0(x) \mid x\right)}{\rho\left(\Phi^0(x) \mid x\right)}, \end{aligned} \tag{9}$$

where $\sigma^2 = \inf_{x,y \in \mathbb{R}^d \times [K]} \mathbb{E}[r^2 \mid x, y]$.

We provide the complete proof of this theorem in the appendix[1]. As expected by Rao-Blackellization, we see that if Φ completely covers Ψ (i.e., for all $x \in \mathbb{R}^d, \Psi(x) \subset \Phi(x)$), then $\hat{V}^{\Phi}_{\mathrm{sIS}}(\pi)$ is unbiased and has more favorable performance than $\hat{V}_{\mathrm{IS}}(\pi)$. Admittedly, Eq. (9) shows that both estimators have similar mean-square error when π puts no mass on potentially irrelevant actions $y \in \Phi^0(x)$. However, during the process of learning the optimal policy or in the event of propensity overfitting, we expect π to put a non-zero mass on potentially irrelevant actions $y \in \Phi^0(x)$, with positive probability in x. For these reasons, we expect $\hat{V}^{\Phi}_{\mathrm{sIS}}(\pi)$ to provide significant improvement over $\hat{V}_{\mathrm{IS}}(\pi)$ for policy learning.

Even though Eq. (9) provides insight into the performance of sIS, unfortunately it cannot be used directly in selecting Φ. We instead propose a greedy heuristic to select a small number of action selectors. For example, $\Phi^p(x)$ corresponds to the top-p labels for instance x according to the logging policy. With this approach, the bias of the $\hat{V}^{\Phi}_{\mathrm{sIS}}(\pi)$ estimator is a decreasing function of p, as the overlap with Ψ increases. Furthermore, the variance increases with p as long as the added actions are irrelevant. In practice, we use a small grid search for $p \in \{10, 20, 50, 100\}$ and choose the optimal p with the SNIS estimator, as in BanditNet. We believe this is a reasonable approach whenever the logging policy ranks the relevant items sufficiently high but can be improved (e.g., top-p for p in 5 to 100).

Policy Optimization for eXtreme Models

We apply sIS for each of the ℓ terms of the policy value from Eq. (5) in order to estimate $V(\pi)$ from bandit feedback, $(x_i, \boldsymbol{Y}_i, \boldsymbol{R}_i)_{i=1}^n$, and learn an optimal policy. As an additional step to reduce the variance, we prune the importance sampling weights of earlier slate actions, following Achiam et al. (2017):

$$\hat{V}^{\Phi}_{\mathrm{sIS}}(\pi) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{\ell} \frac{\pi^{\Phi}(y_{i,j} \mid x_i, y_{1:j-1})}{\rho(y_{i,j} \mid x_i, y_{1:j-1})} r_{i,j}, \tag{10}$$

[1] Please visit https://arxiv.org/abs/2009.12947 for supplementary information.

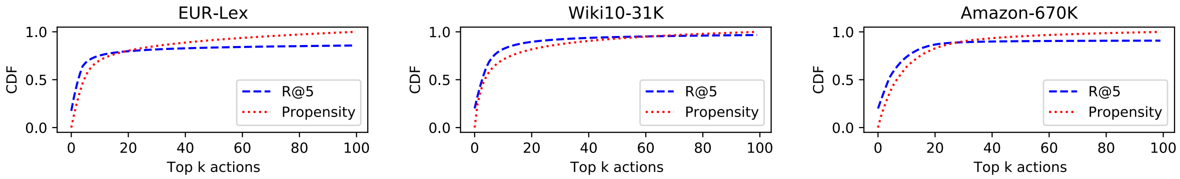

Figure 1: Expected $R@5$ and CDF of the logging policy for the top-k action for each XMC dataset. Exploration is limited to a subset of relevant actions.

