Bayesian Counterfactual Risk Minimization

Ben London [1] **Ted Sandler** [1]

Abstract

We present a Bayesian view of counterfactual risk minimization (CRM) for offline learning from logged bandit feedback. Using PAC-Bayesian analysis, we derive a new generalization bound for the truncated inverse propensity score estimator. We apply the bound to a class of Bayesian policies, which motivates a novel, potentially data-dependent, regularization technique for CRM. Experimental results indicate that this technique outperforms standard L_2 regularization, and that it is competitive with variance regularization while being both simpler to implement and more computationally efficient.

1. Introduction

In industrial applications of machine learning, model development is typically an iterative process, involving multiple trials of offline training and online experimentation. For example, a content streaming service might explore various recommendation strategies in a series of A/B tests. The data that is generated by this process—e.g., impression and interaction logs—can be used to augment training data and further refine a model. However, learning from logged interactions poses two fundamental challenges: (1) the feedback obtained from interaction is always incomplete, since one only observes responses (usually referred to as *rewards*) for actions that were taken; (2) the distribution of observations is inherently biased by the *policy* that determined which action to take in each context.

This problem of learning from logged data has been studied under various names by various authors (Strehl et al., 2010; Dudík et al., 2011; Bottou et al., 2013; Swaminathan and Joachims, 2015a). We adopt the moniker *counterfactual risk minimization* (CRM), introduced by Swaminathan and Joachims (2015a), though it is also known as *offline policy optimization* in the reinforcement learning literature.

[1]Amazon, Seattle, WA, USA. Correspondence to: Ben London <blondon@amazon.com>.

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

The goal of CRM is to learn a policy from data that was logged by a previous policy so as to maximize expected reward (alternatively, minimize risk) over draws of future contexts. Using an analysis based on Bennett's inequality, Swaminathan and Joachims derived an upper bound on the risk of a stochastic policy,[1] which motivated learning with variance-based regularization.

In this work, we study CRM from a Bayesian perspective, in which one's uncertainty over actions becomes uncertainty over hypotheses. We view a stochastic policy as a distribution over hypotheses, each of which is a mapping from contexts to actions. Our work bridges the gap between CRM, which has until now been approached from the frequentist perspective, and Bayesian methods, which are often used to balance exploration and exploitation in contextual bandit problems (Chapelle and Li, 2011).

Using a PAC-Bayesian analysis, we prove an upper bound on the risk of a Bayesian policy trained on logged data. We then apply this bound to a class of Bayesian policies based on the mixed logit model. This analysis suggests an intuitive regularization strategy for Bayesian CRM based on the L_2 distance from the logging policy's parameters. Our *logging policy regularization* (LPR) is effectively similar to variance regularization, but simpler to implement and more computationally efficient. We derive two Bayesian CRM objectives based on LPR, one of which is convex. We also consider the scenario in which the logging policy is unknown. In this case, we propose a two-step procedure to learn the logging policy, and then use the learned parameters to regularize training a new policy. We prove a corresponding risk bound for this setting using a distribution-dependent prior.

We end with an empirical study of our theoretical results. First, we show that LPR outperforms standard L_2 regularization whenever the logging policy is better than a uniform distribution. Second, we show that LPR is competitive with variance regularization, and even outperforms it on certain problems. Finally, we demonstrate that it is indeed possible to learn the logging policy for LPR with negligible impact on performance. These findings establish LPR as a simple, effective method for Bayesian CRM.

[1]In a similar vein, Strehl et al. (2010) proved a lower bound on the expected reward of a deterministic policy.

Note: *All proofs are deferred to the supplemental material.*

2. Preliminaries

Let $\mathcal{X}$ denote a set of *contexts*, and $\mathcal{A}$ denote a finite set of k discrete *actions*. We are interested in finding a *stochastic policy*, $\pi : \mathcal{X} \to \Delta_k$, which maps $\mathcal{X}$ to the probability simplex on k vertices, denoted Δ_k; in other words, π defines a conditional probability distribution over actions given contexts, from which we can sample actions. For a given context, $x \in \mathcal{X}$, we denote the conditional distribution on $\mathcal{A}$ by $\pi(x)$, and the probability mass of a particular action, $a \in \mathcal{A}$, by $\pi(a \,|\, x)$.

Each action is associated with a stochastic, contextual *reward*, given by an unknown function, $\rho : \mathcal{X} \times \mathcal{A} \to [0, 1]$, which we assume is bounded. When an action is played in response to a context, we only observe the reward for said action. This type of incomplete feedback is commonly referred to as *bandit feedback*. We assume a stationary distribution, $\mathbb{D}$, over contexts and reward functions. Our goal will be to find a policy that maximizes the expected reward over draws of $(x, \rho) \sim \mathbb{D}$ and $a \sim \pi(x)$; or, put differently, one that minimizes the *risk*,

$$R(\pi) \triangleq 1 - \mathop{\mathbb{E}}_{(x,\rho)\sim\mathbb{D}} \mathop{\mathbb{E}}_{a\sim\pi(x)} \left[\rho(x, a)\right].$$

We assume that we have access to a dataset of logged observations (i.e., examples), $S \triangleq (x_i, a_i, p_i, r_i)_{i=1}^n$, where (x_i, ρ) were sampled from $\mathbb{D}$; action a_i was sampled with probability $p_i \triangleq \pi_0(a_i \,|\, x_i)$ from a fixed *logging policy*, π_0; and reward $r_i \triangleq \rho(x_i, a_i)$ was observed. The distribution of S, which we denote by $(\mathbb{D} \times \pi_0)^n$, is biased by the logging policy, since we only observe rewards for actions that were sampled from its distribution. Nonetheless, if π_0 has full support, we can obtain an unbiased estimate of $R(\pi)$ by scaling each reward by its *inverse propensity score* (IPS) (Rosenbaum and Rubin, 1983), p_i^{-1}, which yields the *IPS estimator*,

$$\hat{R}(\pi, S) \triangleq 1 - \frac{1}{n} \sum_{i=1}^{n} r_i \, \frac{\pi(a_i \,|\, x_i)}{p_i}.$$

Unfortunately, IPS can have very high variance. This issue can be mitigated by *truncating* (or *clipping*) p_i to the interval $[\tau, 1]$ (as proposed in (Strehl et al., 2010)), yielding

$$\hat{R}_\tau(\pi, S) \triangleq 1 - \frac{1}{n} \sum_{i=1}^{n} r_i \, \frac{\pi(a_i \,|\, x_i)}{\max\{p_i, \tau\}},$$

which we will sometimes refer to as the *empirical risk*. This estimator reduces variance, at the cost of adding bias. However, since $\max\{p_i, \tau\} \geq p_i$, we have that $\hat{R}_\tau(\pi, S) \geq \hat{R}(\pi, S)$, which implies

$$\mathop{\mathbb{E}}_{S\sim(\mathbb{D}\times\pi_0)^n} \left[\hat{R}_\tau(\pi, S)\right] \geq \mathop{\mathbb{E}}_{S\sim(\mathbb{D}\times\pi_0)^n} \left[\hat{R}(\pi, S)\right] = R(\pi).$$

Thus, if $\hat{R}_\tau(\pi, S)$ concentrates, then by minimizing it, we minimize a probabilistic upper bound on the risk.

