Neural Pseudo-Label Optimism for the Bank Loan Problem

Anonymous Author(s)
Affiliation
Address
email

Abstract

We study a class of problems best exemplified by the *bank loan* problem, where the learner only observes whether a customer will repay a loan if the loan is issued to begin with. Crucially, in these settings past decisions change the training data which in turn influences the model. As a consequence, in these settings it is possible for an algorithm to "get stuck" with a self-full-filling bad model, that never corrects false negatives since it never sees the true label for rejected data and hence accumulates infinite regret. While in linear settings this issue can be addressed by adding optimism directly into the model predictions, there are few methods that extend to the function approximation case using Deep Neural Networks. We present Pseudo-Labels for OpTimism (PLOT), a conceptually and computationally simple method applicable to DNNs. PLOT adds an optimistic label to a subset of points the current model rejects, trains the model on all data so far (including these points), and finally uses the resulting *optimistic* model for decision making. PLOT achieves competitive performance on a set of three challenging benchmark problems, requiring minimal hyperparameter tuning. Furthermore, our experiments clearly show that the mechanism more commonly overwrites false-negatives than true-negatives.

1 Introduction

There exist a wide range of scenarios where binary classification models are used in real time decisions, including bank lending [39, 21, 37], criminal recidivism prediction [38, 40, 4], credit card fraud [8, 36], spam detection [19, 34], self-driving motion planning [29, 25], and recommendation systems [30, 11, 17]. In many of these tasks, the feedback structure of an online learner can be modeled as one-sided, since the true labels are only observed for the examples that get "accepted". In this class of problems, which we refer to as the *bank loan problem* (BLP), the learner only observes if a customer will repay a loan if the loan is issued to begin with.

The BLP can be viewed as a variation on the classical active learning problem, adapted to the streaming setting [5, 10, 26]. In this scenario, unlabeled examples arrive sequentially and the learner is tasked with deciding whether to query for its label for a fixed cost in order to build a better model. In our setting the goal is similar, with the difference that the learner's objective is to maximize the number of positive points and minimize the number of negative points along the way.

There is a key, yet often overlooked aspect, of the BLP problem setting: The samples used to train the model, which prescribes which data points we should act upon next, are inherently biased by its *own past predictions*. In practical applications, there is a common belief that the main issue caused by such one-sided sampling is label imbalance [17], as the number of positive examples will be expected to be much higher than overall for the population. Indeed, this biasing of the labels leading to label imbalance can be a challenge, motivating much of the vast literature on label imbalance. However,

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

our challenges go beyond label imbalance. In the BLP setting a model can get stuck in a *self-fulfilling* false rejection loop, in which the very samples that *could* correct the erroneous model never enter the training data in the first place (since they are being rejected by the model). Existing methods with theoretical guarantees that can address this issue in the context of linear models by adding well calibrated optimism to the model, a popular approach being LinUCB and it's variants [9].

Unfortunately there are few methods that scale to the highly practically relevant *function approximation* setting, *e.g.*, to models based on Deep Neural Networks (DNNs). One challenge is that it is difficult to obtain a well-calibrated confidence intervals for these models, which underlie the methods that perform well in the linear settings. Recent work on UCB for Neural Contextual Bandits [42] is a promising step in this direction, providing UCB-style regret guarantees with DNNs. Unlike the general contextual bandit setting, we focus on binary classification, e.g. contextual bandit problems with only two arms. Focusing on this narrower binary action setting enables our method to explicitly add optimism into the DNN representation itself.

The bank loan problem can also be thought of as an instance of learning with partial feedback, a problem that was first studied by [18] under the name "apple tasting". In their work, the authors of [18] suggest to transform any learning procedure into an apple tasting one by randomly flipping some of the negative predictions into positive ones with probability decaying over time. They show upper and lower bounds on the number of mistakes made by the procedure in the partial feedback setting.

Our algorithm (PLOT) is conceptually similar to this approach. It assigns a positive *pseudo label* to some percentage of those query points that the model currently rejects. It then proceeds to update the model bey training on all data obtained so far, including these pseudo labels to obtain an *optimistic model*. Lastly, PLOT uses this updated model to decide whether a given point in the current batch should be accepted or rejected. PLOT attempts to turn potentially *self-fulfilling rejections* into *self-fulfill acceptances*: If the current data point was indeed positive and I already had it in my data set, would I then accept it? Crucially though, in contrast to false *rejections*, the false *accepts* are *self-correcting*. Once the model has (optimistically) accepted enough similar data points and obtained their true, negative label, these will overrule the impact of the optimistic label and result in a reject for novel queries.

While conceptually and computationally simple, we experimentally show that PLOT indeed obtains competitive performance across a set of 3 different benchmark problems in this domain, where with minimal hyperparameter tuning, it matches or outperforms greedy, ϵ-greedy (with a schedule) and state-of-the-art methods from literature. Furthermore our analysis shows that PLOT is 3-5 times more likely to *accept* a datapoint that the current model rejects if the data point is indeed a *true accept*, compared to a *true reject*.

2 Setting

We formally define the bank loan problem as repeated loss minimization in sequential binary classification. In this setting a learner and a data generator interact in a sequential manner through a series of rounds, which we index by time t. At the beginning of every round t, the learner receives a datapoint $\mathbf{x}_t \in \mathbb{R}^d$ (or a batch of datapoints $\mathbf{x}_{t,1}, \cdots, \mathbf{x}_{t,B}$) and has to decide if to observe its label (or labels $y_{t,1}, \cdots, y_{t,B}$) $y_t \in \{0, 1\}$ or not. To simplify the discussion we will focus our description on the single datapoint setting. All definitions below can naturally be extended to the batch scenario.

