Efficient Bandit Algorithms for Online Multiclass Prediction

Sham M. Kakade SHAM@TTI-C.ORG
Shai Shalev-Shwartz SHAI@TTI-C.ORG
Ambuj Tewari TEWARI@TTI-C.ORG
Toyota Technological Institute, 1427 East 60th Street, Chicago, Illinois 60637, USA

Keywords: Online learning, Bandit algorithms, Perceptron

Abstract

This paper introduces the Banditron, a variant of the Perceptron [Rosenblatt, 1958], for the multiclass bandit setting. The multiclass bandit setting models a wide range of practical supervised learning applications where the learner only receives partial feedback (referred to as "bandit" feedback, in the spirit of multi-armed bandit models) with respect to the true label (e.g. in many web applications users often only provide positive "click" feedback which does not necessarily fully disclose a true label). The Banditron has the ability to learn in a multiclass classification setting with the "bandit" feedback which only reveals whether or not the prediction made by the algorithm was correct or not (but does not necessarily reveal the true label). We provide (relative) mistake bounds which show how the Banditron enjoys favorable performance, and our experiments demonstrate the practicality of the algorithm. Furthermore, this paper pays close attention to the important special case when the data is linearly separable — a problem which has been exhaustively studied in the full information setting yet is novel in the bandit setting.

1. Introduction

In the conventional supervised learning paradigm, the learner has access to a data set in which the true labels of the inputs are provided. While attendant learning algorithms in this paradigm are enjoying wide ranging success, their effective application to a number of domains, including many web based applications, hinges on being able to learn in settings where the true labels are not fully disclosed, but rather the learning algorithm only receives some partial feedback. Important domains include both the (financially important) sponsored advertising on webpages and recommender systems. The typical setting is: first, a user queries the system; then using the query and other potentially rich knowledge the system has about the user (e.g. past purchases, their browsing history, etc.) the system makes a suggestion (e.g. it presents the user with a few ads they might click on or songs they might buy); finally, the user either positively or negatively responds to the suggestion. Crucially, the system does not learn what would have happened had other suggestions been presented.

We view such problems as naturally being online, "bandit" versions of multiclass prediction problems, and, in this paper, we formalize such a model. In essence, this multiclass bandit problem is as follows: at each round, the learner receives an input $\mathbf{x}$ (say the users query, profile, and other high dimensional information); the learner predicts some class label $\hat{y}$ (the suggestion); then the learner receives the limited feedback of only whether the chosen label was correct or not. In the conventional, "full information" supervised learning model, a true label y (possibly more than one or none at all) is revealed to the learner at each round — clearly unrealistic in the aforementioned applications. In both cases, the learner desires to make as few mistakes as possible. The bandit version of this problem is clearly more challenging, since, in addition to the issues ones faces for supervised learning (e.g. learning a mapping from a high dimensional input space to the label space), one also faces balancing exploration and exploitation.

This paper considers the workhorse of hypothesis

Appearing in *Proceedings of the 25th International Conference on Machine Learning*, Helsinki, Finland, 2008. Copyright 2008 by the author(s)/owner(s).

spaces, namely linear predictors, in the bandit setting. Somewhat surprisingly, while there has been a staggering number of results on (margin based) linear predictors and much recent work on bandit models, the intersection of these two settings is novel and opens a number of interesting (both theoretical and practical) questions, which we consider. In particular, we pay close attention to the important case where the data are linearly separable, where, in the full information setting, the (efficient) Perceptron algorithm makes a number of mistakes that is asymptotically bounded (so the actual error rate will rapidly converge to 0).

There are a number of related results in the bandit literature. The Exp4 algorithm (for the "experts" setting) of Auer et al. [1998] and the contextual bandit setting of Langford and Zhang [2007] are both bandit settings where the learner has side information (e.g. the input "x") when making a decision — in fact, our setting can be thought of as a special case of the contextual bandit setting[1]. However, these settings consider abstract hypothesis spaces and do not explicitly consider efficient algorithms. Technically related are the bandit algorithms for online convex optimization of Flaxman et al. [2005], Kleinberg [2004], which attempt to estimate a gradient (for optimization) with only partial feedback. However, these algorithms do not apply due to the subtleties of using the non-convex classification loss, which we discuss at the end of Section 2.

This paper provides an *efficient* bandit algorithm, the Banditron, for multiclass prediction using linear hypothesis spaces, which enjoys a favorable mistake bound. We provide empirical results showing our algorithm is quite practical. For the case where the data is linearly separable, our mistake bound is $O(\sqrt{T})$ in T rounds. We also provide results toward characterizing the optimal achievable mistake bound for the linearly separable case (ignoring efficiency issues here) and introduce some important open questions regarding this issue. In the Extensions section, we also discuss update rules which generalize the Winnow algorithm (for L1 margins) and margin-mistake based algorithms to the bandit setting. We also discuss how our algorithm can be extended to ranking and settings where more than one prediction $\hat{y}$ can be presented to the user (e.g. an advertising setting where multiple ads may be presented).