Remark 1. There are other estimators we can consider. For instance, we could truncate the ratio of the policy and the logging policy, $\min\{\pi(a_i \,|\, x_i)/p_i, \tau^{-1}\}$ (Ionides, 2008; Swaminathan and Joachims, 2015a). However, this form of truncation is incompatible with our subsequent analysis because the policy is inside the $\min$ operator. Avoiding truncation altogether, we could use the *self-normalizing* estimator (Swaminathan and Joachims, 2015b), but this is also incompatible, since the estimator does not decompose as a sum of i.i.d. random variables. Finally, we note that our theory *does* apply, with small modifications, to the *doubly-robust* estimator (Dudík et al., 2011).

2.1. Counterfactual Risk Minimization

Our work is heavily influenced by Swaminathan and Joachims (2015a), who coined the term *counterfactual risk minimization* (CRM) to refer to the problem of learning a policy from logged bandit feedback by minimizing an upper bound on the risk. Their bound is a function of the truncated[2] IPS estimator, the sample variance of the truncated IPS-weighted rewards under the policy, $\hat{V}_\tau(\pi, S)$, and a measure of the complexity, $\mathcal{C} : \Pi \to \mathbb{R}_+$, of the class of policies being considered, $\Pi \subseteq \{\pi : \mathcal{X} \to \Delta_k\}$.

$$R(\pi) \leq \hat{R}_\tau(\pi, S) + \mathrm{O}\left(\sqrt{\tfrac{\hat{V}_\tau(\pi,S)\,\mathcal{C}(\Pi)}{n}} + \tfrac{\mathcal{C}(\Pi)}{n}\right). \quad (1)$$

When $\hat{V}_\tau(\pi, S)$ is sufficiently small, the bound's dominating term is $\mathrm{O}(n^{-1})$, which is the so-called "fast" learning rate. This motivates a variance-regularized learning objective,

$$\arg\min_{\pi\in\Pi} \; \hat{R}_\tau(\pi, S) + \lambda\sqrt{\tfrac{\hat{V}_\tau(\pi,S)}{n}}, \quad (2)$$

for a regularization parameter, $\lambda > 0$. Swaminathan and Joachims propose a majorization-minimization algorithm—named *policy optimization for exponential models* (POEM)—to solve this optimization.

3. PAC-Bayesian Analysis

In this work, we view CRM from a Bayesian perspective. We consider stochastic policies whose action distributions are induced by distributions over *hypotheses*. Instead of sampling directly from a distribution on the action set, we sample from a distribution on a *hypothesis space*, $\mathcal{H} \subseteq \{h : \mathcal{X} \to \mathcal{A}\}$, in which each element is a deterministic mapping from contexts to actions.[3] As such, for a distribution, $\mathbb{Q}$, on $\mathcal{H}$, the probability of an action, $a \in \mathcal{A}$,

[2] Though Swaminathan and Joachims used a different form of truncation, their results nonetheless hold for our truncation.

[3] This view of stochastic policies was also used by Seldin et al. (2011) to analyze contextual bandits with PAC-Bayes.

given a context, $x \in \mathcal{X}$, is the probability that a random hypothesis, $h \sim \mathbb{Q}$, maps x to a; that is,

$$\pi_{\mathbb{Q}}(a \,|\, x) \triangleq \Pr_{h \sim \mathbb{Q}} \{h(x) = a\} = \mathop{\mathbb{E}}_{h \sim \mathbb{Q}} [\mathbb{1}\{h(x) = a\}] \,. \quad (3)$$

Usually, the hypothesis space consists of functions of a certain parametric form, so the distribution is actually over the parameter values. We analyze one such class in Section 4.

To analyze Bayesian policies, we use the *PAC-Bayesian* framework (also known as simply *PAC-Bayes*) . The PAC-Bayesian learning paradigm proceeds as follows: first, we fix a hypothesis space, $\mathcal{H}$, and a *prior* distribution, $\mathbb{P}$, on $\mathcal{H}$; then, we receive some data, S, drawn from a fixed distribution; given S, we learn a *posterior*, $\mathbb{Q}$, on $\mathcal{H}$, from which we can sample hypotheses to classify new instances. In our PAC-Bayesian formulation of CRM, the learned posterior becomes our stochastic policy (Equation 3). Given a context, $x \in \mathcal{X}$, we sample an action by sampling $h \sim \mathbb{Q}$ (independent of x) and returning $h(x)$. (In PAC-Bayesian terminology, this procedure is called the *Gibbs classifier*.)

Remark 2. Instead of sampling actions via a posterior over hypotheses, we could equivalently sample policies from a posterior over policies, $\{\pi : \mathcal{X} \to \Delta_k\}$, then sample actions from said policies. The Bayesian policy would then be the expected policy, $\bar{\pi}_{\mathbb{Q}}(a \,|\, x) \triangleq \mathbb{E}_{\pi \sim \mathbb{Q}}[\pi(a \,|\, x)]$. That said, it is more traditional in PAC-Bayes—and perhaps more flexible—to think in terms of the Gibbs classifier, which directly maps contexts to actions.

It is important to note that the choice of prior cannot depend on the training data; however, *the prior can generate the data*. Indeed, we can generate S by sampling $(x_i, \rho) \sim \mathbb{D}$, $h \sim \mathbb{P}$ and logging $(x_i, h(x_i), \pi_0(h(x_i) \,|\, x_i), \rho(x_i, h(x_i)))$, for $i = 1, \ldots, n$. Thus, in the PAC-Bayesian formulation of CRM, *the prior can be the logging policy*. We elaborate on this in Section 4.