Utility We assume the learner's objective is to maximize its cumulative accuracy. That is, the number of correct decisions it has made during training. This is a standard assumption in the literature (see for example [20]). In our model, if the learner decides to observe the datapoint label, it incurs in a (random) reward of $2y_t - 1$ otherwise it incurs in a reward of 0. For example, in the loan scenario this reward model corresponds to a lender that makes a unit gain for a repaid loan, incurs in a unit loss for an unpaid loan and collects zero reward whenever no loan is given to the lender. Throughout this work we will make the following discriminative assumption regarding the distribution of the labels,

Assumption 1 (Labels generation)**.** *We assume the labels $y_t \in \{0, 1\}$ are generated according to the following model:*

$$y_t = \begin{cases} 1 & \textit{with probability } \mu(f_{\theta_\star}(\mathbf{x}_t)) \\ 0 & \textit{o.w.} \end{cases} \tag{1}$$

For some function $f_{\boldsymbol{\theta}_\star} : \mathbb{R}^d \to \mathbb{R}$ parametrized by $\boldsymbol{\theta}_\star \in \Theta$ and where $\mu(z) = \frac{\exp(z)}{1+\exp(z)}$ is the logistic link function. We denote the function class parametrized by $\boldsymbol{\theta} \in \Theta$ as $\mathcal{F}_\Theta = \{f_{\boldsymbol{\theta}} \text{ s.t. } \boldsymbol{\theta} \in \Theta\}$.

In the algorithms we propose the learner's decision at time t is parametrized by a parameter $\boldsymbol{\theta}_t$ and $f_{\boldsymbol{\theta}_t}$ and takes the form,

$$\text{If } f_{\boldsymbol{\theta}_t}(\boldsymbol{x}_t) \geq 0 \text{ accept}$$

We call the function $f_{\boldsymbol{\theta}_t}$ the learner's model. We denote by $a_t \in \{0, 1\}$ to the indicator of whether the learner has decided to accept or not datapoint $\mathbf{x}_t$. We measure the performance of any adaptive procedure that selects between seeing or not the label by its pseudo-regret:

$$\mathcal{R}(t) = \sum_{\ell=1}^{t} \max(0, 2\mu(f_{\boldsymbol{\theta}_\star}(\mathbf{x}_t)) - 1) - a_t(2\mu(f_{\boldsymbol{\theta}_\star}(\mathbf{x}_t)) - 1)$$

For all t we denote the pseudo-reward received at time t by $r_t = a_t(2\mu(f_{\boldsymbol{\theta}_\star}(\mathbf{x}_t)) - 1)$. Minimizing regret is a standard objective in the online learning and bandits literature (see [23]). Notice that as a consequence of Assumption 1, the *optimal* utility maximizing decision rule equals the true model $f_{\boldsymbol{\theta}_\star}$. The problem we have just described can be easily modeled as a setting where at the beginning of every round t learner receives a context (either a single $\mathbf{x}_t \in \mathbb{R}^d$ or a batch of datapoints $\mathbf{x}_{t,1}, \cdots, \mathbf{x}_{t,B}$ and has the choice of selecting one of two actions $\{\text{Accept}, \text{Reject}\}$ (in the batch setting the learner has the ability to select among 2^B actions, corresponding to $\{\text{Accept}, \text{Reject}\}^B$, one $\{\text{Accept}, \text{Reject}\}$ choice per datapoint. It is easy to see this characterization implies an immediate reduction of our problem into the setting of contextual bandits where the payout of one of the actions is known by the learner (Reject).

In order to distinguish the problem we study from a more general contextual bandits setting, we refer to the online classification problem we study as the bank-loan problem.

3 Related Work

Repeated Loss Minimization A closely related problem setting (see Section 2) is Repeated Loss Minimization. Previous works [16] have studied the problem of repeated classification in the setting where the consequence of an acceptance or rejection decision produces a change in the underlying distribution of individuals faced by the decision maker. In this work, the distributional shift induced by the learner's decisions is assumed to be intrinsic to the dynamics of the world. This line of work has recently garnered a flurry of attention and inspired the formalization of different problem domains such as strategic classification [14] and performative prediction [31, 28]. A common theme in these works is the necessity of thinking strategically about the learner's actions and how these may affect its future decisions as a result of the world reacting to them.

In this paper we focus on a different set of problems encountered by decision makers when faced with the problem of repeated loss minimization in the presence of a utility function. We do not think of the world as a strategic decision maker, instead we treat the distribution of covariates faced by the learner as fixed, and focus on understanding the effects that making online decisions can have in the future accuracy and utility experienced by an agent engaged in repeated classification. The main goal in this setting is to devise methods that allow the learner to minimize the future effect of its own uncertainty. Thus, the learner's task is not contingent on the arbitrariness of the world, but on a precise understanding of its own knowledge of the world. This is precisely the type of task that is studied in the bank loan problem setting.

Learning with partial feedback [34] studies the one-sided feedback setting for the application of email spam filtering and show that the approach of [18] was less effective than a simple greedy strategy. [6] proposes an active learning method for linear models to query an example's label randomly with probability based on the model's prediction score for that example. [3] considers the problem of one-sided learning in the group-based fairness context with the goal of satisfying equal opportunity [15] at every round. They consider convex combinations over a finite set of classifiers and arrive at a solution which is a randomized mixture of at most two of these classifiers. [7] studies a setting which generalizes the one-sided feedback, called *partial monitoring*, through considering repeated two-player games in which the player receives a feedback generated by the combined choice

of the player and the environment, proposing a randomized solution. [1] provides a classification of such two-player games in terms of the regret rates attained and [2] study a variant of the problem with side information.