where π^{Φ} designates the distribution π restricted to the set $\Phi(x)$ for every x. Because $\pi(\boldsymbol{Y} \mid x)$ is a copula, one can derive all joint distributions starting from the corresponding one-dimensional marginals (Sklar 1959). In this work, we focus on the case of ordered sampling without replacement to respect an important design restriction: the slate $\boldsymbol{Y}$ must not have redundant actions. For the j-th slate component, the relevant conditional probability is formed from the base marginal probabilities $\pi(y \mid x)$ as follows:

$$\pi^{\Phi}(y_j \mid x, y_{1:j-1}) = \frac{\pi(y_j \mid x)}{\sum\limits_{y' \in \Phi(x)} \pi(y' \mid x) - \sum\limits_{k<j} \pi(y_k \mid x)}. \quad (11)$$

From a computational perspective, the action selector also diminishes the computational burden, leading to efficient computations of the probabilities when the marginals are parameterized by a softmax distribution. Indeed, Eq. (11) depends only on the logits for the actions inside of the set Φ. This helps our approach to scale to large XMC datasets.

As mentioned in the background section, directly maximizing the importance sampling estimate of the policy value in Eq. (10) may be pathological due to propensity overfitting. The BanditNet approach may be adapted to the slate case using a different loss translation scheme for each element:

$$\hat{V}^{\Phi}_{\text{sIS}}(\pi) = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{\ell} \frac{\pi^{\Phi}(y_{i,j} \mid x_i, y_{1:j-1})}{\rho(y_{i,j} \mid x_i, y_{1:j-1})} [r_{i,j} - \lambda_j], \quad (12)$$

and with $(\lambda_1, \dots, \lambda_\ell)$ selected out of a small grid based on the self-normalized importance sampling estimate of the policy value from the training data (Joachims, Swaminathan, and de Rijke 2018). For computational reasons, we only search for a unique λ and, following Joachims, Swaminathan, and de Rijke (2018), we focus on the grid $\{0.7, 0.8, 0.9, 1.0\}$. We refer to this approach as *Policy Optimization for eXtreme Models* (POXM), named after the seminal algorithm from Swaminathan and Joachims (2015a).

Experiments

We evaluate our approach on real-world datasets with a supervised learning to bandit feedback conversion (Dudik, Langford, and Li 2011; Gentile and Orabona 2014). We report results on three datasets from the Extreme Classification Repository (Bhatia et al. 2016), with L ranging from several thousand to half a million (Table 1). EUR-Lex (Mencia and Fürnkranz 2008) has a relatively small label set and each instance has a sparse label set. Wiki10-31K (Zubiaga 2012) has a larger label set as well as more abundant annotations. Finally, Amazon-670K (McAuley and Leskovec 2013) has more than half a million labels. To our knowledge, this is the first time that such action spaces have been considered for BLBF.

Dataset	N_{train}	N_{test}	D	L	$\overline{L}$	$\hat{L}$
EUR-Lex	15,449	3,865	186,104	3,956	5.30	20.79
Wiki10-31K	14,146	6,616	101,938	30,938	18.64	8.52
Amazon-670K	490,449	153,025	135,909	670,091	5.45	3.99

N_{train}: #training instances, N_{test}: #test instances, D: #features, L: #labels and size of the action space, $\overline{L}$: average #labels per instance, $\hat{L}$: the average #instances per label. The partition of training and test is from the data source.

Table 1: XMC datasets used for semi-simulation of eXtreme bandit feedback.

Simulating Bandit Feedback from XMC Datasets

An XMC dataset is a collection of observations $(x_i, \boldsymbol{Y_i^*})_{i=1}^n$ for which each instance x_i is associated with an optimal set of labels $\boldsymbol{Y_i^*}$. To form a logging policy ρ, we train AttentionXML on a small fraction α of the dataset to get estimates of the marginal probability for each label (values are provided in the appendix). These probabilities must be normalized in order to sum to one, as expected in the multi-label setting (Wydmuch et al. 2018). The ground-truth labels may be used to investigate whether Φ^p (the top p actions from ρ) approximately satisfies the Ψ-covering condition. On the EUR-LeX dataset the obtained logging policy on its top 20 action covers around 75% of the rewards (Figure 1). Using more actions may be suboptimal as these may add variance with only a marginal benefit on the bias, as captured by Theorem 1. Finally, we form bandit feedback for slates of size ℓ by sampling without replacement from ρ. The reward is a binary variable depending on whether the chosen action belongs to the reference set $\boldsymbol{Y}^*$. We fix $\ell = 5$ in all experiments.