3.1. Risk Bounds

The heart of our analysis is an application of the PAC-Bayesian theorem—a generalization bound for Bayesian learning—to upper-bound the risk. The particular PAC-Bayesian bound we use is by McAllester (2003). Omitting details (in the interest of space), the bound states that, for any fixed prior, $\mathbb{P}$ on $\mathcal{H}$, with probability at least $1 - \delta$ over draws of the data, S, all posteriors, $\mathbb{Q}$, on $\mathcal{H}$, satisfy

$$R(\mathbb{Q}) \leq \hat{R}(\mathbb{Q}, S) + \tilde{\mathrm{O}}\left(\sqrt{\frac{\hat{R}(\mathbb{Q}, S) D_{\mathrm{KL}}(\mathbb{Q} \| \mathbb{P})}{n}} + \frac{D_{\mathrm{KL}}(\mathbb{Q} \| \mathbb{P})}{n}\right).$$

where $R(\mathbb{Q})$ and $\hat{R}(\mathbb{Q}, S)$ are the risk and empirical risk, respectively. The hallmark of a PAC-Bayesian bound is the KL divergence from the prior to the posterior. This can be interpreted as a complexity measure, similar to the VC dimension, covering number or Rademacher complexity (Mohri et al., 2012). The divergence penalizes posteriors that stray from the prior, effectively penalizing overfitting.

One attractive property of McAllester's bound is that, if the empirical risk is sufficiently small, then the generalization error, $R(\mathbb{Q}) - \hat{R}(\mathbb{Q}, S)$, is $\mathrm{O}(n^{-1})$. Thus, the bound captures both realizable and non-realizable learning problems.

To apply the PAC-Bayesian theorem to CRM, we design a loss function that allows us to state the risk of a policy in terms of the risk of the Gibbs classifier. This yields the following risk bound.

Theorem 1. *Let $\mathcal{H} \subseteq \{h : \mathcal{X} \to \mathcal{A}\}$ denote a hypothesis space mapping contexts to actions. For any $n \geq 1$, $\delta \in (0, 1)$, $\tau \in (0, 1)$ and fixed prior, $\mathbb{P}$, on $\mathcal{H}$, with probability at least $1 - \delta$ over draws of $S \sim (\mathbb{D} \times \pi_0)^n$, the following holds simultaneously for all posteriors, $\mathbb{Q}$, on $\mathcal{H}$:*

$$\begin{aligned} R(\pi_{\mathbb{Q}}) \leq\; & \hat{R}_\tau(\pi_{\mathbb{Q}}, S) \\ & + \sqrt{\frac{2\left(\hat{R}_\tau(\pi_{\mathbb{Q}}, S) - 1 + \frac{1}{\tau}\right)\left(D_{\mathrm{KL}}(\mathbb{Q} \| \mathbb{P}) + \ln \frac{n}{\delta}\right)}{\tau\,(n-1)}} \\ & + \frac{2\left(D_{\mathrm{KL}}(\mathbb{Q} \| \mathbb{P}) + \ln \frac{n}{\delta}\right)}{\tau\,(n-1)}. \end{aligned} \quad (4)$$

It is important to note that the truncated IPS estimator, $\hat{R}_\tau$, can be negative, achieving its minimum at $1 - \tau^{-1}$. This means that when $\hat{R}_\tau$ is minimized, the middle $\mathrm{O}(n^{-1/2})$ term disappears and the $\mathrm{O}(n^{-1})$ term dominates the bound, yielding the "fast" learning rate. That said, our bound may not be as tight as Swaminathan and Joachims' (Equation 1), since the variance is sometimes smaller than the mean. To achieve a similar rate, we could perhaps use Seldin et al.'s (2012) PAC-Bayesian Bernstein bound.

Though Theorem 1 assumes that the truncation parameter, τ, is fixed *a priori*, we can derive a risk bound (given in the supplemental material) that holds for all τ simultaneously—meaning, τ can be data-dependent, such as the 10^{th} percentile of the logged propensities.

Theorem 1 has an intriguing interpretation when the prior is defined as the logging policy. In this case, one can minimize an upper bound on the risk by minimizing the empirical risk while keeping the learned policy close to the logging policy. We explore this idea, and its relationship to variance regularization, in the next section.

4. Mixed Logit Models

We will apply our PAC-Bayesian analysis to the following class of stochastic policies. We first define a hypothesis space, $\mathcal{H} \triangleq \{h_{w,\gamma} : w \in \mathbb{R}^d, \gamma \in \mathbb{R}^k\}$, of functions of the form

$$h_{w,\gamma}(x) \triangleq \mathop{\arg\max}_{a \in \mathcal{A}} w \cdot \phi(x, a) + \gamma_a, \quad (5)$$

where $\phi(x, a) \in \mathbb{R}^d$ outputs features of the context and action, subject to $\sup_{x\in\mathcal{X},a\in\mathcal{A}} \|\phi(x, a)\| \leq B$. If each γ_a is sampled from a *standard Gumbel* distribution, $\mathrm{Gum}(0, 1)$ (location 0, scale 1), then $h_{w,\gamma}(x)$ produces a sample from a *softmax* model,

$$\varsigma_w(a \,|\, x) \triangleq \frac{\exp(w \cdot \phi(x, a))}{\sum_{a'\in\mathcal{A}} \exp(w \cdot \phi(x, a'))}. \tag{6}$$

Further, if w is normally distributed, then $h_{w,\gamma}(x)$ has a *logistic-normal* distribution (Aitchison and Shen, 1980).

We define the posterior, $\mathbb{Q}$, as a Gaussian over softmax parameters, $w \sim \mathcal{N}(\mu, \sigma^2\mathbf{I})$, for some learned $\mu \in \mathbb{R}^d$ and $\sigma^2 \in (0, \infty)$, with standard Gumbel perturbations, $\gamma \sim \mathrm{Gum}(0, 1)^k$. As such, we have that

$$\pi_{\mathbb{Q}}(a \,|\, x) = \mathop{\mathbb{E}}_{w,\gamma}\left[\mathbb{1}\{h_{w,\gamma}(x) = a\}\right] = \mathop{\mathbb{E}}_{w\sim\mathcal{N}(\mu,\sigma^2\mathbf{I})}\left[\varsigma_w(a \,|\, x)\right]. \tag{7}$$

This model is alternately referred to as a *mixed logit* or *random parameter logit*.