Contextual Bandits As we have described in Section 2, the bank loan problem can be formulated in the language of contextual bandits. For example, in [27] the authors study the problem of repeated loss minimization in the context of the loan and bail scenarios, where a learner (the bank) wants to give out loans to customers who arrive to the institution one by one. Here, the authors study the bank loan problem in the presence of offline logged data, a markedly different scenario than the one we treat in this work. A fundamental component of our work is the easy and empirically robust extension of our algorithmic ideas into the DNN scenario. Recent works have forayed into the analysis and deployment of bandit algorithms in the setting of function approximation, most notably the authors of [32] conduct an extensive empirical evaluation of existing bandit algorithms on public datasets. A more recent line of work has focused on devising ways to add optimism to the predictions of neural network models. Methods such as Neural UCB and Shallow Neural UCB [42, 41] are designed to add an optimistic bonus to the model predictions, that is a function of the representation layers of the model. Their theoretical analysis is inspired by Neural Tangent Kernel (NTK) insights. Other recent works in the contextual bandit literature such as [13] have started to pose the question of how to extend theoretically valid guarantees into the function approximation scenario, so far with limited success [35].

4 Background

We start by showing that when the function class $\mathcal{F}_\Theta$ corresponds to a family of linear functions, the bank loan problem can be fully described as a generalized contextual linear bandit problem.

4.1 Optimism in Realizable Logistic Linear Models

For the sake of illustration in this section we assume the labels $y_t \in \{0, 1\}$ are generated according to a generalized linear model where $f_{\boldsymbol{\theta}}(\mathbf{x}) = \mathbf{x}^\top \boldsymbol{\theta}$ for all $\boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^d$ and $\boldsymbol{x} \in \mathbb{R}^d$.

Since the true reward parameter $\boldsymbol{\theta}_\star$ is unknown we will estimate it using samples. At any round $t \in [T]$, a natural way of computing an estimator of $\boldsymbol{\theta}_\star$ given past accepted datapoints $\{\boldsymbol{x}_\ell\}_{\ell \in [t-1]}$ and labels $\{y_\ell\}_{\ell \in [t-1]}$ is by minimizing the ℓ_2-regularized cross-entropy loss

$$\mathcal{L}_t(\boldsymbol{\theta}) := -\sum_{\ell=1}^{t-1} y_\ell \log\left(\mu(\boldsymbol{\theta}^\top \boldsymbol{x}_\ell) - (1 - y_\ell) \log\left(1 - \mu(\boldsymbol{\theta}^\top \boldsymbol{x}_\ell)\right)\right) + \frac{\lambda \|\boldsymbol{\theta}\|_2^2}{2}$$

In this setting the loss function is strictly convex, and the empirical minimizer is defined to be

$$\widehat{\boldsymbol{\theta}}_t := \operatorname*{arg\,min}_{\boldsymbol{\theta} \in \mathbb{R}^d} \mathcal{L}_t(\boldsymbol{\theta}). \tag{2}$$

We proceed to use a adapt a result by [33] Proposition 7 who studied the online logistic bandits problem to establish that at every time-step and every datapoint $\boldsymbol{x}$ the difference between the predicted value $\mu(\boldsymbol{\theta}_t^\top \boldsymbol{x})$ and the true value of $\mu(\boldsymbol{\theta}_\star^\top \boldsymbol{x})$ is small.

Lemma 1. *There exists a confidence radius function $\beta_t : [0, 1] \to \mathbb{R}$ such that for any $\delta \in (0, 1]$ the event*

$$\mathcal{E}_\delta := \left\{ \textit{for all } t \in [N], \boldsymbol{x} \in \mathbb{R}^d : \left| \mu(\boldsymbol{\theta}_\star^\top \boldsymbol{x}) - \mu(\widehat{\boldsymbol{\theta}}_t^\top \boldsymbol{x}) \right| \leq \sqrt{\kappa} \beta_t(\delta) \|\boldsymbol{x}\|_{\boldsymbol{\Sigma}_t^{-1}} \right\}, \tag{3}$$

then $\mathbb{P}(\mathcal{E}_\delta) \geq 1 - \delta$. Where $\boldsymbol{\Sigma}_t := \kappa \mathbf{I} + \sum_{\ell=1}^{t-1} \boldsymbol{x}_\ell \boldsymbol{x}_\ell^\top$, for all $t \geq 1$ and $\|\mathbf{v}\|_{\mathbf{A}} = \sqrt{\mathbf{v}^\top \mathbf{A} \mathbf{v}}$ and $\kappa := \max_{\boldsymbol{x}: \|\boldsymbol{x}\| \leq S} \sup_{\boldsymbol{\theta}: \|\boldsymbol{\theta}\| \leq B} \frac{1}{\mu'(\boldsymbol{\theta}^\top \boldsymbol{x})}$.

A typical approach to deal with uncertainty in the learner's predictions and ensure she does not ignore points that may yield a positive reward, nor tries out points that incur in a large loss, is by adding some amount of optimism into the learner's predictions. The amount of optimism needs to be

sufficient to ensure the learner's predictions overestimate the true outcomes, but it also requires to degrade appropriately to ensure these optimistic predictions converge to the true values as more data is gathered by the learner. Any method that ensures optimism holds for all its model predictions will be accepting every single point that should be accepted (in our setting any $\mathbf{x}$ such that $f_{\boldsymbol{\theta}_\star}(\mathbf{x}) \geq 0$) If the optimism overestimation degrades at the appropriate rate, such method will also be able to reject poorly performing points and achieve a low false positive rate. For a more in depth discussion on the role of optimism in deriving regret guarantees for bandit algorithms see Section A in the Appendix.

Lemma 6 can be used to construct an optimistic estimator for $\mu(\boldsymbol{\theta}_\star^\top \mathbf{x})$ for all $\mathbf{x} \in \mathbb{R}^d$ of the form $\mu(\hat{\boldsymbol{\theta}}_t^\top \mathbf{x}) + \sqrt{\kappa}\beta_t(\delta)\|\boldsymbol{x}\|_{\boldsymbol{\Sigma}_t^{-1}}$. Algorithm 1 makes a careful use of this optimistic estimator to decide which datapoints to accept or reject.