2. Problem Setting

We now formally define the problem of online multiclass prediction in the bandit setting. Online learning is performed in a sequence of consecutive rounds. On round t, the learner is given an instance vector $\mathbf{x}_t \in \mathbb{R}^d$ and is required to predict a label out of a set of k predefined labels which we denote by $[k] = \{1, \ldots, k\}$. We denote the predicted label by $\hat{y}_t$. In the full information case, after predicting the label, the learner receives the correct label associated with $\mathbf{x}_t$, which we denote by $y_t \in [k]$. We consider a bandit setting, in which the feedback received by the learner is $\mathbf{1}[\hat{y}_t \neq y_t]$, where $\mathbf{1}[\pi]$ is 1 if predicate π holds and 0 otherwise. That is, the learner knows if it predicted an incorrect label, but it does not know the identity of the correct label. The learner's ultimate goal is to minimize the number of prediction mistakes, M, it makes along its run, where:

$$M = \sum_{t=1}^{T} \mathbf{1}[\hat{y}_t \neq y_t] \ .$$

To make M small, the learner may update its prediction mechanism after each round so as to be more accurate in later rounds.

The prediction of the algorithm at round t is determined by a hypothesis, $h_t : \mathbb{R}^d \to [k]$, where h_t is taken from a class of hypotheses $\mathcal{H}$. In this paper we focus on the class of margin based linear hypotheses. Formally, each $h \in \mathcal{H}$ is parameterized by a matrix of weights $W \in \mathbb{R}^{k \times d}$ and is defined to be:

$$h(\mathbf{x}) \ = \ \underset{j \in [k]}{\mathrm{argmax}} \ (W\mathbf{x})_j \ , \tag{1}$$

where $(W\mathbf{x})_j$ is the jth element of the vector obtained by multiplying the matrix W with the vector $\mathbf{x}$. Since each hypothesis is parameterized by a weight matrix, we refer to a matrix W also as a hypothesis — by that we mean that the prediction is defined as given in Eq. (1). To evaluate the performance of a weight matrix W on an example $(\mathbf{x}, y)$ we check whether W makes a prediction mistake, namely determine if $\arg\max_j (W\mathbf{x})_j \neq y$.

The class of margin based linear hypotheses for multiclass learning has been extensively studied in the full information case [Duda and Hart, 1973, Vapnik, 1998, Weston and Watkins, 1999, Elisseeff and Weston, 2001, Crammer and Singer, 2003]. Our starting point is a simple adaptation of the Perceptron algorithm [Rosenblatt, 1958] for multiclass prediction in the full information case (this adaptation is called Kesler's construction in [Duda and Hart, 1973, Crammer and Singer, 2003]). Despite its age and simplicity,

[1] The contextual bandit setting can be thought of as a general cost sensitive classification problem with bandit feedback. While their setting is an i.i.d. one, we make no statistical assumptions.

the Perceptron has proven to be quite effective in practical problems, even when compared to state-of-the-art large margin algorithms [Freund and Schapire, 1999]. We denote by W^t the weight matrix used by the Perceptron at round t. The Perceptron starts with the all zero matrix $W^1 = \mathbf{0}$ and updates it as follows

$$W^{t+1} \;=\; W^t + U^t \quad , \tag{2}$$

where $U^t \in \mathbb{R}^{k\times d}$ is the matrix defined by

$$U^t_{r,j} \;=\; x_{t,j}\left(\mathbf{1}[y_t = r] - \mathbf{1}[\hat{y}_t = r]\right) \; . \tag{3}$$

In other words, if there is no prediction mistake (i.e. $y_t = \hat{y}_t$), then there is no update (i.e. $W^{t+1} = W^t$), and if there is a prediction mistake, then $\mathbf{x}_t$ is added to the y_tth row of the weight matrix and subtracted from the $\hat{y}_t$th row of the matrix.

A relative mistake bound can be proven for the multiclass Perceptron algorithm. The difficulty with providing mistake bounds for any (efficient) algorithm in this setting stems from the fact that the classification loss is non-convex. Hence, performance bounds are commonly evaluated using the multiclass *hinge-loss* — what might be thought of as a convex relaxation of the classification loss. In particular, the hinge-loss of W on $(\mathbf{x}, y)$ is defined as follows:

$$\ell(W;(\mathbf{x},y)) \;=\; \max_{r\in[k]\setminus\{y\}} \left[1 - (W\mathbf{x})_y + (W\mathbf{x})_r\right]_+ \quad , \tag{4}$$

where $[a]_+ = \max\{a, 0\}$ is the hinge function. The hinge-loss will be zero only if $(W\mathbf{x})_y - (W\mathbf{x})_r \geq 1$ for all $r \neq y$. The difference $(W\mathbf{x})_y - (W\mathbf{x})_r$ is a generalization of the notion of *margin* from binary classification. Let $\hat{y} = \mathrm{argmax}_r (W\mathbf{x})_r$ be the prediction of W. Note that if $\hat{y} \neq y$ then $\ell(W;(\mathbf{x},y)) \geq 1$. Thus, the hinge-loss is a convex upper bound on the zero-one loss function, $\ell(\mathbf{w};(\mathbf{x},y)) \geq \mathbf{1}[\hat{y} \neq y]$.