We can define the prior in any way that seems reasonable—without access to training data, of course. In the absence of any prior knowledge, a logical choice of prior is the standard (zero mean, unit variance) multivariate normal distribution, with standard Gumbel perturbations. This prior corresponds to a Bayesian policy that takes uniformly random actions, and motivates standard L_2 regularization of μ. However, we know that the data was generated by the logging policy, and this knowledge motivates a different kind of prior (hence, regularizer). If the logging policy performs better than a uniform action distribution—which we can verify empirically, using IPS with the logs—then it makes sense to define the prior in terms of the logging policy.

Let us assume that the logging policy is known (we relax this assumption in Section 5) and has a softmax form (Equation 6), with parameters $\mu_0 \in \mathbb{R}^d$. We define the prior, $\mathbb{P}$, as an isotropic Gaussian centered at the logging policy's parameters, $w \sim \mathcal{N}(\mu_0, \sigma_0^2\mathbf{I})$, for some predetermined $\sigma_0^2 \in (0, \infty)$, with standard Gumbel perturbations, $\gamma \sim \mathrm{Gum}(0, 1)^k$. This prior encodes a belief that the logging policy, while not perfect, is a good starting point. Using the logging policy to define the prior does not violate the PAC-Bayes paradigm, since the logging policy is fixed before generating the training data. The Bayesian policy induced by this prior may not correspond to the actual logging policy, but we can define the prior any way we want.

Remark 3. We used isotropic covariances for the prior and posterior in order to simplify our analysis and presentation. That said, it is possible to use more complex covariances.

4.1. Bounding the KL Divergence

The KL divergence between the above prior and posterior constructions motivates an interesting regularizer for CRM. To derive it, we upper-bound the KL divergence by a function of the model parameters.

Lemma 1. *For distributions* $\mathbb{P} \triangleq \mathcal{N}(\mu_0, \sigma_0^2\mathbf{I}) \times \mathrm{Gum}(0, 1)^k$ *and* $\mathbb{Q} \triangleq \mathcal{N}(\mu, \sigma^2\mathbf{I}) \times \mathrm{Gum}(0, 1)^k$*, with* $\mu_0, \mu \in \mathbb{R}^d$ *and* $0 < \sigma^2 \leq \sigma_0^2 < \infty$*,*

$$D_{\mathrm{KL}}(\mathbb{Q}\|\mathbb{P}) \leq \frac{\|\mu - \mu_0\|^2}{2\sigma_0^2} + \frac{d}{2}\ln\frac{\sigma_0^2}{\sigma^2}. \tag{8}$$

One implication of Lemma 1, captured by the term $\|\mu - \mu_0\|^2$, is that, to generalize, the learned policy's parameters should stay close to the logging policy's parameters. This intuition concurs with Swaminathan and Joachims's (2015a) variance regularization, since one way to reduce the variance is to not stray too far from the logging policy. It is also reminiscent of *trust region policy optimization* (Schulman et al., 2015), a reinforcement learning algorithm in which each update to the policy's action distribution is constrained to not diverge too much from the current one.[4] Implementing Lemma 1's guideline in practice requires a simple modification to the usual L_2 regularization: instead of $\lambda\|\mu\|^2$ (where $\lambda > 0$ controls the amount of regularization), use $\lambda\|\mu - \mu_0\|^2$. Of course, this assumes that the logging policy's parameters, μ_0, are known; we address the scenario in which the logging policy is unknown in Section 5.

4.2. Approximating the Action Probabilities

In practice, computing the posterior action probabilities (Equation 7) of a mixed logit model is difficult, since there is no analytical expression for the mean of the logistic-normal distribution (Aitchison and Shen, 1980). It is therefore difficult to log propensities, or to compute the IPS estimator, which is a function of the learned and logged probabilities. Since it is easy to sample from a mixed logit, we can use Monte Carlo methods to estimate the probabilities. Alternatively, we can bound the probabilities by a function of the mean parameters, μ.

Lemma 2. *If* $\sup_{x\in\mathcal{X},a\in\mathcal{A}} \|\phi(x, a)\| \leq B$*, then*

$$\varsigma_\mu(a \,|\, x)\, e^{-\frac{\sigma^2B^2}{2}} \;\leq\; \pi_{\mathbb{Q}}(a \,|\, x) \;\leq\; \varsigma_\mu(a \,|\, x)\, e^{2\sigma^2B^2}.$$

By Lemma 2, the softmax probabilities induced by the mean parameters provide lower and upper bounds on the probabilities of the mixed logit model. The bounds tighten as the variance, σ^2, becomes smaller. For instance, if $\sigma^2 = \mathrm{O}(n^{-1})$, then $\pi_{\mathbb{Q}}(a \,|\, x) \to \varsigma_\mu(a \,|\, x)$ as $n \to \infty$.

During learning, we can use the lower bound of the learned probabilities to upper-bound the IPS estimator. We over-

[4] Interestingly, via Fenchel duality and Cauchy-Schwarz, a bound like Equation 8 holds for the KL divergence between softmax action distributions: $D_{\mathrm{KL}}(\varsigma_\mu(x)\|\varsigma_{\mu_0}(x)) \leq \mathrm{O}(\|\mu - \mu_0\|)$.

load our previous notation to define a new estimator,

$$\hat{R}_\tau(\mu, \sigma^2, S) \triangleq 1 - \frac{\exp(-\frac{\sigma^2 B^2}{2})}{n} \sum_{i=1}^{n} r_i \frac{\varsigma_\mu(a_i \mid x_i)}{\max\{p_i, \tau\}}.$$

This estimator is biased, but the bias decreases with σ^2. Importantly, $\hat{R}_\tau(\mu, \sigma^2, S)$ is easy to compute (assuming the action set is not too large), since it avoids the logistic-normal integral.

When the learned posterior is deployed, we can log the upper bound of the propensities, so that future training with the logged data has an upper bound on the IPS estimator.

4.3. Bayesian CRM for Mixed Logit Models

We now present a risk bound for the Bayesian policy, $\pi_{\mathbb{Q}}$, using the softmax policy, ς_μ, on the mean parameters, μ.