Algorithm 1 GLM accept / reject

Input: Regularizer $\lambda \geq 0$, confidence radius function $\beta_t : [0,1] \to \mathbb{R}$.
for $t = 1, \cdots$ **do**
1. Calculate $\widehat{\boldsymbol{\theta}}_t$.
2. Choose $a_t = \begin{cases} 1 & \text{if } \mu(\boldsymbol{x}_t^\top \widehat{\boldsymbol{\theta}}_t) + \sqrt{\kappa}\beta_t(\delta)\|\boldsymbol{x}\|_{\boldsymbol{\Sigma}_t^{-1}} \geq \frac{1}{2} \\ 0 & \text{o.w.} \end{cases}$
3. If $a_t = 1$, observe y_t.

end

4.2 Regret analysis for Realizable Logistic Linear Models

We start by showing that the repeated binary classification problem in the setting of realizable logistic linear models can be reduced to a generalized contextual linear bandit problem whose contexts are sets of size 2 containing the query point $\mathbf{x}_t$ and the zero vector $\mathbf{0}$. The results of this Section work under the following assumption.

Boundedness We assume that $\|\boldsymbol{x}_t\| \leq S$ for all $\boldsymbol{x}_t$ for some known value $S > 0$ and $\|\boldsymbol{\theta}_\star\| \leq B$ for some known constant $B > 0$.

We proceed to show that whenever the linear realizability assumption holds $f_{\boldsymbol{\theta}_\star}(\mathbf{x}) = \mathbf{x}^\top \boldsymbol{\theta}_\star$ the bank-loan problem can

Lemma 2. *The repeated classification problem in the realizable linear setting can be cast as a contextual logistic bandit problem where the context sets at time t are of the form* $\{\mathbf{x}_t, \mathbf{0}\}$.

Proof. Notice that $\mu(\mathbf{0}^\top \boldsymbol{\theta}_\star) = 1/2$ irrespective of the value of $\boldsymbol{\theta}_\star$. Furthermore, the pseudo-reward (and therefore the pseudo-regret) accrued by rejecting a datapoint is equivalent to playing the zero vector $\mathbf{0}$ as an action. This completes the reduction between the repeated classification under a realizable logistic model to the contextual bandit setting. □

Denote by $\mathbf{z}_t$ to the vector in $\{\mathbf{0}, \mathbf{x}_t\}$ played by the algorithm. The pseudo-reward received at time t satisfies $r_t = 2\mu(\mathbf{z}_t^\top \widehat{\boldsymbol{\theta}}_t) - 1$. Algorithm 1 satisfies the following regret guarantee,

Theorem 1. *[Regret guarantee] Let $\delta \in (0,1)$. With probability at least $1 - 2\delta$, the regret of Algorithm 1 satisfies,*

$$\mathcal{R}(t) \leq \widetilde{\mathcal{O}}(d\sqrt{\kappa}t)$$

where $\kappa := \max_{\boldsymbol{x}:\|\boldsymbol{x}\|\leq S} \sup_{\boldsymbol{\theta}:\|\boldsymbol{\theta}\|\leq B} \frac{1}{\mu'(\boldsymbol{\theta}^\top \boldsymbol{x})}$ and $\widetilde{\mathcal{O}}(\cdot)$ hides factors logarithmic in δ and constants dependencies such as B and S.

The proof of Theorem 1 can be found in Appendix B.

5 Pseudo-Labels for Optimism

In this setting we make the following version of realizability Assumption 1. We note that although we make this assumption for the sake of our theoretical results, in practice our algorithms do not require this to hold.

Assumption 2 (Neural Realizability)**.** *There exists a DNN $f_{\boldsymbol{\theta}_\star} : \mathbb{R}^d \to \mathbb{R}$ parametrized by $\boldsymbol{\theta}_\star \in \mathbb{R}^k$ and of known architecture such that for all $\mathbf{x}_t \in \mathbb{R}^d$:*

$$y_t = \begin{cases} 1 & \textit{with probability } \mu(f_{\boldsymbol{\theta}_\star}(\mathbf{x}_t)) \\ 0 & \textit{o.w.} \end{cases} \tag{4}$$

Although we make Assumption 2 for the purpose of analyzing and explaining our algorithms, in practice it is not necessary that this assumption holds. An additional difficulty that arises in the neural realizable scenario that is not present in the linear setting is that there may not exist a vector $\mathbf{z} \in \mathbb{R}^d$ with which to model the payoff of the Reject action. If this is the case there may not exist an easy way to represent the bank loan problem as a natural instance of a two action contextual bandit problem with the payoffs fully specified by the neural function class at hand. We can get around this issue because in the bank loan problem it is enough to compare the model's prediction with the neutral probability $1/2$.

Recall that in our setting, the learner interacts with the environment in a sequential manner during time-steps $t = 1, \ldots$ observing points $\{\mathbf{x}_t\}_{t=1}^{\infty}$ and labels y_t only during those timesteps when $a_t = 1$. The learner also experiences a reward of $a_t(2\mu(f_{\boldsymbol{\theta}_\star}(\mathbf{x}_t)) - 1)$, a quantity that is never fully revealed to the learner. Whenever Assumption 2 holds, a natural way of computing an estimator $\widehat{\boldsymbol{\theta}}_t$ of $\boldsymbol{\theta}_\star$ is via maximum-likelihood estimation. If we denote by $\mathcal{D}_t = \{(\mathbf{x}_\ell, y_\ell) \text{ s.t.} a_\ell = 1 \text{ and } \ell \leq t-1\}$ as the dataset of accepted points up to the learners decision at time $t-1$, the regularized log-likelihood (or negative cross-entropy loss) can be written as,

$$\mathcal{L}^\lambda(\boldsymbol{\theta}|\mathcal{D}_t) = \sum_{(\mathbf{x},y)\in\mathcal{D}_t} y \log\left(f_{\boldsymbol{\theta}}(\boldsymbol{x})\right) + (1-y)\log\left(1 - \mu(f_{\boldsymbol{\theta}}(\mathbf{x})\right) - \frac{\lambda_{\boldsymbol{\theta}}}{2}\|\boldsymbol{\theta}\|_2^2$$

Just as in the case of generalized linear contextual bandits discussed in Section 4.1, it likely that utilizing the model given by $f_{\hat{\boldsymbol{\theta}}_t}$ may lead to catastrophic under or over estimation of the true response $\mu(f_{\boldsymbol{\theta}_\star}(\mathbf{x}_t))$ for any query point $\mathbf{x}_t$. In order to alleviate this we make use of a form of implicit optimism resulting from optimizing for what we term as the pseudo label counterfactual loss.