The Perceptron mistake bound holds for any sequence of examples and compares the number of mistakes made by the Perceptron with the cumulative hinge-loss of any fixed weight matrix $W^\star$, even one defined with prior knowledge of the sequence. Formally, let $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_T, y_T)$ be a sequence of examples and assume for simplicity that $\|\mathbf{x}_t\| \leq 1$ for all t. Let $W^\star$ be any fixed weight matrix. We denote by

$$L = \sum_{t=1}^{T} \ell(W^\star;(\mathbf{x}_t, y_t)) \quad , \tag{5}$$

the cumulative hinge-loss of $W^\star$ over the sequence of examples and by

$$D = 2\,\|W^\star\|_F^2 = 2\sum_{r=1}^{k}\sum_{j=1}^{d}(W^\star_{i,j})^2 \quad , \tag{6}$$

the *complexity* of $W^\star$. Here $\|\cdot\|_F^2$ denotes the Frobenius norm. Then the number of prediction mistakes of the multiclass Perceptron is at most,

$$M \;\leq\; L + D + \sqrt{L\,D} \quad . \tag{7}$$

Algorithm 1 The Banditron

```
Parameters: $\gamma \in (0, 0.5)$
Initialize $W^1 = \mathbf{0} \in \mathbb{R}^{k\times d}$
for $t = 1, 2, \ldots, T$ do
  Receive $\mathbf{x}_t \in \mathbb{R}^d$
  Set $\hat{y}_t = \arg\max_{r\in[k]}(W^t\mathbf{x}_t)_r$
  $\forall r \in [k]$ define $P(r) = (1-\gamma)\mathbf{1}[r = \hat{y}_t] + \frac{\gamma}{k}$
  Randomly sample $\tilde{y}_t$ according to $P$
  Predict $\tilde{y}_t$ and receive feedback $\mathbf{1}[\tilde{y}_t = y_t]$
  Define $\tilde{U}^t \in \mathbb{R}^{k\times d}$ such that:
    $\tilde{U}^t_{r,j} = x_{t,j}\left(\frac{\mathbf{1}[y_t=\tilde{y}_t]\mathbf{1}[\tilde{y}_t=r]}{P(r)} - \mathbf{1}[\hat{y}_t = r]\right)$
  Update: $W^{t+1} = W^t + \tilde{U}^t$
end for
```

A proof of the above mistake bound can be found for example in Fink et al. [2006]. The mistake bound in Eq. (7) consists of three terms: the loss of $W^\star$, the complexity of $W^\star$, and a sub-linear term which is often negligible. In particular, when the data is separable (i.e. $L = 0$), the number of mistakes is bounded by D.

Unfortunately, the Perceptron's update cannot be implemented in the bandit setting as we do not know the identity of y_t. One direction is to work directly with the hinge-loss (which is convex) and try to use the bandit algorithms for online convex optimization of Flaxman et al. [2005], Kleinberg [2004]. In this work, they attempt to find an unbiased estimate of the gradient using only bandit feedback (i.e. using only the loss received as feedback). However, since the only feedback the learner receives is $\mathbf{1}[\hat{y}_t \neq y_t]$, one does not necessarily even know the hinge-loss for the chosen decision, $\hat{y}_t$, due to dependence of the hinge loss on the true label y_t. Hence, the results of Flaxman et al. [2005], Kleinberg [2004] are not directly applicable.

3. The Banditron

We now present the Banditron in Algorithm 1, which is an adaptation of the multiclass Perceptron for the bandit case.

Similar to the Perceptron, at each round we let $\hat{y}_t$ be the best label according to the current weight matrix W^t, i.e. $\hat{y}_t = \mathrm{argmax}_r (W^t\mathbf{x}_t)_r$. Most of the time the Banditron exploits the quality of the current weight matrix by predicting the label $\hat{y}_t$. Unlike the Perceptron, if $\hat{y}_t \neq y_t$, then we can not make an update since we are blind to the identity of y_t. Roughly speaking, it is difficult to learn when we exploit using W^t. For this reason, on some of the rounds we let the algorithm explore (with probability $1-\gamma$) and uniformly predict a random label from $[k]$. We denote by $\tilde{y}_t$ the predicted label. On rounds in which we explore, (so $\tilde{y}_t \neq \hat{y}_t$), if we additionally receive a positive feedback, i.e. $\tilde{y}_t = y_t$, then we indirectly obtain the full information regarding the identity of y_t, and we can therefore update our weight matrix using this positive instance. The parameter γ controls the exploration-exploitation tradeoff.

The above intuitive argument is formalized by defining the update matrix $\tilde{U}^t$ to be a function of the randomized prediction $\tilde{y}_t$. We emphasize that $\tilde{U}^t$ accesses the correct label y_t only through the indicator $\mathbf{1}[y_t = \tilde{y}_t]$ and is thus adequate for the Bandit setting. As we show later in Lemma 4, the expected value of the Banditron's update matrix $\tilde{U}^t$ is exactly the Perceptron's update matrix U^t. While there a number of other variants which also perform unbiased updates, we have found this one provides the most favorable performance (empirically speaking).

The following theorem provides a bound on the expected number of mistakes the Banditron makes.

Theorem 1. *(Mistake Bound). Assume that for the sequence of examples, $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_T, y_T)$, we have, for all t, $\mathbf{x}_t \in \mathbb{R}^d$, $\|\mathbf{x}_t\| \leq 1$, and $y_t \in [k]$. Let $W^\star$ be any matrix, let L be the cumulative hinge-loss of $W^\star$ as defined in Eq. (5), and let D be the complexity of $W^\star$ (i.e. $D = 2\|W^\star\|_F^2$). Then the number of mistakes M made by the Banditron satisfies*

$$\mathbb{E}[M] \;\leq\; L + \gamma T + 3\max\left\{\tfrac{k\,D}{\gamma}\,,\,\sqrt{D\,\gamma\,T}\right\} + \sqrt{\tfrac{k\,D\,L}{\gamma}} \quad .$$

where expectation is taken with respect to the randomness of the algorithm.