Evaluation Metrics

P@k (Precision at k), nDCG@k (normalized Discounted Cumulative Gain at k) as well as PSP@k (Propensity Scored Precision at k) are widely used metrics for evaluating XMC methods (Jain, Prabhu, and Varma 2016; Bhatia et al. 2016). We adapt these metrics to the evaluation of stochastic policies by taking expectations of the relevant statistics over

Methods	R@3	R@5	nDCR@3	nDCR@5	PSR@3	PSR@5
			EUR-Lex			
Logging policy	33.79	31.23	33.77	34.07	22.33	21.66
Direct Method	39.58	32.22	42.64	38.69	25.81	26.58
BanditNet	15.60	13.51	17.68	16.29	8.58	8.48
PM-BanditNet	20.44	15.13	24.17	20.42	9.51	9.52
POXM	**52.38**	**44.48**	**55.73**	**51.64**	**35.42**	**35.25**
AttentionXML	*73.08*	*61.10*	*76.37*	*70.49*	*51.29*	*53.86*
			Wiki10-31K			
Logging policy	42.49	38.80	43.57	41.13	7.43	7.46
Direct Method	48.96	38.16	55.72	46.38	8.22	7.78
BanditNet	49.92	36.16	56.54	45.08	7.20	7.20
PM-BanditNet	49.06	37.04	55.91	45.74	7.09	7.04
POXM	**60.45**	**53.03**	**64.22**	**58.26**	**10.70**	**10.58**
AttentionXML	*77.78*	*68.78*	*79.94*	*73.19*	*17.05*	*17.93*
			Amazon-670K			
Logging policy	17.89	17.05	18.77	18.65	13.06	13.06
Direct Method	23.42	20.14	25.16	23.28	15.82	16.30
BanditNet	16.83	14.54	17.18	16.11	11.67	11.67
PM-BanditNet	17.31	14.76	17.67	16.42	12.05	12.05
POXM	**26.89**	**23.72**	**28.93**	**27.22**	**19.59**	**20.75**
AttentionXML	*40.67*	*36.94*	*43.04*	*41.35*	*32.36*	*35.12*

Table 2: Performance comparisons of POXM and other competing methods over the three medium-scale datasets. All experiments are conducted with bandit feedback. In italic are the results from the AttentionXML manuscript, for the full-information feedback on all the training data (the supervised learning skyline).

slates of size k (with distinct items). For example, R@k (Reward at k) is defined as:

$$\mathrm{R@}k = \mathbb{E}_{\pi(y_1,\dots,y_k)} \frac{1}{k} \sum_{l=1}^{k} \mathbb{1}\{y_l \in \boldsymbol{Y}^*\}. \tag{13}$$

Similarly, we define nDCR@k and PSR@k (analogous to nDCG@k and PSP@k). We estimate those metrics using sampling without replacement.

Competing Methods and Experimental Settings

We compare POXM to other offline policy learning methods in the specific context of AttentionXML (You et al. 2019b). Furthermore, hyperparameters that are specific to AttentionXML are fixed across all experiments (Table 2 of You et al. (2019b)) so that our results are not confounded by those choices. To reduce training time and focus on how well each method deals with large action spaces, we use the LSTM weights from AttentionXML and treat those as fixed for all experiments. Finally, we noticed that the scale of gradients of the objective function for IS-based methods was different from supervised learning methods (similarly reported in Joachims, Swaminathan, and de Rijke (2018)). Consequently, we lowered the learning rate for these algorithms from $1\mathrm{e}{-4}$ to $5\mathrm{e}{-5}$.