Definition 1. *Let $\mathcal{D} = \{(\mathbf{x}_\ell, y_\ell)\}_{\ell=1}^N$ be a datset consisting of* labeled *covariates $\mathbf{x}_\ell \in \mathbb{R}^d$ and responses $y_\ell \in \{0,1\}$ and let $\mathcal{B} = \{\mathbf{x}^{(j)}\}_{j=1}^B \subset \mathbb{R}^d$ be a dataset of* unlabeled *covariates. We define the pseudo label counterfactual loss of $(\mathcal{D}, \mathcal{B})$ as,*

$$\begin{aligned}\mathcal{L}^{\mathcal{C}}(\boldsymbol{\theta}|\mathcal{D},\mathcal{B}) &= \mathcal{L}^\lambda(\boldsymbol{\theta}|\mathcal{D} \cup \{(\mathbf{x},1) \textit{ for } \mathbf{x} \in \mathcal{B}\}) \\ &= \underbrace{\sum_{(\mathbf{x},y)\in\mathcal{D}} y\log\left(f_{\boldsymbol{\theta}}(\mathbf{x})\right) + (1-y_s)\log(1-\mu(f_{\boldsymbol{\theta}}(\mathbf{x})))}_{\textit{cross-entropy loss}} + \underbrace{\sum_{\boldsymbol{x}\in\mathcal{B}} \log(f_{\boldsymbol{\theta}}(\mathbf{x}))}_{\textit{counter-factual loss}} - \frac{\lambda\|\boldsymbol{\theta}\|^2}{2}\end{aligned}$$

Let's take a closer look at the pseudo-label counterfactual loss. Given any pair of labeled-unlabeled datasets $(\mathcal{D}, \mathcal{B})$, the counterfactual loss $\mathcal{L}^{\mathcal{C}}(\theta|\mathcal{D},\mathcal{B})$ aims to minimize the cross-entropy of a dataset of pairs $(\mathbf{x}, y)$ of the form $(\mathbf{x}, y) \in \mathcal{B}$ or $(\mathbf{x}, 1)$ such that $\mathbf{x} \in \mathcal{B}$. In other words, the minimizer of $\mathcal{L}^{\mathcal{C}}(\boldsymbol{\theta}|\mathcal{D},\mathcal{B})$ is trying to satisfy two objectives at once: achieve a good fit of the labeled data, while at the same time, encouraging the resulting classifier to assign a positive label to the unlabeled points in $\mathcal{B}$. The model $\widehat{\boldsymbol{\theta}}^{\mathcal{C}}$ resulting from minimizing $\mathcal{L}^{\mathcal{C}}$ will therefore strive to be optimistic over $\mathcal{B}$ while keeping a low loss value, and consequently a high accuracy (in case Assumption 2 holds) over the true labels of the points in $\mathcal{D}$. We note that if $\mathcal{D}$ is much larger than $\mathcal{B}$, optimizing $\mathcal{L}^{\mathcal{C}}$ will favor models that are accurate over $\mathcal{D}$ instead of optimistic over $\mathcal{B}$. Whenever $|\mathcal{D}| \ll |\mathcal{B}|$ the opposite is true.

Based on these insights, we design Pseudo Labels for Optimism (PLOT), an algorithm that utilizes the pseudo label counterfactual loss to inject the appropriate amount of optimism into the learner's decisions: high for points that have not been seen much during training, and low for those points whose acceptance may cause a catastrophic loss increase over the points accepted by the learner so far.

Algorithm 2 Pseudo Labels for Optimism (PLOT)

Input: $\epsilon-$greedy exploration parameter,
Initialize accepted dataset $\mathcal{D}_1 = \emptyset$
for $t = 1, \cdots$ **do**

1. Observe batch $\mathcal{B}_t = \{\mathbf{x}_t^{(j)}\}_{j=1}^B$ and sample $Q_t = \{q_j\}_{j=1}^B$ such that $q_j \overset{i.i.d.}{\sim} \text{Ber}(\epsilon)$
2. Build the MLE estimator,
$$\widehat{\boldsymbol{\theta}}_t = \min_{\boldsymbol{\theta}} \mathcal{L}_t(\boldsymbol{\theta}|\mathcal{D}_t)$$
3. Compute the pseudo-label filtered batch $\widetilde{\mathcal{B}}_t = \{(\mathbf{x}_t^{(j)}, 1) \text{ s.t. } f_{\widehat{\boldsymbol{\theta}}_t}(\mathbf{x}_t^{(j)}) < 0 \text{ and } q_j = 1\}$.
4. Calculate the minimizer of the empirical pseudo-label counterfactual loss,
$$\widehat{\boldsymbol{\theta}}_t^{\mathcal{C}} = \min_{\boldsymbol{\theta}} \mathcal{L}_t^{\mathcal{C}}(\boldsymbol{\theta}|\mathcal{D}_t, \widetilde{\mathcal{B}}_t),$$
3. For all $\mathbf{x}^{(j)} \in \mathcal{B}_t$ compute acceptance decision via $a_t^{(j)} = \begin{cases} 1 & \text{if } f_{\widehat{\boldsymbol{\theta}}_t^{\mathcal{C}}}(\mathbf{x}_t^{(j)}) \geq 0 \\ 0 & \text{o.w.} \end{cases}$
4. Update $\mathcal{D}_{t+1} \leftarrow \mathcal{D}_t \cup \{(\mathbf{x}_t^{(j)}, y_t^{(j)})\}_{j \in \{1, \cdots, B\} \text{ s.t. } a_t^{(j)} = 1}$.

end

During the very first round ($t = 1$) of the algorithm PLOT accepts all the points in $\mathcal{B}_t$. In subsequent rounds PLOT makes use of a dual $\epsilon-$greedy and MLE-greedy filtering subroutine to find a subset of the current batch composed of those points that are *both* currently being predicted as rejects by the MLE estimator and have been selected by the $\epsilon-$greedy schedule (see step 3 of PLOT).