Before turning to the proof of Thm. 1 let us first optimize the exploration-exploitation parameter γ in different scenarios. First, assume that the data is separable, that is $L = 0$. In this case, we can obtain a mistake bound of $O(\sqrt{T})$. In fact the following corollary shows that an $O(\sqrt{T})$ bound is achievable whenever the cumulative hinge-loss of $W^\star$ is small enough.

Corollary 2 (Low noise)**.** *Assume that the conditions stated in Thm. 1 hold and that there exists $W^\star$ with fixed complexity D and loss $L \leq O(\sqrt{D\,k\,T})$. Then, by setting $\gamma = \sqrt{k\,D/T}$ we obtain the bound $\mathbb{E}[M] \leq O(\sqrt{k\,D\,T})$.*

Next, let us consider the case where we have a constant (average) noise level of ρ, i.e. there exists $\rho \in (0,1)$ such that $L \leq \rho T$. In this case,

Corollary 3 (High noise)**.** *Assume that the conditions stated in Thm. 1 hold and that there exists $W^\star$ with fixed complexity D and loss $L \leq \rho\,T$ for a constant $\rho \in (0,1)$. Then, by setting $\gamma = \rho\,(k\,D/T)^{1/3}$ we obtain the bound $\mathbb{E}[M] \leq \rho T\,(1+\epsilon)$ where $\epsilon = O((k\,D)^{1/3}\,T^{-1/3})$.*

We note that the bound in the above corollary can be also written in an additive form as: $\mathbb{E}[M] - L \leq O(T^{2/3})$. However, since we are not giving proper regret bounds as we compare mistakes to hinge-loss we prefer to directly bound $\mathbb{E}[M]$.

Analysis: To prove Thm. 1 we first show that the random matrix $\tilde{U}^t$ is an unbiased estimator of the update matrix U^t used by the Perceptron. Formally, let $\mathbb{E}_t[\tilde{U}^t]$ be the expected value of $\tilde{U}^t$ conditioned on $\tilde{y}_1, \ldots, \tilde{y}_{t-1}$. Then:

Lemma 4. *Let $\tilde{U}^t$ be as defined in Algorithm 1 and let U^t be as defined in Eq. (3). Then, $\mathbb{E}_t[\tilde{U}^t] = U^t$.*

Proof. For each $r \in [k]$ and $j \in [d]$ we have

$$\begin{aligned}\mathbb{E}_t[\tilde{U}^t_{r,j}] &= \sum_{i=1}^{k} P(i)x_{t,j}\left(\tfrac{\mathbf{1}[i=y_t]\mathbf{1}[i=r]}{P(r)} - \mathbf{1}[\hat{y}_t = r]\right)\\ &= x_{t,j}\left(\mathbf{1}[y_t = r] - \mathbf{1}[\hat{y}_t = r]\right) \;=\; U^t_{r,j} \quad ,\end{aligned}$$

which completes the proof. □

Next, we bound the expected squared norm of $\tilde{U}^t$.

Lemma 5. *Let $\tilde{U}^t$ be as defined in Algorithm 1. Then,*

$$\mathbb{E}_t[\,\|\tilde{U}^t\|_F^2\,] \leq 2\,\|\mathbf{x}_t\|^2\left(\tfrac{k}{\gamma}\,\mathbf{1}[y_t \neq \hat{y}_t] + \gamma\,\mathbf{1}[y_t = \hat{y}_t]\right) \ .$$

Proof. We first observe that

$$\|\tilde{U}^t\|_F^2 \;=\; \begin{cases}\|\mathbf{x}_t\|^2\left(\frac{1}{P(y_t)^2}+1\right) & \text{if } \tilde{y}_t = y_t \neq \hat{y}_t\\ \|\mathbf{x}_t\|^2\left(\frac{1}{P(y_t)}-1\right)^2 & \text{if } \tilde{y}_t = y_t = \hat{y}_t\\ \|\mathbf{x}_t\|^2 & \text{if } \tilde{y}_t \neq y_t\end{cases}$$

Therefore, if $y_t \neq \hat{y}_t$ then

$$\begin{aligned}\frac{\mathbb{E}_t[\|\tilde{U}^t\|_F^2]}{\|\mathbf{x}_t\|^2} &= P(y_t)\left(\tfrac{1}{P(y_t)^2}+1\right) + (1-P(y_t))\\ &= 1 + \tfrac{1}{P(y_t)} = 1 + \tfrac{k}{\gamma} \leq \tfrac{2k}{\gamma}\ ,\end{aligned}$$

and if $y_t = \hat{y}_t$ then

$$\begin{aligned}\frac{\mathbb{E}_t[\|\tilde{U}^t\|_F^2]}{\|\mathbf{x}_t\|^2} &= P(y_t)\left(\tfrac{1}{P(y_t)}-1\right)^2 + (1-P(y_t))\\ &= \tfrac{1}{P(y_t)} - 1 \leq \tfrac{1}{1-\gamma} - 1 \leq \tfrac{\gamma}{1-\gamma} \leq 2\gamma\ .\end{aligned}$$

Combining the two cases concludes our proof. □

Equipped with the above two lemmas, we are ready to prove Thm. 1.

