Counterfactual Risk Minimization

Adith Swaminathan
Cornell University
Ithaca, NY 14850
adith@cs.cornell.edu

Thorsten Joachims
Cornell University
Ithaca, NY 14850
tj@cs.cornell.edu

1. INTRODUCTION

We develop a learning principle and an efficient algorithm for batch learning from logged bandit feedback. Unlike in supervised learning, where the algorithm receives training examples \((x_i, y_i^*)\) with annotated correct labels \(y_i^*\); bandit feedback merely provides a cardinal reward \(\delta_i \in \mathbb{R}\) for the prediction \(y_i\) that the logging system made for context \(x_i\). Such bandit feedback is ubiquitous in online systems (e.g., observing a click \(\delta_i\) on ad \(y_i\) for query \(x_i\)), while “correct” labels (e.g., the best ad \(y_i^*\) for query \(x_i\)) are difficult to assess.

Our work builds upon recent approaches to the off-policy evaluation problem [3], [6], [7], [9], where we re-use data collected from the interaction logs of one bandit algorithm to evaluate another system. These approaches use counterfactual reasoning [3] to derive an unbiased estimate of the system’s performance. Our work centers around the insight that, to perform robust learning, it is not sufficient to have just an unbiased estimate of system performance. We must also reason about how the variances of these estimators differ across the hypothesis space, and pick the hypothesis with the tightest conservative bound on system performance.

We first derive generalization error bounds analogous to structural risk minimization [1] for a stochastic hypothesis family. The constructive nature of these bounds suggests a general principle – Counterfactual Risk Minimization (CRM) – for designing methods for batch learning from bandit feedback. Using the CRM principle, we derive a new learning algorithm – Policy Optimizer for Exponential Models (POEM) – for structured output prediction. We evaluate POEM on several multi-label classification problems and verify that its empirical performance supports the theory.

Existing approaches for batch learning from logged bandit feedback fall into two categories. The first approach reduces the problem to supervised learning, using techniques like cost weighted classification [10] or the Offset Tree algorithm [2] when the space of possible predictions is small. In contrast, our approach generalizes structured output prediction with exponential-sized prediction spaces.

The second approach uses propensity scoring [12], [13], [8] to derive unbiased estimators from the logged data. These estimators are used for a small set of candidate policies, and the best estimated candidate is picked via exhaustive search. In contrast, our approach can be optimized via gradient descent, over hypothesis families of infinite size. The current work focuses on the inverse propensity scoring estimator (which assumes we have a good model of the logging system [2], [9]), but the results we derive hold equally for doubly robust estimators (which are more efficient when we additionally have a good model of user feedback [5]). In the current work, we concentrate on the case where the logging system was a stationary, stochastic policy. Techniques like exploration scavenging [6] and bootstrapping [10] allow us to perform counterfactual evaluation even when the logging system was deterministic or adaptive.

A full version of this abstract can be found on arXiv [14].

2. METHOD

To formalize the problem of batch learning from bandit feedback, consider a structured prediction problem that takes as input \(x\in\mathcal{X}\) and outputs a prediction \(y\in\mathcal{Y}\). In multi-label document classification, \(x\) could be a news article and \(y\) a bitvector indicating the labels assigned to this article. The inputs are assumed drawn i.i.d. from a fixed but unknown distribution \(Pr(X)\).

Consider the hypothesis space \(\mathcal{H}\) of stochastic policies. A hypothesis \(h(\mathcal{Y} | x)\in\mathcal{H}\) defines a probability distribution over the output space \(\mathcal{Y}\), and the hypothesis makes predictions by sampling, \(y\sim h(\mathcal{Y} | x)\). Note that stochastic policy families include deterministic rules. We denote \(h(\mathcal{Y} | x)\) by \(h(x)\), and the probability assigned by \(h(x)\) to \(y\) as \(h(y | x)\).

In interactive learning systems, we only observe feedback \(\delta(x, y)\) for the \(y\) presented to the user, but not for any of the other possible predictions \(\mathcal{Y}\setminus y\). In this work, feedback \(\delta: \mathcal{X}\times\mathcal{Y}\rightarrow\mathbb{R}\) is a cardinal loss. Small values for \(\delta(x, y)\) indicate user satisfaction with \(y\) for \(x\), while large values indicate dissatisfaction. The expected loss – called risk – of a hypothesis \(R(h)\) is defined as,

\[
R(h) = \mathbb{E}_{\mathcal{X},\mathcal{Y} \sim Pr(X)Pr(Y)} [\delta(x, y)].
\]

The goal of the system is to find a hypothesis \(h\in\mathcal{H}\) that has minimum risk (i.e., provides maximum user satisfaction).