This dual filtering mechanism ensures that only a small proportion (based on the ϵ) of the datapoints are ever considered to be included into the empirical pesudo label counterfactual loss. The MLE filtering mechanism further ensures the not all the points selected by $\epsilon-$greedy are further investigated, but only those that are currently being rejected by the MLE model. This has the effect of preventing the pesudo-label filtered batch from growing too large.

As we have mentioned above, the relative sizes of the labeled and unlabeled batches has an effect on the degree of optimism the algorithm will inject into its predictions. As the size of the collected dataset grows, the inclusion of $\widetilde{\mathcal{B}}_t$, has less and less effect on $\widehat{\boldsymbol{\theta}}_t^{\mathcal{C}}$. In the limit, once the dataset is sufficiently large, the inclusion of $\widetilde{\mathcal{B}}_t$ into the pseudo-label loss has vanishing effect.

The following guarantee shows that in the case of separable data satisfying the neural generative assumption 2, the MLE and counterfactual models $\widehat{\boldsymbol{\theta}}_t, \widehat{\boldsymbol{\theta}}_t^{\mathcal{C}}$ computed by Algorithm 3 converge to a classifier with perfect accuracy.

Theorem 2. *Let $\mathcal{P}$ be a distribution over covariate, label pairs $(\mathbf{x}, y)$ satisfying*

1. *All $\mathbf{x} \in \text{supp}(\mathcal{P})$ are bounded $\|\text{x}\| \leq B$.*
2. *The conditional distributions of the labels y satisfy the data generating Assumption 2 with $\mathcal{F}$ a class of smooth DNN functions.*
3. *For all $\mathbf{x}$ in the distribution $|f_{\boldsymbol{\theta}_\star}(\mathbf{x})| \geq \tau > 0$*

Then the greedy classifiers induced by the MLE and counterfactual models $\widehat{\boldsymbol{\theta}}_t, \widehat{\boldsymbol{\theta}}_t^{\mathcal{C}}$ computed by Algorithm 3 Algorithm 2 converge to a classifier with perfect accuracy.

The proof can be found in Appendix E. As a consequence of Theorem 2, we conclude that PLOT is able to converge in the limit to a perfectly accurate classifier whenever a margin condition holds. In the following section we conduct extensive empirical studies of PLOT and demonstrate competitive finite time regret on a variety of public classification datasets.

6 Experimental Results

We evaluate the performance of PLOT[1] on three binary classification problems in the BLP setting. In round t, the algorithm observes context $x_t \in R^d$, and classifies the point, a.k.a accepts/rejects the point. If the point is accepted, a reward of one is given if that point is from the target class, and zero otherwise. If the point is rejected, the reward is censored – always zero. We focus on two datasets from the UCI Collection [12], the Adult dataset and the Bank dataset. Additionally we make use of MNIST [24] (d=784). The Adult dataset is defined as a binary classification problem, where the positive class has income > $50k. The Bank dataset is also binary, with the positive class as a successful marketing conversion. On MNIST, we convert the multi-class problem to binary classification by taking the positive class to be images of the digit 5, and treating all other images as negatives. Our main metric of interest is regret, measured against a baseline model trained on the entire dataset. The baseline model is used instead of the true label, as many methods cannot achieve oracle accuracy on real-world problems even with access to the whole dataset.

We focus on comparing our method to other neural algorithms, as prior papers [32], [22], [42] generally find neural models to have the best performance on these datasets. In particular, we focus on NeuralUCB[42] as a strong benchmark method. We perform a grid search over a few values of the hyperparameter of NeuralUCB, considering $\{0.1, 1, 4, 10\}$. We report results from the best value. We also consider neural greedy and ϵ-greedy methods. For ϵ-greedy, we follow [22], and give the method an unfair advantage. We use a decayed schedule, dropping to 0.1% exploration by T=2000. Otherwise, the performance is too poor to plot. For computational efficiency, we run our method on batches of data, with batch size n = 32. We average results over 5 runs, running for a horizon of t = 2000 rounds. Our dataset consists of the points accepted by the algorithm, for which we have the true labels. At each timestep, we train a neural network on this data, for a fixed number of steps. Then, a second neural network is cloned from those weights. A new dataset with pseudolabel data and historical data is constructed, and the second neural network is trained on that dataset for the same number of epochs. This allows us to keep a continuously trained model which only sees true labels. The pseudolabel model only ever sees one batch of pseudolabels. Each experiment runs on a single Nvidia Pascal GPU, and replicated experiments, distinct datasets, and methods can be run in parallel, depending on GPU availability.

Our method's cumulative regret is consistently competitive with other methods, and outperforms on MNIST. In addition, the variance of our method is much lower than that of NeuralUCB and Greedy, showing very consistent performance across the five experiments.

Along with regret plots, we provide a breakdown of the decisions made by our model. As described in Section 5, the pseudo-label model only acts on ϵ-percent of points classified as negative by the base model. We provide the cumulative probability of acceptance of true positive and true negative points acted on by the pseudo-label model. As the base model improves, the pseudolabel model receives less false positives, and can become more confident in supporting rejections from the base model. To differentiate this decaying process from the pseudo-label learning, we highlight the significant gap between the probability of accepting positives and the probability of accepting negatives in our method. This shows that the PLOT method is not simply performing a decayed-ϵ strategy, but rather learning when to inject optimism into the base classifier.