Theorem 2. *Let $\mathcal{H}$ denote the hypothesis space defined in Equation 5, and let $\pi_{\mathbb{Q}}$ denote the mixed logit policy defined in Equation 7. For any $n \geq 1$, $\delta \in (0,1)$, $\tau \in (0,1)$, $\mu_0 \in \mathbb{R}^d$ and $\sigma_0^2 \in (0, \infty)$, with probability at least $1-\delta$ over draws of $S \sim (\mathbb{D} \times \pi_0)^n$, the following holds simultaneously for all $\mu \in \mathbb{R}^d$ and $\sigma^2 \in (0, \sigma_0^2]$:*

$$R(\pi_{\mathbb{Q}}) \leq \hat{R}_\tau(\mu, \sigma^2, S) + \sqrt{\frac{\left(\hat{R}_\tau(\mu, \sigma^2, S) - 1 + \frac{1}{\tau}\right)\left(\Gamma(\mu_0, \sigma_0^2, \mu, \sigma^2) + 2\ln\frac{n}{\delta}\right)}{\tau\,(n-1)}} + \frac{\Gamma(\mu_0, \sigma_0^2, \mu, \sigma^2) + 2\ln\frac{n}{\delta}}{\tau\,(n-1)}, \tag{9}$$

$$\textit{where} \quad \Gamma(\mu_0, \sigma_0^2, \mu, \sigma^2) \triangleq \frac{\|\mu - \mu_0\|^2}{\sigma_0^2} + d\ln\frac{\sigma_0^2}{\sigma^2}. \tag{10}$$

Theorem 2 provides an upper bound on the risk that can be computed with training data. Moreover, the bound is differentiable and smooth, meaning it can be optimized using gradient-based methods. This motivates a new regularized learning objective for Bayesian CRM.

Proposition 1. *The following optimization minimizes an upper bound on Equation 9:*

$$\operatorname*{arg\,min}_{\mu \in \mathbb{R}^d,\, \sigma^2 \in (0, \sigma_0^2]} \hat{R}_\tau(\mu, \sigma^2, S) + \frac{\|\mu - \mu_0\|^2}{\sigma_0^2\,\tau\,(n-1)} - \frac{d\ln\sigma^2}{\tau\,(n-1)}. \tag{11}$$

Equation 11 is unfortunately non-convex. However, we can upper-bound $\hat{R}_\tau(\mu, \sigma^2, S)$ to obtain an objective that is differentiable, smooth and *convex*.

Proposition 2. *The following convex optimization minimizes an upper bound on Equation 9:*

$$\operatorname*{arg\,min}_{\mu \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} -\frac{r_i \ln \varsigma_\mu(a_i \mid x_i)}{\max\{p_i, \tau\}} + \frac{\|\mu - \mu_0\|^2}{\sigma_0^2\,\tau\,(n-1)}, \tag{12}$$

with $\sigma^2 \triangleq \min\left\{\frac{2d}{B^2\tau(n-1)}\left(\frac{1}{n}\sum_{i=1}^{n}\frac{r_i}{\max\{p_i,\tau\}}\right)^{-1}, \sigma_0^2\right\}$.

Conveniently, Equation 12 is equivalent to a weighted softmax regression with a modified L_2 regularizer. This optimization can be solved using standard methods, with guaranteed convergence to a global optimum. Moreover, by decoupling the optimizations of μ and σ^2 in the upper bound (refer to the proof for details), we can solve for the optimal σ^2 in closed form.

Equation 12 also has a connection to *policy gradient* methods (Sutton et al., 2000). We discuss this connection in the supplemental material.

In practice, one usually tunes the amount of regularization to optimize the empirical risk on a held-out validation dataset. By Propositions 1 and 2, this is equivalent to tuning the variance of the prior, σ_0^2. Though μ_0 could in theory be any fixed vector, the case when it is the parameters of the logging policy corresponds to an interesting regularizer. This regularizer instructs the learning algorithm to keep the learned policy close to the logging policy, which effectively reduces the variance of the estimator.

Using Theorem 2, we can examine how the parameters σ_0^2 and σ^2 affect the bias-variance trade-off. Recall from Lemma 2 that higher values of σ^2 increase the bias of the estimator, $\hat{R}_\tau(\mu, \sigma^2, S)$. To reduce this bias, we want σ^2 to be small; e.g., $\sigma^2 = \Theta(n^{-1})$ results in negligible bias. However, if we also have $\sigma_0^2 = \Theta(1)$, then $\Gamma(\mu_0, \sigma_0^2, \mu, \sigma^2)$—which can be interpreted as the variance of the estimator—has a term, $d\ln\frac{\sigma_0^2}{\sigma^2} = \mathrm{O}(d\ln n)$, that depends linearly on the number of features, d. When d is large, this term can dominate the risk bound. The dependence on d is eliminated when $\sigma_0^2 = \sigma^2$; but if $\sigma_0^2 = \Theta(n^{-1})$, then $\Gamma(\mu_0, \sigma_0^2, \mu, \sigma^2) = \mathrm{O}(\|\mu - \mu_0\|^2 n)$, which makes the risk bound vacuous.

5. When the Logging Policy Is Unknown

In Section 4, we assumed that the logging policy was known and used it to construct a prior. However, there may be settings in which the logging policy is unknown. We can nonetheless construct a prior that approximates the logging policy by learning from its logged actions.

At first, this idea may sound counterintuitive. After all, the prior is supposed to be fixed before drawing the training data. However, the expected value of a function of the data is constant with respect to any realization of the data. Thus, the expected estimator of the logging policy is independent of the data, and can serve as a valid prior. We then just need to show that the estimator concentrates around its mean. This type of analysis was introduced by Catoni (2007) and later developed by Lever et al. (2010), among others.

Overloading our previous notation, let $L : \mathbb{R}^d \times \mathcal{X} \times \mathcal{A} \to \mathbb{R}_+$ denote a loss function that measures the fit of param-

eters $w \in \mathbb{R}^d$, given context $x \in \mathcal{X}$ and action $a \in \mathcal{A}$. We will assume that L is both convex and β-Lipschitz with respect to w. This assumption is satisfied by, e.g., the negative log-likelihood. For a dataset, $S \sim (\mathbb{D} \times \pi_0)^n$, containing logged contexts and actions, let

$$F(w, S) \triangleq \frac{1}{n} \sum_{i=1}^{n} L(w, x_i, a_i) + \lambda \|w\|^2 \tag{13}$$

denote a regularized objective; let

$$\hat{\mu}_0(S) \triangleq \arg\min_{w \in \mathbb{R}^d} F(w, S) \tag{14}$$

denote its minimizer; and let $\bar{\mu}_0 \triangleq \mathbb{E}_{S \sim (\mathbb{D} \times \pi_0)^n}[\hat{\mu}_0(S)]$ denote the expected minimizer. Since $\bar{\mu}_0$ is a constant, it is independent of any realization of S. We can therefore construct a Gaussian prior around $\bar{\mu}_0$, which makes the KL divergence proportional to $\|\mu - \bar{\mu}_0\|^2$.