To achieve this goal, we wish to use past interaction logs of the system for batch learning. Assume that the logging system acted according to a stationary policy \(h_0(x)\) with full support over \(\mathcal{Y}\). The data collected from this system is

\[
\mathcal{D} = \{(x_1, y_1, \delta_1, p_1), \ldots, (x_n, y_n, \delta_n, p_n)\},
\]
where \( y_i \sim h_0(x_i) \), \( \delta_i \equiv \delta(x_i, y_i) \), and \( p_i \equiv h_0(y_i \mid x_i) \) is the propensity of the logging system to predict \( y_i \) for input \( x_i \).

The Inverse Propensity Scoring approach \([5, 7]\) uses \( \mathcal{D} \) to Monte Carlo approximate \( \hat{R}(h) \) via importance sampling,

\[
\hat{R}(h) = 1/n \sum_{i=1}^{n} \delta_i h(y_i \mid x_i)/p_i. \tag{3}
\]

At first thought, one may think that directly estimating \( \hat{R}(h) \) over \( h \in \mathcal{H} \) and picking the empirical minimizer is a valid learning strategy. Unfortunately, there are several potential pitfalls.

First, this estimator has unbounded variance, since \( p_i \approx 0 \) in \( \mathcal{D} \) can cause \( \mathbb{E}_D[\hat{R}(h)] \) to be arbitrarily far away from the true risk \( R(h) \). This problem can be fixed by “clipping” the importance sampling weights \([4]\).

Second, importance sampling typically estimates \( \hat{R}(h) \) of different hypotheses \( h \in \mathcal{H} \) with vastly different variances. Consider two hypotheses \( h_1 \) and \( h_2 \), where \( h_1 \) is similar to \( h_0 \), but where \( h_2 \) samples predictions that were not well explored by \( h_0 \). Importance sampling gives us low-variance estimates for \( \hat{R}(h_1) \), but highly variable estimates for \( \hat{R}(h_2) \).

For a bounded \( \delta(\cdot, \cdot) \in [0, 1] \), define \( u_n \equiv \delta_i - \bar{\delta} \), \( \bar{\delta} = \frac{1}{n} \sum_{i=1}^{n} \{ u_i - \bar{\delta} \}^2 \),

\[
\bar{\delta} = \frac{1}{n(n-1)} \sum_{i,j=1}^{n} (u_i - u_j)^2,
\]

\[
\hat{R}(h) \equiv \frac{1}{n} \sum_{i=1}^{n} u_i^4/n, \tag{4}
\]

where \( M > 0 \) is a hyper-parameter chosen to trade-off bias and variance in the importance sampling estimate. Using an empirical Bernstein argument \([11]\), we prove that w.h.p.,

\[
\forall h \in \mathcal{H} : R(h) \leq \hat{R}(h) + \mathcal{O}\left( \sqrt{\bar{\delta} n/m} \right). \tag{5}
\]

Full details of the proof (which requires us to define a notion of capacity for stochastic hypothesis classes), is given in \([14]\).

This bound motivates the Counterfactual Risk Minimization principle: jointly optimize the objective \( \hat{R}(h) \) as well as its empirical standard deviation, where the latter serves as a data-dependent regularizer.

\[
\hat{h}^{CRM} = \arg\min_{h \in \mathcal{H}} \left\{ \hat{R}(h) + \lambda \sqrt{\bar{\delta} n/m} \right\}. \tag{6}
\]

\( M > 0 \) and \( \lambda \geq 0 \) are regularization hyper-parameters that can be selected via validation.

Using the CRM principle, we now derive a learning algorithm called POEM (Policy Optimizer for Exponential Models) for structured output prediction using stochastic hypotheses of the form

\[
h_w(y \mid x) = \exp\{w \cdot \phi(x, y)\}/\mathcal{Z}(x), \tag{7}
\]

where \( \mathcal{Z}(x) = \sum_{y \in \mathcal{Y}} \exp\{w \cdot \phi(x, y')\} \) is the partition function. These models are the stochastic (soft-max) extension of (hard-max) Structured SVMs, where \( w \) is a \( d \)-dimensional weight vector and \( \phi(x, y) \) is a \( d \)-dimensional joint feature map. For example, in multi-label document classification, for a news article \( x \) and a possible assignment of labels \( y \) represented as a bitvector, \( \phi(x, y) \) could simply be a concatenation of the bag-of-words features of the document \( x \), one copy for each of the assigned labels in \( y \), \( \mathcal{Z} \otimes y \).