7 Conclusion

We propose PLOT, a novel algorithm that provides end-to-end optimism for the bank loan problem with neural networks. Rather than post-hoc optimism separate from the neural net training, optimism is directly incorporated into the neural net loss function through the addition of optimistic pseudo-labels. We provide regret guarantees for PLOT, and demonstrate its performance on real-world problems. The method's strong performance on real-world problems illustrates the value of end-to-end optimism.

Our current analysis and deployment of PLOT is focused on the bank loan problem, where pseudo-label semantics are most clear. Due to the deep connections with active learning and binary contextual bandits, extending this work to larger action spaces is an interesting future direction.

[1] Google Colab: shorturl.at/bhiH3

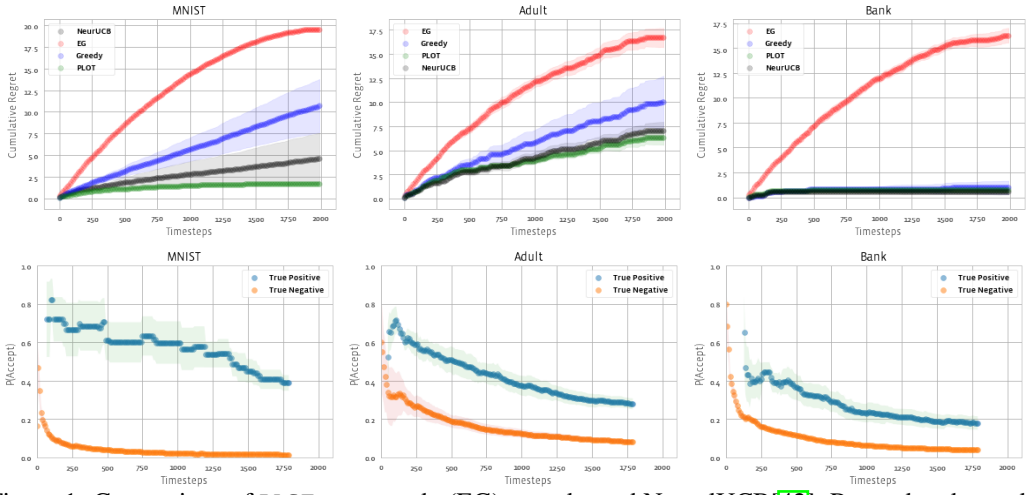

Figure 1: Comparison of PLOT to ϵ-greedy (EG), greedy, and NeuralUCB[42]. Reward and pseudo-label accuracy are reported as a function of the timestep. One standard deviation from the mean, computed across the five experiments, is shaded.

8 Statement of Broader Impact

Explicitly incorporating optimism into neural representation learning is key to ensuring optimal exploration in the *bank loan* problem. Other methods for exploration run the risk of feature blindness, where a neural network loses its uncertainty over certain features. When a representation learning method falls victim to this, additional optimism is insufficient to ensure exploration. This has ramifications for fairness and safe decision making. We believe that explicit optimism is a key step forward for safe and fair decision making.

However, we do want to provide caution around our method's limitations. Our regret guarantees and empirical results assume I.I.D data, and may not prevent representation collapse in non-stationary and adversarial settings. Additionally, although our empirical results show strong performance in the non-separable setting, our regret guarantees only hold uniformly in the separable setting.

References

[1] A. Antos, G. Bartók, D. Pál, and C. Szepesvári. Toward a classification of finite partial-monitoring games. *Theoretical Computer Science*, 473:77–99, 2013.

[2] G. Bartók and C. Szepesvári. Partial monitoring with side information. In *International Conference on Algorithmic Learning Theory*, pages 305–319. Springer, 2012.

[3] Y. Bechavod, K. Ligett, A. Roth, B. Waggoner, and S. Z. Wu. Equal opportunity in online classification with partial feedback. In *Advances in Neural Information Processing Systems*, pages 8972–8982, 2019.

[4] R. Berk. An impact assessment of machine learning risk forecasts on parole board decisions and recidivism. *Journal of Experimental Criminology*, 13(2):193–216, 2017.

[5] A. Bordes, S. Ertekin, J. Weston, and L. Bottou. Fast kernel classifiers with online and active learning. *Journal of Machine Learning Research*, 6(Sep):1579–1619, 2005.

[6] N. Cesa-Bianchi, C. Gentile, and L. Zaniboni. Worst-case analysis of selective sampling for linear classification. *Journal of Machine Learning Research*, 7(Jul):1205–1230, 2006.

[7] N. Cesa-Bianchi, G. Lugosi, and G. Stoltz. Regret minimization under partial monitoring. *Mathematics of Operations Research*, 31(3):562–580, 2006.

[8] P. K. Chan, W. Fan, A. L. Prodromidis, and S. J. Stolfo. Distributed data mining in credit card fraud detection. *IEEE Intelligent Systems and Their Applications*, 14(6):67–74, 1999.

[9] W. Chu, L. Li, L. Reyzin, and R. Schapire. Contextual bandits with linear payoff functions. In G. Gordon, D. Dunson, and M. Dudík, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 208–214, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR.

[10] W. Chu, M. Zinkevich, L. Li, A. Thomas, and B. Tseng. Unbiased online active learning in data streams. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 195–203, 2011.

[11] P. Covington, J. Adams, and E. Sargin. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*, pages 191–198, 2016.

[12] D. Dua and C. Graff. UCI machine learning repository, 2017.

[13] D. Foster and A. Rakhlin. Beyond ucb: Optimal and efficient contextual bandits with regression oracles. In *International Conference on Machine Learning*, pages 3199–3210. PMLR, 2020.

[14] M. Hardt, N. Megiddo, C. Papadimitriou, and M. Wootters. Strategic classification. In *Proceedings of the 2016 ACM conference on innovations in theoretical computer science*, pages 111–122, 2016.