We compare POXM to several baselines. First, we report results for the Direct Method (DM), a supervised learning baseline where AttentionXML is trained with a partial classification loss, using only the feedback from ℓ actions for each instance. The deterministic policy picks the top-k actions from the predicted value, akin to Prabhu et al. (2020). Second, we use BanditNet as a baseline. For this, we train AttentionXML using gradients of Eq. (10), but without conditioning on action set Φ. Instead, we use the full softmax (akin to Joachims, Swaminathan, and de Rijke (2018)) or we approximate it with negative sampling (Mikolov et al. 2013) at training time (only for the Amazon-670K dataset). Finally, we also investigate the effect of the partial matching strategy of Wang et al. (2016); Li, Kim, and Zitouni (2015) while training with BanditNet (referred to as BanditNet-PM). In this baseline we ignore feedback from actions that are not in Φ^p.

Results

Table 2 shows the performance results of POXM and other competing methods. POXM consistently outperformed the logging policy and always significantly improved over the competing methods. As expected, the performance is lower than AttentionXML learned on the full training set. The direct method also improved over the logging policy but only marginally, which is attributable to the bias from the logging policy. BanditNet and its partial matching variant did not improve over the logging policy on both EUR-Lex and Amazon-670K. We believe this is due to the sparsity of the rewards. Indeed, BanditNet outperforms the logging policy as well as the Direct Method baseline on Wiki10-31K that has many more labels per instance. Furthermore, we see that partial matching has a positive effect on BanditNet for EUR-LeX but not for the other datasets.

For all choices of logging policy in Figure 1, the optimal value of p selected by POXM is the smallest possible (p =

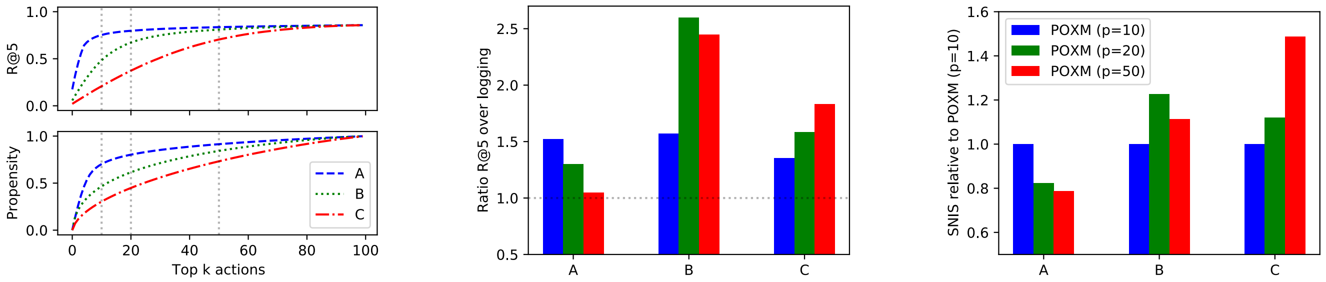

Figure 2: Data-driven selection of p on the EUR-LeX dataset. Left: logging policy statistics for three randomization scenarios (A, B, C, described in appendix). Middle: R@5 performance for each POXM variant and each logging policy. Right: SNIS estimates used for selection of p in POXM.

10). Therefore, we investigated how the algorithm behaved with more stochastic policies on the EUR-LeX dataset. For this, we injected Gumbel noise into the label probabilities (details in the appendix) and analyzed the performance of POXM for logging policies with $p \in \{10, 20, 50\}$. We provide summary statistics for the three logging policies and report the results of POXM in Figure 2. We see that each logging policy has a best performing value of p (middle) that is aligned with the summary statistics of the logging policy (left) as well as the normalized importance sampling (SNIS) policy value estimate (right). This shows that POXM keeps improving over the logging policy for more stochastic policies and that SNIS is a reasonable procedure for selecting the parameter p.

Discussion

We have presented POXM: a scalable algorithmic framework for learning XMC classifiers from bandit feedback. On real-world datasets, we have shown that POXM is systematically able to improve over the logging policy. This is not the case for the current state-of-the-art method, BanditNet. The latter does not always improve over the logging policy, which may be attributable to propensity overfitting.