The CRM principle gives rise to the following training objective for this hypothesis space.

\[
w^* = \arg\min_{w \in \mathbb{R}^d} \sum_{i=1}^{n} w_i^2 + \lambda \sqrt{\mathbb{E}[w_i^2]} + \mu \| w \|^2. \tag{8}
\]

While the objective in Equation \([5]\) is not convex in \( w \) (even for \( \lambda = 0 \), we find that gradient descent is sufficient to find a good local optimum that generalizes well. For efficient large-scale training, we derived a stochastic gradient algorithm that uses repeated linear variance majorization (see \([14]\)).

3. EXPERIMENTS

Consider multi-label classification with input \( x \in \mathbb{R}^d \) and prediction \( y \in \{0, 1\}^d \) using the feature map \( \phi(x, y) = x \otimes y \).

We conducted experiments on several multi-label datasets from the LibSVM repository summarized in Table 1.

<table>
<thead>
<tr>
<th>Name</th>
<th>p(# features)</th>
<th>q(# labels)</th>
<th>n-train</th>
<th>n-test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scene</td>
<td>294</td>
<td>6</td>
<td>1211</td>
<td>1196</td>
</tr>
<tr>
<td>Yeast</td>
<td>103</td>
<td>14</td>
<td>1500</td>
<td>917</td>
</tr>
<tr>
<td>TMC</td>
<td>30438</td>
<td>22</td>
<td>21519</td>
<td>7077</td>
</tr>
<tr>
<td>LYRL</td>
<td>47236</td>
<td>4</td>
<td>23149</td>
<td>781265</td>
</tr>
<tr>
<td>Media</td>
<td>120</td>
<td>101</td>
<td>30993</td>
<td>12914</td>
</tr>
</tbody>
</table>

For all datasets, we took aside 25% of \( n_{\text{train}} \) as validation set (for hyper-parameter selection) and treat the rest as the training set. We employ the Supervised \( \rightarrow \) Bandit conversion experiment methodology \([1]\). A supervised dataset \( \mathcal{D} = \{(x_i, y_i) \ldots (x_n, y_n)\} \) is used to simulate a bandit feedback dataset from a logging policy \( h_0 \) by sampling \( y_i \sim h_0(x_i) \) and collecting feedback \( \Delta(y_i^*, y_i) \). For \( h_0 \), in principle, we could use any arbitrary stochastic policy with full support on \( Y \). We choose a CRF (Conditional Random Field) trained on 10% of the training set using default hyper-parameters as \( h_0 \). \( \Delta(y^*, y) \) here is the Hamming loss between the supervised label \( y^* \) vs. the sampled label \( y \) for input \( x \). Note that using the doubly robust policy optimization scheme \([5]\) will involve an Offset Tree reduction \([2]\) that is exponential in \( q \) and is intractable for even moderate number of labels. After learning weight vectors (e.g. \( w_{\text{crf}} \)) on \( \mathcal{D} \), we report the expected loss per test instance

\[
\hat{R}(w) = \frac{1}{n} \sum_{y \in \mathcal{Y}} \mathbb{E}_{y \sim h_0(x)}[\Delta(y^*, y)].
\]

The expected Hamming loss of \( h_0 \) is the baseline to beat. Lower loss is better. The objective in Equation \([5]\) with \( \lambda = 0 \) is indicative of the state-of-the-art for counterfactual learning (optimized via L-BFGS and reported as IPS), and Equation \([3]\) optimized using SGD (see \([14]\)) is reported as POEM. We also report results for a supervised CRF trained on the entire training set as a baseline in Table 2 despite its unfair advantage of having access to the supervised labels.

<table>
<thead>
<tr>
<th>Name</th>
<th>Scene</th>
<th>Yeast</th>
<th>TMC</th>
<th>LYRL</th>
<th>Media</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPS</td>
<td>1.472</td>
<td>5.189</td>
<td>3.292</td>
<td>1.400</td>
<td>8.639</td>
</tr>
<tr>
<td>POEM</td>
<td>1.103</td>
<td>2.133</td>
<td>1.082</td>
<td>12.125</td>
<td></td>
</tr>
<tr>
<td>CRF</td>
<td>1.054</td>
<td>3.957</td>
<td>2.032</td>
<td>1.018</td>
<td>3.641</td>
</tr>
</tbody>
</table>

Across different datasets, the data-dependent variance regularizer of POEM consistently finds a hypothesis that generalizes better. For an extensive empirical analysis, see \([14]\).

This research was funded in part through NSF Awards IIS-1247637 and IIS-1217686, the JTCII Cornell-Technion Research Fund, and a gift from Bloomberg.
4. REFERENCES