[15] M. Hardt, E. Price, and N. Srebro. Equality of opportunity in supervised learning. In *Advances in neural information processing systems*, pages 3315–3323, 2016.

[16] T. Hashimoto, M. Srivastava, H. Namkoong, and P. Liang. Fairness without demographics in repeated loss minimization. In *International Conference on Machine Learning*, pages 1929–1938. PMLR, 2018.

[17] X. He, J. Pan, O. Jin, T. Xu, B. Liu, T. Xu, Y. Shi, A. Atallah, R. Herbrich, S. Bowers, et al. Practical lessons from predicting clicks on ads at facebook. In *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising*, pages 1–9, 2014.

[18] D. P. Helmbold, N. Littlestone, and P. M. Long. Apple tasting. *Information and Computation*, 161(2):85–139, 2000.

[19] N. Jindal and B. Liu. Review spam detection. In *Proceedings of the 16th international conference on World Wide Web*, pages 1189–1190, 2007.

[20] N. Kilbertus, M. G. Rodriguez, B. Schölkopf, K. Muandet, and I. Valera. Fair decisions despite imperfect predictions. In *International Conference on Artificial Intelligence and Statistics*, pages 277–287. PMLR, 2020.

[21] G. Kou, Y. Peng, and C. Lu. Mcdm approach to evaluating bank loan default models. *Technological and Economic Development of Economy*, 20(2):292–311, 2014.

[22] B. Kveton, C. Szepesvari, S. Vaswani, Z. Wen, T. Lattimore, and M. Ghavamzadeh. Garbage in, reward out: Bootstrapping exploration in multi-armed bandits. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3601–3610. PMLR, 09–15 Jun 2019.

[23] T. Lattimore and C. Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

[24] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[25] U. Lee, S. Yoon, H. Shim, P. Vasseur, and C. Demonceaux. Local path planning in a complex environment for self-driving car. In *The 4th Annual IEEE International Conference on Cyber Technology in Automation, Control and Intelligent*, pages 445–450. IEEE, 2014.

[26] J. Lu, P. Zhao, and S. C. Hoi. Online passive-aggressive active learning. *Machine Learning*, 103(2):141–183, 2016.

[27] B. Metevier, S. Giguere, S. Brockman, A. Kobren, Y. Brun, E. Brunskill, and P. Thomas. Offline contextual bandits with high probability fairness guarantees. *Advances in neural information processing systems*, 32, 2019.

[28] J. Miller, J. C. Perdomo, and T. Zrnic. Outside the echo chamber: Optimizing the performative risk. *arXiv preprint arXiv:2102.08570*, 2021.

[29] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli. A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on intelligent vehicles*, 1(1):33–55, 2016.

[30] M. J. Pazzani and D. Billsus. Content-based recommendation systems. In *The adaptive web*, pages 325–341. Springer, 2007.

[31] J. C. Perdomo, T. Zrnic, C. Mendler-Dünner, and M. Hardt. Performative prediction. *arXiv preprint arXiv:2002.06673*, 2020.

[32] C. Riquelme, G. Tucker, and J. Snoek. Deep bayesian bandits showdown: An empirical comparison of bayesian deep networks for thompson sampling. *arXiv preprint arXiv:1802.09127*, 2018.

[33] Y. Russac, L. Faury, O. Cappé, and A. Garivier. Self-concordant analysis of generalized linear bandits with forgetting. In *International Conference on Artificial Intelligence and Statistics*, pages 658–666. PMLR, 2021.

[34] D. Sculley. Practical learning from one-sided feedback. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 609–618, 2007.

[35] R. Sen, A. Rakhlin, L. Ying, R. Kidambi, D. Foster, D. Hill, and I. Dhillon. Top-k extreme contextual bandits with arm hierarchy. *arXiv preprint arXiv:2102.07800*, 2021.

[36] A. Srivastava, A. Kundu, S. Sural, and A. Majumdar. Credit card fraud detection using hidden markov model. *IEEE Transactions on dependable and secure computing*, 5(1):37–48, 2008.

[37] A. K. Tiwari. Machine learning application in loan default prediction. *Machine Learning*, 4(5), 2018.

[38] N. Tollenaar and P. Van der Heijden. Which method predicts recidivism best?: a comparison of statistical, machine learning and data mining predictive models. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 176(2):565–584, 2013.

[39] C.-F. Tsai and M.-L. Chen. Credit rating by hybrid machine learning techniques. *Applied soft computing*, 10(2):374–380, 2010.

[40] P. Wang, R. Mathieu, J. Ke, and H. Cai. Predicting criminal recidivism with support vector machine. In *2010 International Conference on Management and Service Science*, pages 1–9. IEEE, 2010.

[41] P. Xu, Z. Wen, H. Zhao, and Q. Gu. Neural contextual bandits with deep representation and shallow exploration. *arXiv preprint arXiv:2012.01780*, 2020.

[42] D. Zhou, L. Li, and Q. Gu. Neural contextual bandits with ucb-based exploration. In *International Conference on Machine Learning*, pages 11492–11502. PMLR, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] **Yes, in our societal impact section we discuss limitations of our work.**

(c) Did you discuss any potential negative societal impacts of your work? [Yes] **We have added a section on the broader impact of our method.**

(d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...

(a) Did you state the full set of assumptions of all theoretical results? [Yes]

(b) Did you include complete proofs of all theoretical results? [Yes] **The appendix contains detailed proofs of our theoretical claims**

3. If you ran experiments...

(a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] **Our plots include error bars for 5 runs.**

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] **We described the types of GPUs used, as well as the parallelism used in our method.**

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes] We cited the MNIST and Adult dataset creators.

(b) Did you mention the license of the assets? [N/A]

(c) Did you include any new assets either in the supplemental material or as a URL? [Yes] **We have included our code in the supplemental material.**

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] **The datasets we used are public and previously published**

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] **The datasets we used are public and previously published, and do not have PII/offensive content**

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] **No crowdsourcing or human subjects.**

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]