Since F is strongly convex, its minimizer exhibits *uniform algorithmic stability*; meaning, it is robust to perturbations of the training data. Due to this property, the random variable $\hat{\mu}_0(S)$ concentrates around its mean, $\bar{\mu}_0$ (Liu et al., 2017). Thus, with high probability, $\|\hat{\mu}_0(S) - \bar{\mu}_0\|$ is small, and $\|\mu - \bar{\mu}_0\|$ is approximately $\|\mu - \hat{\mu}_0(S)\|$.

Theorem 3. *Let $\mathcal{H}$ denote the hypothesis space defined in Equation 5, and let $\pi_{\mathbb{Q}}$ denote the mixed logit policy defined in Equation 7. Let $\hat{\mu}_0(S)$ denote the minimizer defined in Equation 14, for a convex, β-Lipschitz loss function. For any $n \geq 1$, $\delta \in (0, 1)$, $\tau \in (0, 1)$ and $\sigma_0^2 \in (0, \infty)$, with probability at least $1 - \delta$ over draws of $S \sim (\mathbb{D} \times \pi_0)^n$, the following holds for all $\mu \in \mathbb{R}^d$ and $\sigma^2 \in (0, \sigma_0^2]$:*

$$R(\pi_{\mathbb{Q}}) \leq \hat{R}_\tau(\mu, \sigma^2, S) + \sqrt{\frac{\left(\hat{R}_\tau(\mu, \sigma^2, S) - 1 + \frac{1}{\tau}\right)\left(\hat{\Gamma}(\hat{\mu}_0(S), \sigma_0^2, \mu, \sigma^2) + 2 \ln \frac{2n}{\delta}\right)}{\tau\,(n-1)}} + \frac{\hat{\Gamma}(\hat{\mu}_0(S), \sigma_0^2, \mu, \sigma^2) + 2 \ln \frac{2n}{\delta}}{\tau\,(n-1)}, \tag{15}$$

where

$$\hat{\Gamma}(\hat{\mu}_0(S), \sigma_0^2, \mu, \sigma^2) \triangleq \frac{\left(\|\mu - \hat{\mu}_0(S)\| + \frac{\beta}{\lambda}\sqrt{\frac{2 \ln \frac{4}{\delta}}{n}}\right)^2}{\sigma_0^2} + d \ln \frac{\sigma_0^2}{\sigma^2}.$$

It is straightforward to show that Propositions 1 and 2 hold for Theorem 3 with $\mu_0 \triangleq \hat{\mu}_0(S)$, which motivates a two-step learning procedure for Bayesian CRM: (1) using logged data, S, but ignoring rewards, solve Equation 14 to estimate softmax parameters, $\hat{\mu}_0(S)$, that approximate the logging policy; (2) using S again, including the rewards, solve Equation 11 or 12, with $\mu_0 \triangleq \hat{\mu}_0(S)$, to train a new mixed logit policy.

Remark 4. Throughout, we have assumed that the logged data includes the propensities, which enable IPS weighting. Given that we can learn to approximate the logging policy, it seems natural to use the learned propensities in the absence of the true propensities. In practice, this may work, though we do not provide any formal guarantees for it.

6. Experiments

Our Bayesian analysis of CRM suggests a new regularization technique, which we will henceforth refer to as *logging policy regularization* (LPR). Using the logging policy to construct a prior, we regularize by the squared distance between the (learned) logging policy's softmax parameters, μ_0, and the posterior mean, μ, over softmax parameters. In this section, we empirically verify the following claims: (1) LPR outperforms standard L_2 regularization whenever the logging policy outperforms a uniform action distribution; (2) LPR is competitive with variance regularization (i.e., POEM), and is also faster to optimize; (3) when the logging policy is unknown, we can estimate it from logged data, then use the estimator in LPR with little deterioration in performance.

We will use the class of mixed logit models from Section 4. For simplicity, we choose to only optimize the posterior mean, μ, assuming that the posterior variance, σ^2, is fixed to some small value, e.g., n^{-1}. This is inconsequential, since we will approximate the posterior action probabilities, $\pi_{\mathbb{Q}}(a \,|\, x)$, with a softmax of the mean parameters, $\varsigma_\mu(a \,|\, x)$. By Lemma 2, with small σ^2, this is a reasonable approximation. In a small departure from our analysis, we add an unregularized bias term for each action.

We evaluate two methods based on LPR. The first method, inspired by Proposition 1, combines LPR with the truncated IPS estimator:

$$\arg\min_{\mu \in \mathbb{R}^d} \hat{R}_\tau(\varsigma_\mu, S) + \lambda \|\mu - \mu_0\|^2, \tag{16}$$

where $\tau \in (0, 1)$ and $\lambda \geq 0$ are free parameters. We call this method IPS-LPR. The second method, inspired by Proposition 1, is a convex upper bound:

$$\arg\min_{\mu \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} -\frac{r_i \ln \varsigma_\mu(a_i \,|\, x_i)}{\max\{p_i, \tau\}} + \lambda \|\mu - \mu_0\|^2. \tag{17}$$

Since the first term is essentially a weighted negative log-likelihood, we call this method WNLL-LPR.

We compare the above methods to several baselines. The first baseline is IPS with standard L_2 regularization, which essentially replaces $\|\mu - \mu_0\|^2$ with $\|\mu\|^2$ in Equation 16. We call this baseline IPS-L2. The second baseline is POEM (Swaminathan and Joachims, 2015a), which solves the variance regularized objective in Equation 2 (with Π as the class of softmax policies with parameters $\mu \in \mathbb{R}^d$) using a majorization-minimization algorithm. We also test a variant of POEM that adds L_2 regularization, which we refer to as POEM-L2.

All methods require some form of IPS truncation. For IPS-L2, IPS-LPR and WNLL-LPR, we use $\max\{p_i, \tau\}^{-1}$; for POEM and POEM-L2, we use $\min\{\pi(a_i \,|\, x_i)/p_i, \tau^{-1}\}$,

per Swaminathan and Joachims's original formulation. In all experiments, we set $\tau \triangleq 0.01$.

Since all methods support stochastic first-order optimization, we use AdaGrad (Duchi et al., 2011) with mini-batches of 100 examples. We set the learning rate to 0.1 and the smoothing parameter to one, which we find necessary for numerical stability. Unless otherwise stated, we run training for 500 epochs, with random shuffling of the training data at each epoch. All model parameters are initialized to zero, and all runs of training are seeded such that every method receives the same sequence of examples.