All public datasets for eXtreme multi-label classification present the problem of imbalanced label distribution. Indeed, certain important labels (commonly referred to as *tail labels*), with more descriptive power, might be rarely used because of biases inherent to the data collection process. Although we do not provide a specific treatment of tail labels in this manuscript, we proposed in the appendix a simple extension of POXM (named wPOXM) based on Jain, Prabhu, and Varma (2016) to address this problem. Briefly, we extended the traditional data generating process for BLBF to treat the labels as noisy, and assumed that our observation scheme is biased towards the head labels. This leads to a slight modification of the sIS estimator and the POXM procedure to include the label propensity scores. wPOXM significantly improved over POXM for all propensity-weighted metrics, with 4.77% improvement of the PSR@3 metric. We leave more refined analyses for future work.

An important point in the XMC literature is computational efficiency. In this study, we used a machine with 8 GPUs Tesla K80 to run our experiments. This is mainly because our implementation relies on AttentionXML, itself implemented in PyTorch. The runtime of POXM on each dataset ranges from less than one hour for EUR-LeX to less than three hours for Amazon-670K. An important aspect of POXM's implementation is the reduced softmax computation. We verified this on the Amazon-670K dataset in which we tracked the runtime for growing size of the parameter p. For less than $p \leq 100$ actions, POXM took around 3s to backpropagate through 1,000 samples. However, this runtime was multiplied by ten for p=10,000 (25s) and we could not run POXM for $p \geq$ 20,000 because of an out-of-memory error. An interesting research direction would be to apply this framework to other XMC algorithms.

A performance gap remains between POXM and the skyline performance from the supervised method AttentionXML. It is possible that alternative parameterizations of the policy π may further improve performance; for example, using a probabilistic latent tree for policy gradients as in Chen et al. (2019a) or using the Gumbel-Top-k trick (Kool, van Hoof, and Welling 2020a). Furthermore, doubly robust estimators (Dudik, Langford, and Li 2011; Su et al. 2019; Wang et al. 2019) may further help in incorporating prior knowledge about the reward function.

Acknowledgements

We acknowledge Kush Batia, Lerrel Pinto, Adam Stooke, Sujay Sanghavi, Hsiang-Fu Yu, Arya Mazumdar and Rajat Sen for helpful conversations.

References

Achiam, J.; Held, D.; Tamar, A.; and Abbeel, P. 2017. Constrained Policy Optimization. In *International Conference on Machine Learning*.

Agrawal, R.; Gupta, A.; Prabhu, Y.; and Varma, M. 2013. Multi-label learning with millions of labels: Recommending advertiser bid phrases for web pages. In *International World Wide Web Conference*.

Atan, O.; Zame, W. R.; and Mihaela Van Der Schaar. 2018. Counterfactual Policy Optimization Using Domain-Adversarial Neural Networks. In *ICML CausalML Workshop*.

Babbar, R.; and Schölkopf, B. 2017. Dismec: Distributed sparse machines for extreme multi-label classification. In *International Conference on Web Search and Data Mining*.

Babbar, R.; and Schölkopf, B. 2019. Data scarcity, robustness and extreme multi-label classification. *Machine Learning* .

Bai, X.; Guan, J.; and Wang, H. 2019. A Model-Based Reinforcement Learning with Adversarial Training for Online Recommendation. In *Advances in Neural Information Processing Systems*.

Bhatia, K.; Dahiya, K.; Jain, H.; Mittal, A.; Prabhu, Y.; and Varma, M. 2016. The extreme classification repository: Multi-label datasets and code. URL http://manikvarma.org/downloads/XC/XMLRepository.html.

Bhatia, K.; Jain, H.; Kar, P.; Varma, M.; and Jain, P. 2015. Sparse local embeddings for extreme multi-label classification. In *Advances in Neural Information Processing Systems*.

Casella, G.; and Robert, C. P. 1996. Rao-Blackwellisation of sampling schemes. *Biometrika* .

Chang, W.-C.; Yu, H.-F.; Zhong, K.; Yang, Y.; and Dhillon, I. 2019. X-BERT: eXtreme Multi-label Text Classification with using Bidirectional Encoder Representations from Transformers. *arXiv* .