Proof of Thm. 1. Throughout the proof we use the notation $\langle W^\star, W^t \rangle := \sum_{r=1}^{k} \sum_{j=1}^{d} W^\star_{r,j} W^t_{r,j}$. We prove the theorem by bounding $\mathbb{E}[\langle W^\star, W^{T+1} \rangle]$ from above and from below starting with a lower bound. We can first use the fact that $W^1 = \mathbf{0}$ to rewrite $\mathbb{E}[\langle W^\star, W^{T+1} \rangle]$ as $\sum_{t=1}^{T} \Delta_t$ where

$$\Delta_t := \mathbb{E}[\langle W^\star, W^{t+1} \rangle] - \mathbb{E}[\langle W^\star, W^t \rangle] \ .$$

Expanding the definition of W^{t+1} and using Lemma 4 we obtain that for all t, $\Delta_t = \mathbb{E}[\langle W^\star, \tilde{U}^t \rangle] = \mathbb{E}[\langle W^\star, U^t \rangle]$. Next, we note that the definition of the hinge-loss given in Eq. (4) implies that the following holds regardless of the value of $\hat{y}_t$

$$\ell(W^\star, (\mathbf{x}_t, y_t)) \ \geq \ \mathbf{1}[\hat{y}_t \neq y_t] - \langle W^\star, U^t \rangle \ .$$

Therefore $\Delta_t \ \geq \ \mathbb{E}[\,\mathbf{1}[\hat{y}_t \neq y_t]\,] - \ell(W^\star, (\mathbf{x}_t, y_t))$. Summing over t we obtain the lower bound

$$\mathbb{E}[\langle W^\star, W^{T+1} \rangle] \ = \ \sum_{t=1}^{T} \Delta_t \ \geq \ \mathbb{E}[\hat{M}] - L \ , \qquad (8)$$

where $\hat{M} := \sum_{t=1}^{T} \mathbf{1}[\hat{y}_t \neq y_t]$ and L is as defined in Eq. (5). Next, we show an upper bound on $\mathbb{E}[\langle W^\star, W^{T+1} \rangle]$. Using Cauchy-Schwartz inequality we have $\langle W^\star, W^{T+1} \rangle \ \leq \ \|W^\star\|_F \, \|W^{T+1}\|_F$. To ease our notation, we use the shorthand $\|\cdot\|$ for denoting the Frobenius norm. Using the definition of D given in Eq. (6), the concavity of the sqrt function, and Jensen's inequality we obtain that

$$\mathbb{E}[\langle W^\star, W^{T+1} \rangle] \leq \sqrt{\tfrac{D\,\mathbb{E}[\,\|W^{T+1}\|^2\,]}{2}} \ . \qquad (9)$$

We therefore need to upper bound the expected value of $\|W^{T+1}\|^2$. Expanding the definition of W^{T+1} we get that

$$\begin{aligned} \mathbb{E}[\|W^{T+1}\|^2] \ &= \ \mathbb{E}[\|W^T\|^2 + \langle W^T, \tilde{U}^T \rangle + \|\tilde{U}^T\|^2] \\ &= \ \sum_{t=1}^{T} \Big(\mathbb{E}[\langle W^t, \tilde{U}^t \rangle] + \mathbb{E}[\,\|\tilde{U}^t\|^2\,] \Big) \ . \end{aligned}$$

Using Lemma 4 we obtain that $\mathbb{E}[\langle W^t, \tilde{U}^t \rangle] = \mathbb{E}[\langle W^t, U^t \rangle] \leq 0$, where the second inequality follows from the definition of U^t and $\hat{y}_t$. Combining this with Lemma 5 and with the assumption $\|\mathbf{x}_t\| \leq 1$ for all t we obtain that

$$\begin{aligned} \mathbb{E}[\|W^{T+1}\|^2] \ &\leq \ \sum_{t=1}^{T} \mathbb{E}\left(\tfrac{2k}{\gamma} \mathbf{1}[y_t \neq \hat{y}_t] + 2\gamma \mathbf{1}[y_t = \hat{y}_t] \right) \\ &\leq \ \tfrac{2k}{\gamma}\, \mathbb{E}[\hat{M}] + 2\,\gamma\,T \ . \end{aligned}$$

Plugging the above into Eq. (9) and using the inequality $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$ we get the upper bound

$$\mathbb{E}[\langle W^\star, W^{T+1} \rangle] \leq \sqrt{\tfrac{D\,k\,\mathbb{E}[\hat{M}]}{\gamma}} + \sqrt{D\,\gamma\,T} \ .$$

Comparing the above upper bound with the lower bound given in Eq. (8) and rearranging terms yield

$$\mathbb{E}[\hat{M}] - \sqrt{\tfrac{D\,k\,\mathbb{E}[\hat{M}]}{\gamma}} - \left(L + \sqrt{D\,\gamma\,T} \right) \leq 0 \ .$$

Standard algebraic manipulations give the bound

$$\mathbb{E}[\hat{M}] \ \leq \ L + \sqrt{\tfrac{D\,k\,L}{\gamma}} + 3 \max\left\{ \tfrac{D\,k}{\gamma} \, , \, \sqrt{D\,\gamma\,T} \right\} \ .$$

Finally, our proof is concluded by noting that in expectation we are exploring no more than γT of the rounds and thus $\mathbb{E}[M] \leq \mathbb{E}[\hat{M}] + \gamma T$. □

4. Mistake Bounds Under Separability

In this section we present results towards characterizing the optimal achievable rate for the case where the data is separable. Here, in the full-information setting, the mistake bound of the Perceptron algorithm is finite and bounded by D. We now present an (inefficient) algorithm showing that the achievable mistake bound in the bandit setting is also finite — thus the Banditron's mistake bound of $O(\sqrt{T})$ leaves significant room for improvement (though the algorithm is quite simple and has reasonable performance, which we demonstrate in the next section).