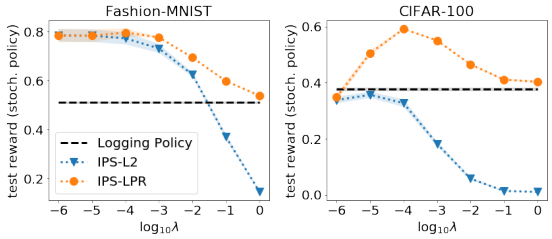
We report results on two benchmark image classification datasets: Fashion-MNIST (Xiao et al., 2017) and CIFAR-100 (Krizhevsky and Hinton, 2009). Fashion-MNIST consists of 70,000 (60,000 training; 10,000 testing) grayscale images from 10 categories of apparel and accessories. We extract features from each image by normalizing pixel intensities to $[0, 1]$ and flattening the (28×28)-pixel grid to a 784-dimensional vector. CIFAR-100 consists of 60,000 (50,000 training; 10,000 testing) color images from 100 general object categories. As this data is typically modeled with deep convolutional neural networks, we use transfer learning to extract features expressive enough to yield decent performance with the class of log-linear models described in Section 4. Specifically, we use the last hidden layer of a pre-trained ResNet-50 network (He et al., 2016), which was trained on ImageNet (Deng et al., 2009), to output 2048-dimensional features for CIFAR-100.

Following prior work, we use a standard supervised-to-bandit conversion to simulate logged bandit feedback (Beygelzimer and Langford, 2009). We start by randomly sampling 1,000 training examples (without replacement) to train a softmax logging policy using supervised learning. We then use the logging policy to sample a label (i.e., action) for each remaining training example. The reward is one if the sampled label matches the true label, and zero otherwise. We repeat this procedure 10 times, using 10 random splits of the training data, to generate 10 datasets of logged contexts, actions, propensities and rewards.

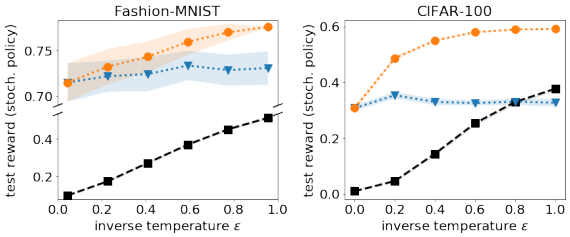
We compare methods along two metrics. Our primary metric is the expected reward under the stochastic policy, $\mathbb{E}_{a\sim\pi(x)}[\rho(x, a)]$, averaged over the testing data. Our secondary metric—which is not directly supported by our analysis, but is nonetheless of interest—is the reward of the deterministic *argmax policy*, $\rho(x, \arg\max_{a\in\mathcal{A}} \pi(a \mid x))$.

6.1. Logging Policy as Prior

We first investigate our claim that the logging policy is a better prior than a standard normal distribution, thus motivating LPR over L_2 regularization. For each simulated log dataset, we train new policies using IPS-L2 and IPS-LPR, with regularization parameter values $\lambda = 10^{-6}, 10^{-5}, \ldots, 1$. Figure 1a plots the expected test reward as a function of λ. The dotted black line indicates the performance of the logging policy. We find that IPS-LPR outperforms IPS-L2 for each value of λ; meaning, for any amount of regularization, IPS-LPR is always better. Further, while the performance of IPS-L2 degrades to that of a uniform action distribution as we over-regularize, the performance of IPS-LPR converges to that of the logging policy. This illustrates the natural intuition that a policy that acts better than random guessing is an informative prior.



(a) Varying the amount of regularization.



(b) Varying the quality of the logging policy.

Figure 1: L_2 regularization vs. LPR. Each line is the average of 10 trials, with shading to indicate the 95% confidence interval. Figure 1a plots the expected test reward as a function of λ. Figure 1b analyzes a spectrum of logging policies from the uniform action distribution ($\epsilon = 0$) to the trained distribution ($\epsilon = 1$).

An implication of this statement is that, as the logging policy's action distribution becomes more uniform, its efficacy as a prior should diminish. To verify this, we construct a sequence of logging policies that interpolate between the above logging policy and the uniform distribution, by multiplying the weights by an inverse-temperature parameter, $\epsilon = 0, 0.2, \ldots, 1$. We then generate log datasets for each logging policy, and train new policies using IPS-L2 and IPS-LPR, with $\lambda \triangleq 0.001$. As expected (see Figure 1b), the performance of IPS-LPR gradually converges to that of IPS-L2 as the logging policy converges to uniform.

One could also ask what happens when when the logging policy is *worse* than a uniform distribution. Indeed, though not shown here, we find that IPS-LPR performs worse than IPS-L2 in that scenario. However, one could reasonably argue that such a scenario is unlikely to occur in practice, since there is no point to deploying a logging policy that performs worse than a uniform distribution.

6.2. Comparison to POEM

As discussed earlier, LPR relates to variance regularization in that one way to minimize variance is to keep the new policy close to the logging policy. We are therefore prompted to investigate how LPR compares to variance regularization (i.e., POEM) in practice. In this experiment, our goal is to achieve the highest expected reward for each method on each log dataset, without looking at the testing data. Accordingly, we tune the regularization parameter, λ, using 5-fold cross-validation on each log dataset, with truncated IPS estimation of expected reward on the holdout set. For simplicity, we use grid search over powers of ten; $\lambda = 10^{-8}, \ldots, 10^{-3}$ for LPR and $\lambda = 10^{-3}, \ldots, 10^{2}$ for variance regularization. For POEM-L2, we tune the L_2 regularization parameter (in the same range as LPR) by fixing the variance regularization parameter to its optimal value. During parameter tuning, we limit training to 100 epochs. Once the parameter values have been selected, we train a new policy on the entire log dataset for 500 (Fashion-MNIST) or 1000 (CIFAR-100) epochs and evaluate it on the testing data.

Table 1 reports the results of this experiment. For completeness, we include results for all proposed methods and baselines, including the logging policy. On Fashion-MNIST, the variance regularization baselines (POEM and POEM-L2) achieve the highest expected reward, but the LPR methods (IPS-LPR and WNLL-LPR) are competitive. Indeed, the differences between these methods are not statistically significant according to a paired t-test with significance threshold 0.05. Meanwhile, all four significantly outperform IPS-L2 and the logging policy. Interestingly, WNLL-LPR performs best in terms of the argmax policy, perhaps owing to the fact that it is optimizing what is essentially a classification loss. Indeed, in classification problems with bandit feedback and binary rewards, the first term in Equation 17 is an unbiased estimator of the expected negative log-likelihood, which is a surrogate for the expected misclassification rate of the argmax policy.