Chen, H.; Dai, X.; Cai, H.; Zhang, W.; Wang, X.; Tang, R.; Zhang, Y.; and Yu, Y. 2019a. Large-scale interactive recommendation with tree-structured policy gradient. In *AAAI Conference on Artificial Intelligence*.

Chen, M.; Beutel, A.; Covington, P.; Jain, S.; Belletti, F.; and Chi, E. H. 2019b. Top-k off-policy correction for a REINFORCE recommender system. In *International Conference on Web Search and Data Mining*.

Chen, X.; Li, S.; Li, H.; Jiang, S.; Qi, Y.; and Song, L. 2019c. Generative Adversarial User Model for Reinforcement Learning Based Recommendation System. In *International Conference on Machine Learning*.

Degris, T.; White, M.; and Sutton, R. S. 2012. Off-policy actor-critic. In *International Conference on Machine Learning*.

Dudik, M.; Langford, J.; and Li, L. 2011. Doubly Robust Policy Evaluation and Learning. In *International Conference on Machine Learning*.

Gentile, C.; and Orabona, F. 2014. On multilabel classification and ranking with bandit feedback. *The Journal of Machine Learning Research* .

Guo, C.; Mousavi, A.; Wu, X.; Holtmann-Rice, D.; Kale, S.; Reddi, S.; and Kumar, S. 2019. Breaking the Glass Ceiling for Embedding-Based Classifiers for Large Output Spaces. In *Advances in Neural Information Processing Systems*.

Jain, H.; Prabhu, Y.; and Varma, M. 2016. Extreme multi-label loss functions for recommendation, tagging, ranking & other missing label applications. In *International Conference on Knowledge Discovery and Data Mining*.

Jasinska, K.; Dembczynski, K.; Busa-Fekete, R.; Pfannschmidt, K.; Klerx, T.; and Hullermeier, E. 2016. Extreme F-measure maximization using sparse probability estimates. In *International Conference on Machine Learning*.

Joachims, T.; Swaminathan, A.; and de Rijke, M. 2018. Deep Learning with Logged Bandit Feedback. In *International Conference on Learning Representations*.

Joachims, T.; Swaminathan, A.; and Schnabel, T. 2017. Unbiased learning-to-rank with biased feedback. In *International Conference on Web Search and Data Mining*.

Johansson, F.; Shalit, U.; and Sontag, D. 2016. Learning Representations for Counterfactual Inference. In *International Conference on Machine Learning*.

Khandagale, S.; Xiao, H.; and Babbar, R. 2020. Bonsai-diverse and shallow trees for extreme multi-label classification. *Machine Learning* .

Kool, W.; van Hoof, H.; and Welling, M. 2020a. Ancestral Gumbel-Top-k Sampling for Sampling Without Replacement. *Journal of Machine Learning Research* .

Kool, W.; van Hoof, H.; and Welling, M. 2020b. Estimating gradients for discrete random variables by sampling without replacement. In *International Conference on Learning Representations*.

Langford, J.; Strehl, A.; and Wortman, J. 2008. Exploration scavenging. In *International Conference on Machine learning*.

Lefortier, D.; Swaminathan, A.; Gu, X.; Joachims, T.; and de Rijke, M. 2016. Large-scale Validation of Counterfactual Learning Methods: A Test-Bed. In *What If workshop: NeurIPS*.

Li, L.; Kim, J. Y.; and Zitouni, I. 2015. Toward predicting the outcome of an A/B experiment for search relevance. In *International Conference on Web Search and Data Mining*.

Li, S.; Abbasi-Yadkori, Y.; Kveton, B.; Muthukrishnan, S.; Vinay, V.; and Wen, Z. 2018. Offline evaluation of ranking policies with click models. In *International Conference on Knowledge Discovery and Data Mining*.

Liu, J.; Chang, W.-C.; Wu, Y.; and Yang, Y. 2017. Deep learning for extreme multi-label text classification. In *International Conference on Research and Development in Information Retrieval*.

Lopez, R.; Li, C.; Yan, X.; Xiong, J.; Jordan, M.; Qi, Y.; and Song, L. 2020. Cost-Effective Incentive Allocation via Structured Counterfactual Inference. In *AAAI Conference in Artificial Intelligence*.