First, as a technical tool, we make the interesting observation that the halving algorithm (generalized to the multiclass setting) is also applicable to the bandit setting. The algorithm is as follows: Let $\mathcal{H}'$ be the current set of "active" experts, which is initialized to the full set, i.e. $\mathcal{H} = \mathcal{H}'$ at $t = 1$. At each round t, we predict using the majority prediction r (i.e. the prediction $r \in [k]$ which the most hypotheses in $\mathcal{H}'$ predict). If we are correct, we make no update. If we are incorrect, we remove from the active set, $\mathcal{H}'$, those $h \in \mathcal{H}'$ which predicted the incorrect label r. Crucially, this (generalized) halving algorithm is implementable with only the bandit feedback that we receive. This algorithm enjoys the following mistake bound.

Lemma 6. *(Halving Algorithm). The halving algorithm (in the bandit setting) makes at most $k \ln |\mathcal{H}|$ mistakes on any sequence in which there exists some hypothesis in $\mathcal{H}$ which makes no mistakes.*

Proof. Whenever the algorithm makes a mistake, the size of active set is reduced by at least a $1 - 1/k$

fraction, since majority prediction uses a fraction of hypothesis (from the active set) that is at least $1/k$. Since the algorithm never removes a perfect hypothesis from the active set, the maximal number of mistakes, M, that can occur until $\mathcal{H}'$ consists of only perfect hypotheses satisfies $(1-1/k)^M|\mathcal{H}| \geq 1$. Using the inequality $(1-1/k) \leq e^{-1/k}$ and solving for M leads to the claim. □

Using this, the following theorem shows that the achievable bound for the number of mistakes is asymptotically finite. Unfortunately, the result has a dimensionality dependence on d. The algorithm essentially uses the margin condition to construct an appropriately sized cover for $\mathcal{H}$, the set of all linear hypotheses, and runs the halving algorithm on this cover.

Theorem 7. *There exists a deterministic algorithm (in the bandit setting), taking D as input, which makes at most $O(k^2 d \ln(Dd))$ mistakes on any sequence (where $\|\mathbf{x}_t\| \leq 1$) that is linearly separable at margin 1 by some $W^\star$, with $2||W^\star||_F^2 \leq D$.*

Proof. (sketch) Since the margin is 1, it is straightforward to show that if W is a perturbation of $W^\star$ which satisfies $||W^\star - W||_\infty \leq O(\frac{1}{\sqrt{d}})$, then the data is still linearly separable under W. By noting that each coordinate in $W^\star$ is (rather crudely) bounded by $\sqrt{D}$, there exists a discretized grid of $\mathcal{H}$ of size $O(\sqrt{Dd})^{kd}$ which contains a linear separator. The algorithm simply runs the halving algorithm on this cover. □

This result is in stark contrast to the Perceptron mistake bound which has no dependence on the dimension d. We now provide a mistake bound with no dependence on the dimension. Unfortunately, it is not asymptotically finite, as it is has a rather mild dependence on the time — it is $O(D \ln T)$ (ignoring k and higher order terms), while the Perceptron mistake bound is $O(D)$.

Theorem 8. *There exists a randomized algorithm (in the bandit setting), taking as inputs D, T and $\delta > 0$, such that with probability greater than $1-\delta$ the algorithm makes at most $O(k^2 D \ln \frac{T+k}{\delta}(\ln D + \ln\ln \frac{T+k}{\delta}))$ mistakes on any T length sequence (where $\|\mathbf{x}_t\| \leq 1$) that is linearly separable at margin 1 by some $W^\star$, with $2||W^\star||_F^2 \leq D$.*

The algorithm first constructs a random projection operator which projects any $\mathbf{x}$ into a space of dimension $d' = O(D \ln \frac{T+k}{\delta})$, and then it runs the previous algorithm in this lower dimensional space. The proof essentially consists of using the results in Arriaga and Vempala [2006] to argue that the (multiclass) margin is preserved under this random projection.

Proof. (sketch) It is simpler to rescale $W^\star$ such that $||W^\star||_F = 1$ and the margin is $1/\sqrt{D}$. Consider the $T+k$ points x_1 to x_T and the (row) vectors $W_1^\star, \ldots W_k^\star$, whose norms are all bounded by 1. Let P be a matrix of dimension $d' \times d$, where each entry of P is independently sampled from $U(-1,1)$. Define the projection operator $\Pi(v) = \frac{1}{\sqrt{d'}}Pv$. Corollary 2 of Arriaga and Vempala [2006] (essentially a result from the JL lemma) shows that if $d' = O(D \ln \frac{T+k}{\delta})$ then this projection additively preserves the inner products of these points up to $\frac{1}{3\sqrt{D}}$, i.e. $|\Pi(W_r^\star) \cdot \Pi(\mathbf{x}_t) - W_r^\star \cdot \mathbf{x}_t| \leq \frac{1}{3\sqrt{D}}$. It follows that, after the projection, the data is linearly separable with margin at least $\frac{1}{3\sqrt{D}}$. Letting $\Pi(W^\star)$ denote the matrix where each row of $W^\star$ has been projected, then, also by the JL lemma, the norm $||\Pi(W^\star)||_F$ will (rather crudely) be bounded by 2 (recall $||W^\star||_F = 1$). Hence, the projected data is linearly separable at margin $1/(3\sqrt{D})$ by a weight matrix that has norm $O(1)$, which is identical to being separable at margin 1 with weight vector of complexity $O(D)$. The algorithm is to first create a random projection matrix (which can be constructed without knowledge of the sequence) and then we can run the previous algorithm on the lower dimensional space d'. Since we have shown that the margin is preserved (up to a constant) in the lower dimensional space, the result follows from the previous Theorem 7, with d' as the dimensionality. □

We discuss open questions in the Extensions section.