The CIFAR-100 data presents a more challenging learning problem than Fashion-MNIST, since it has a much larger action set, and several times as many features. It is perhaps due to these difficulties that the baselines are unable to match the performance of the logging policy—which, despite being trained on far less data, is given full supervision. Meanwhile, both LPR methods outperform the logging policy by wide margins. We believe this is due to the fact that LPR is designed with incremental training in mind. The new policy is encouraged to stay close to the logging policy not just to hedge against overfitting, but also because the logging policy is assumed to be a good starting point.

It is worth comparing the running times of POEM and LPR. Recall that POEM is a majorization-minimization algorithm designed to enable stochastic optimization of a variance-regularized objective. At each epoch of training, POEM constructs an upper bound to the objective by processing all examples in the training data. This additional computation effectively doubles POEM's time complexity relative to the LPR methods, which only require one pass over the data per epoch. On Fashion-MNIST, we find that POEM is on average 25% slower than IPS-LPR.

Table 1: Test set rewards for Fashion-MNIST and CIFAR-100, averaged over 10 trials, with 5-fold cross-validation of regularization parameters at each trial.

	Fashion-MNIST		CIFAR-100	
Method	**stoch.**	**argmax**	**stoch.**	**argmax**
Logging Policy	0.5123	0.7099	0.3770	0.4797
IPS-L2	0.7778	0.7890	0.3475	0.3624
POEM	0.8060	0.8124	0.3338	0.3392
POEM-L2	0.8050	0.8126	0.3486	0.3641
IPS-LPR	0.7955	0.8154	0.5553	0.6134
WNLL-LPR	0.7978	0.8305	0.6143	0.6272
IPS-LLPR	0.7950	0.8153	0.5455	0.6077
WNLL-LLPR	0.7978	0.8305	0.6143	0.6272

6.3. Learning the Logging Policy

Per Section 5, when the logging policy is unknown, we can estimate its softmax parameters, μ_0, then use the estimate, $\hat{\mu}_0(S)$, in LPR. We now verify this claim empirically on Fashion-MNIST. Using the log datasets from the previous sections, we learn the logging policy with the regularized negative log-likelihood, $L(w, x, a) \triangleq -\ln \varsigma_w(a \,|\, x)$. We optimize this objective using 100 epochs of AdaGrad, with the same settings as the other experiments. We set the regularization parameter aggressively high, $\lambda \triangleq 0.01$, to ensure that the learned distribution does not become too peaked. Given $\hat{\mu}_0(S)$ for each log dataset, we then train new policies using IPS-LPR and WNLL-LPR, with the same λ values tuned in Section 6.2. The results of this experiment are given in the bottom section of Table 1, as methods IPS-LLPR and WNLL-LLPR (for *learned* LPR). The rewards are nearly identical to those when the logging policy is known, thus demonstrating that LPR does not require the actual logging policy in order to be effective.

7. Conclusion

We have presented a PAC-Bayesian analysis of counterfactual risk minimization, for learning Bayesian policies from logged bandit feedback. We applied our risk bound to a class of mixed logit policies, from which we derived two Bayesian CRM objectives based on logging policy regularization. Our empirical study indicated that LPR can achieve significant improvements over existing methods.

Acknowledgements

We thank Thorsten Joachims for thoughtful discussions and helpful feedback.

References

J. Aitchison and S. Shen. Logistic-normal distributions: Some properties and uses. *Biometrika*, 67(2):261–272, 1980.

A. Beygelzimer and J. Langford. The offset tree for learning with partial labels. In *Knowledge Discovery and Data Mining*, 2009.

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

O. Catoni. *PAC-Bayesian Supervised Classification: The Thermodynamics of Statistical Learning*, volume 56 of *Institute of Mathematical Statistics Lecture Notes – Monograph Series*. Institute of Mathematical Statistics, 2007.

O. Chapelle and L. Li. An empirical evaluation of Thompson sampling. In *Neural Information Processing Systems*, 2011.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F.-F. Li. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.

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

P. Germain, A. Lacasse, F. Laviolette, and M. Marchand. PAC-Bayesian learning of linear classifiers. In *International Conference on Machine Learning*, 2009.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

E. Ionides. Truncated importance sampling. *Journal of Computational and Graphical Statistics*, 17(2):295–311, 2008.

A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

J. Langford and J. Shawe-Taylor. PAC-Bayes and margins. In *Neural Information Processing Systems*, 2002.

G. Lever, F. Laviolette, and J. Shawe-Taylor. Distribution-dependent PAC-Bayes priors. In *Algorithmic Learning Theory*, 2010.

T. Liu, G. Lugosi, G. Neu, and D. Tao. Algorithmic stability and hypothesis complexity. In *International Conference on Machine Learning*, 2017.

D. McAllester. PAC-Bayesian model averaging. In *Computational Learning Theory*, 1999.

D. McAllester. Simplified PAC-Bayesian margin bounds. In *COLT*, pages 203–215, 2003.

M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of Machine Learning*. Adaptive computation and machine learning. MIT Press, 2012. ISBN 978-0-262-01825-8.

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

J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, 2015.

M. Seeger. PAC-Bayesian generalisation error bounds for Gaussian process classification. *Journal of Machine Learning Research*, 3:233–269, 2002.

Y. Seldin, P. Auer, F. Laviolette, J. Shawe-Taylor, and R. Ortner. PAC-Bayesian analysis of contextual bandits. In *Neural Information Processing Systems*, 2011.

Y. Seldin, F. Laviolette, N. Cesa-Bianchi, J. Shawe-Taylor, and Peter Auer. PAC-Bayesian inequalities for martingales. *IEEE Transactions on Information Theory*, 58 (12):7086–7093, 2012.

A. Strehl, J. Langford, L. Li, and S. Kakade. Learning from logged implicit exploration data. In *Neural Information Processing Systems*, 2010.

R. Sutton, D. McAllester, S. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Neural Information Processing Systems*, 2000.

A. Swaminathan and T. Joachims. Batch learning from logged bandit feedback through counterfactual risk minimization. *Journal of Machine Learning Research*, 16: 1731–1755, 2015a.

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

H. Xiao, K. Rasul, and R. Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017.