McAuley, J.; and Leskovec, J. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *International Conference on Recommender Systems*.

Mencia, E. L.; and Fürnkranz, J. 2008. Efficient pairwise multilabel classification for large-scale problems in the legal domain. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*.

Prabhu, Y.; Kag, A.; Harsola, S.; Agrawal, R.; and Varma, M. 2018. Parabel: Partitioned label trees for extreme classification with application to dynamic search advertising. In *International World Wide Web Conference*.

Prabhu, Y.; Kusupati, A.; Gupta, N.; and Varma, M. 2020. Extreme Regression for Dynamic Search Advertising. In *International Conference on Web Search and Data Mining*.

Rahul; Dahiya, H.; and Singh, D. 2019. A Review of Trends and Techniques in Recommender Systems. In *International Conference on Internet of Things: Smart Innovation and Usages*.

Sklar, A. 1959. Fonctions de repartition a n-dimensions et leurs marges. *Publications de l'Institut Statistique de l'Université de Paris* .

Su, Y.; Wang, L.; Santacatterina, M.; and Joachims, T. 2019. CAB: Continuous adaptive blending for policy evaluation and learning. In *International Conference on Machine Learning*.

Swaminathan, A.; and Joachims, T. 2015a. Counterfactual Risk Minimization: Learning from Logged Bandit Feedback. In *International Conference on Machine Learning*.

Swaminathan, A.; and Joachims, T. 2015b. The Self-Normalized Estimator for Counterfactual Learning. In *Advances in Neural Information Processing Systems*.

Swaminathan, A.; Krishnamurthy, A.; Agarwal, A.; Dudik, M.; Langford, J.; Jose, D.; and Zitouni, I. 2017. Off-policy evaluation for slate recommendation. In *Advances in Neural Information Processing Systems*.

Tagami, Y. 2017. AnnexML: Approximate nearest neighbor search for extreme multi-label classification. In *International Conference on Knowledge Discovery and Data Mining*.

Wang, L.; Bai, Y.; Bhalla, A.; and Joachims, T. 2019. Batch Learning from Bandit Feedback through Bias Corrected Reward Imputation. In *ICML Workshop on Real-World Sequential Decision Making*.

Wang, Y.; Yin, D.; Jie, L.; Wang, P.; Yamada, M.; Chang, Y.; and Mei, Q. 2016. Beyond ranking: Optimizing whole-page presentation. In *International Conference on Web Search and Data Mining*.

Wu, H.; and Wang, M. 2018. Variance Regularized Counterfactual Risk Minimization via Variational Divergence Minimization. In *International Conference on Machine Learning*.

Wydmuch, M.; Jasinska, K.; Kuznetsov, M.; Busa-Fekete, R.; and Dembczynski, K. 2018. A no-regret generalization of hierarchical softmax to extreme multi-label classification. In *Advances in Neural Information Processing Systems*.

Yen, I. E.-H.; Huang, X.; Dai, W.; Ravikumar, P.; Dhillon, I.; and Xing, E. 2017. Ppdsparse: A parallel primal-dual sparse method for extreme classification. In *International Conference on Knowledge Discovery and Data Mining*.

Yen, I. E.-H.; Huang, X.; Ravikumar, P.; Zhong, K.; and Dhillon, I. 2016. Pd-sparse: A primal and dual sparse approach to extreme multiclass and multilabel classification. In *International Conference on Machine Learning*.

You, R.; Zhang, Z.; Dai, S.; and Zhu, S. 2019a. HAXMLNet: Hierarchical Attention Network for Extreme Multi-Label Text Classification. *arXiv* .

You, R.; Zhang, Z.; Wang, Z.; Dai, S.; Mamitsuka, H.; and Zhu, S. 2019b. AttentionXML: Label Tree-based Attention-Aware Deep Model for High-Performance Extreme Multi-Label Text Classification. In *Advances in Neural Information Processing Systems*.

Zubiaga, A. 2012. Enhancing navigation on Wikipedia with social tags. *arXiv* .