5. Experiments

In this section, we report some experimental results for the Banditron algorithm on synthetic and real world data sets. For each data set, we ran Banditron for a wide range of values of the exploration parameter γ. For each value of γ, we report the average error rate, where the averaging is over 10 independent runs of Banditron.

The results are shown on Fig. 1. Each column corresponds to one data set. The top figures plot the error rates of Banditron (for the best value of γ) and Perceptron as a function of the number of examples seen. We show these on a log-log scale to get a better visual indication of the asymptotics of these algorithms. The bottom figures plot the final error rates on the complete data set as a function of γ. As expected, setting γ too low or too high leads to higher error rates.

The first data set, denoted by SynSep, is a 9-class, 400-dimensional synthetic data set of size 10^6. The idea is to have a simple simulation of generating a text

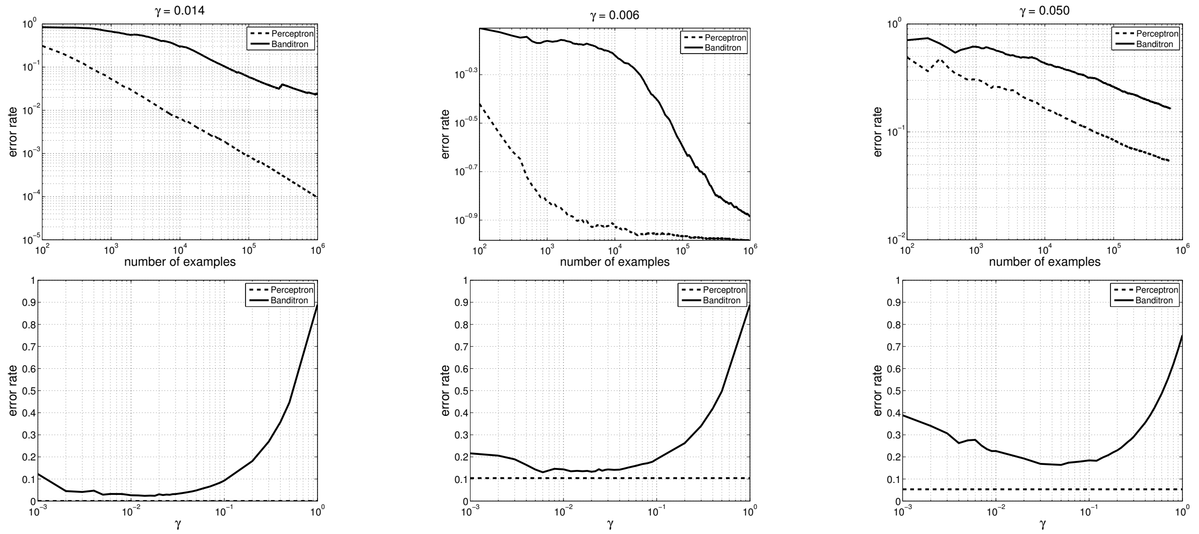

Figure 1. Error rates of Perceptron (dashed) and Banditron (solid) on the SynSep (left), SynNonSep (middle), and Reuters4 (right) data sets. The 9-class synthetic data sets are generated as follows. We fix 9 bit-vectors $v_1, \ldots, v_9 \in \{0,1\}^{400}$ each of which has 20 to 40 bits turned on in its first 120 coordinates. The supports of some of these vectors overlap. The vectors v_i correspond to 9 topics where topic i has "keywords" that correspond to the bits turned on in v_i. To generate an example, we randomly choose a v_i and randomly turn off 5 bits in its support. Further, we randomly turn on 20 additional bits in the last 280 coordinates. The last 280 coordinates thus correspond to common words that can appear in a document from any topic.

document. The coordinates represent different words in a small vocabulary of size 400. See the caption of Figure 1 for details. We ensure, by construction, that SynSep is linearly separable. The left plots in Figure 1 show the results for this data set. Since this is a separable data set, Perceptron makes a bounded number of mistakes and its error rate plot falls at a rate of $1/T$ yielding a slope of -1 on a log-log plot. Corollary 2 predicts that error rate for Banditron should decay faster than $1/\sqrt{T}$ and we indeed see a slope of approximately -0.55 on the log-log plot. The second data set, denoted by SynNonSep, is constructed in the same way as SynSep except that we introduce 5% label noise. This makes the data set non-separable. The middle plots in Fig. 1 show the results for SynNonSep. The Perceptron error rate decays till it drops to 10% and then becomes constant. Banditron does not decay appreciably till 10^4 examples after which it falls rapidly to its final value of $10^{-0.89} = 13\%$.

We construct our third data set Reuters4 from the Reuters RCV1 collection. Documents in the Reuters data set can have more than one label. We restrict ourselves to those documents that have exactly one label from the following set of labels: {ccat, ecat, gcat, mcat}. This gives us a 4-class data set of size 673,768 which includes about 84% of the documents in the original Reuters data set. We do this because the model considered in this paper assumes that every instance has a single true label. See the Extensions section for a discussion about dealing with multiple labels. We represent each document using bag-of-words, which leads to 346,810 dimensions. The right plots in Fig. 1 show the results for Reuters4. The final error rates for Perceptron and Banditron ($\gamma = 0.05$) are 5.3% and 16.3% respectively. However, it is clear from the top plot that as the number of examples grows, the error rate of Banditron is dropping at a rate comparable to that of Perceptron.

6. Extensions and Open Problems

We now discuss a few extensions of the Banditron algorithm and some open problems. These extensions may possibly improve the performance of the algorithm and also broaden the set of applications that can be tackled by our approach. Due space constraints, we confine ourselves to a rather high level overview.

Label Ranking: So far we assumed that each instance vector is associated with a *single* correct label and we must correctly predict this particular label. In many applications this binary dichotomy is inadequate as each label is associated with a degree of relevance, which reflects to what extent it is relevant to the instance vector in hand. Furthermore, it is sometime natural to predict a subset of the labels rather than a single label. For example, consider again the problem of sponsored advertising on webpages described in the Introduction. Here, the system presents the user with a few ads. If the user positively responds to one of the suggestions (say by a "click"), this implies that the user prefers this suggestion over the other suggestions, but it does not necessarily mean that the other suggestions are completely wrong.

We now briefly discuss a possible extension of the Banditron algorithm for this case (using techniques from Crammer et al. [2006]). On each round, we first find the r top ranked labels (where ranking is according to $\langle \mathbf{w}_r, \mathbf{x}_t \rangle$). With probability $1-\gamma$ we exploit and predict these labels. With probability γ we explore and randomly change one of the top ranked labels with another label which is ranked lower by our model. If we are exploring and the user chooses the replaced label, then we obtain a feedback that can be used for improving our model. The Banditron analysis can be generalized to this case, leading to bounds on the number of rounds in which the user negatively responds to our advertisement system.

Multiplicative Updates and Margin-Based Updates: While deriving the Banditron algorithm, our starting point was the Perceptron, which is an extremely simple online learning algorithm for the full information case. Over the years, many improvements of the Perceptron were suggested (see for example Shalev-Shwartz and Singer [2007] and the references therein). It is therefore interesting to study which algorithms can be adapted to the Bandit setting. We conjecture that it is relatively straightforward to adapt the multiplicative update scheme [Littlestone, 1988, Kivinen and Warmuth, 1997] to the bandit setting while achieving mistake bounds similar to the mistake bounds we derived for the Banditron. It is also possible to adapt margin-based updates (i.e. updating also when there is no prediction mistake but only a margin violation) to the bandit setting. Here, however, it seems that the resulting mistake bounds for the low noise case are inferior to the bound we obtained for the Banditron.

Achievable Rates and Open Problems: The immediate question is how to improve our rate of $O(T^{2/3})$ to $O(\sqrt{T})$ in the general setting with an efficient algorithm. We conjecture this is at least possible by some (possibly inefficient) algorithm. Important open questions in the separable case are: What is the optimal mistake bound? In particular, does there exist a finite mistake bound which has no dimensionality dependence? Furthermore, are there efficient algorithms which achieve the mistake bound of $O(D \ln T)$, provided in Theorem 8 (or better)? Practically speaking, this last question is of the most importance, as then we would have an algorithm that actually achieves a very small mistake bound in certain cases.

References

R.I. Arriaga and S. Vempala. An algorithmic theory of learning: Robust concepts and random projection. *Mach. Learn.*, 63(2), 2006.

P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. Gambling in a rigged casino: the adversarial multi-armed bandit problem. In *Proceedings of the 36th Annual FOCS*, 1998.

K. Crammer and Y. Singer. Ultraconservative online algorithms for multiclass problems. *Journal of Machine Learning Research*, 3:951–991, 2003.

K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. Online passive aggressive algorithms. *Journal of Machine Learning Research*, 7:551–585, Mar 2006.

R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis.* Wiley, 1973.

A. Elisseeff and J. Weston. A kernel method for multi-labeled classification. In *Advances in Neural Information Processing Systems 14*, 2001.

M. Fink, S. Shalev-Shwartz, Y. Singer, and S. Ullman. Online multiclass learning by interclass hypothesis sharing. In *Proceedings of the 23rd International Conference on Machine Learning*, 2006.

A. Flaxman, A. Kalai, and H. B. McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 385–394, 2005.

Y. Freund and R. E. Schapire. Large margin classification using the perceptron algorithm. *Machine Learning*, 37 (3):277–296, 1999.

J. Kivinen and M. Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, 132(1):1–64, January 1997.

R.D. Kleinberg. Nearly tight bounds for the continuum-armed bandit problem. *NIPS*, 2004.

J. Langford and T. Zhang. The epoch-greedy algorithm for contextual multi-armed bandits. *NIPS*, 2007.

N. Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2:285–318, 1988.

F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–407, 1958. (Reprinted in *Neurocomputing* (MIT Press, 1988).).

S. Shalev-Shwartz and Y. Singer. A primal-dual perspective of online learning algorithms. *Machine Learning Journal*, 2007.

V. N. Vapnik. *Statistical Learning Theory.* Wiley, 1998.

J. Weston and C. Watkins. Support vector machines for multi-class pattern recognition. In *Proceedings of the Seventh European Symposium on Artificial Neural Networks*, April 1999.