
Efficient Online Bandit Multiclass Learning with $\tilde{O}(\sqrt{T})$ Regret

Alina Beygelzimer¹ Francesco Orabona² Chicheng Zhang³

Abstract

We present an efficient second-order algorithm with $\tilde{O}(\frac{1}{\eta}\sqrt{T})^1$ regret for the bandit online multiclass problem. The regret bound holds simultaneously with respect to a family of loss functions parameterized by η , for a range of η restricted by the norm of the competitor. The family of loss functions ranges from hinge loss ($\eta = 0$) to squared hinge loss ($\eta = 1$). This provides a solution to the open problem of (Abernethy, J. and Rakhlin, A. An efficient bandit algorithm for \sqrt{T} -regret in online multiclass prediction? In *COLT*, 2009). We test our algorithm experimentally, showing that it also performs favorably against earlier algorithms.

1. Introduction

In the online multiclass classification problem, the learner must repeatedly classify examples into one of k classes. At each step t , the learner observes an example $\mathbf{x}_t \in \mathbb{R}^d$ and predicts its label $\tilde{y}_t \in [k]$. In the full-information case, the learner observes the true label $y_t \in [k]$ and incurs loss $\mathbb{1}[\tilde{y}_t \neq y_t]$. In the bandit version of this problem, first considered in (Kakade et al., 2008), the learner only observes its incurred loss $\mathbb{1}[\tilde{y}_t \neq y_t]$, i.e., whether or not its prediction was correct. Bandit multiclass learning is a special case of the general contextual bandit learning (Langford & Zhang, 2008) where exactly one of the losses is 0 and all other losses are 1 in every round.

The goal of the learner is to minimize its regret with respect to the best predictor in some reference class of predictors, that is the difference between the total number of mistakes the learner makes and the total number of mis-

takes of the best predictor in the class. Kakade et al. (2008) proposed a bandit modification of the Multiclass Perceptron algorithm (Duda & Hart, 1973), called the Banditron, that uses a reference class of linear predictors. Note that even in the full-information setting, it is difficult to provide a true regret bound. Instead, performance bounds are typically expressed in terms of the total multiclass hinge loss of the best linear predictor, a tight upper bound on 0-1 loss.

The Banditron, while computationally efficient, achieves only $O(T^{2/3})$ expected regret with respect to this loss, where T is the number of rounds. This is suboptimal as the Exp4 algorithm of Auer et al. (2003) can achieve $\tilde{O}(\sqrt{T})$ regret for the 0-1 loss, albeit very inefficiently. Abernethy & Rakhlin (2009) posed an open problem: Is there an efficient bandit multiclass learning algorithm that achieves expected regret of $\tilde{O}(\sqrt{T})$ with respect to any reasonable loss function?

The first attempt to solve this open problem was by Cramer & Gentile (2013). Using a stochastic assumption about the mechanism generating the labels, they were able to show a $\tilde{O}(\sqrt{T})$ regret, with a second-order algorithm.

Later, Hazan & Kale (2011), following a suggestion by Abernethy & Rakhlin (2009), proposed the use of the log-loss coupled with a softmax prediction. The softmax depends on a parameter that controls the smoothing factor. The value of this parameter determines the exp-concavity of the loss, allowing Hazan & Kale (2011) to prove worst-case regret bounds that range between $O(\log T)$ and $O(T^{\frac{2}{3}})$, again with a second-order algorithm. However, the choice of the smoothing factor in the loss becomes critical in obtaining strong bounds.

The original Banditron algorithm has been also extended in many ways. Wang et al. (2010) have proposed a variant based on the exponentiated gradient algorithm (Kivinen & Warmuth, 1997). Valizadegan et al. (2011) proposed different strategies to adapt the exploration rate to the data in the Banditron algorithm. However, these algorithms suffer from the same theoretical shortcomings as the Banditron.

There has been significant recent focus on developing efficient algorithms for the general contextual bandit problem (Dudík et al., 2011; Agarwal et al., 2014; Rakhlin & Sridharan, 2016; Syrgkanis et al., 2016a;b). While solving

¹Yahoo Research, New York, NY ²Stony Brook University, Stony Brook, NY ³University of California, San Diego, La Jolla, CA. Correspondence to: Alina Beygelzimer <beygel@yahoo-inc.com>, Francesco Orabona <francesco@orabona.com>, Chicheng Zhang <chichengzhang@ucsd.edu>.

¹ $\tilde{O}(\cdot)$ hides logarithmic factors.

a more general problem that does not make assumptions on the structure of the reward vector or the policy class, these results assume that contexts or context/reward pairs are generated i.i.d., or the contexts to arrive are known beforehand, which we do not assume here.

In this paper, we follow a different route. Instead of designing an ad-hoc loss function that allows us to prove strong guarantees, we propose an algorithm that simultaneously satisfies a regret bound with respect to all the loss functions in a family of functions that are tight upper bounds to the 0-1 loss. The algorithm, named Second Order Banditron Algorithm (SOBA), is efficient and based on the second-order Perceptron algorithm (Cesa-Bianchi et al., 2005). The regret bound is of the order of $\tilde{O}(\sqrt{T})$, providing a solution to the open problem of Abernethy & Rakhlin (2009).

2. Definitions and Settings

We first introduce our notation. Denote the rows of a matrix $\mathbf{V} \in \mathbb{R}^{k \times d}$ by $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k$. The vectorization of \mathbf{V} is defined as $\text{vec}(\mathbf{V}) = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k]^T$, which is a vector in \mathbb{R}^{kd} . We define the reverse operation of reshaping a $kd \times 1$ vector into a $k \times d$ matrix by $\text{mat}(\mathbf{V})$, using a row-major order. To simplify notation, we will use \mathbf{V} and $\text{vec}(\mathbf{V})$ interchangeably throughout the paper. For matrices \mathbf{A} and \mathbf{B} , denote by $\mathbf{A} \otimes \mathbf{B}$ their Kronecker product. For matrices \mathbf{X} and \mathbf{Y} of the same dimension, denote by $\langle \mathbf{X}, \mathbf{Y} \rangle = \sum_{i,j} \mathbf{X}_{i,j} \mathbf{Y}_{i,j}$ their inner product. We use $\|\cdot\|$ to denote the ℓ_2 norm of a vector, and $\|\cdot\|_F$ to denote the Frobenius norm of a matrix. For a positive definite matrix \mathbf{A} , we use $\|x\|_{\mathbf{A}} = \sqrt{\langle x, \mathbf{A}x \rangle}$ to denote the Mahalanobis norm of x with respect to \mathbf{A} . We use $\mathbf{1}_k$ to denote the vector in \mathbb{R}^k whose entries are all 1s.

We use $\mathbb{E}_{t-1}[\cdot]$ to denote the conditional expectation given the observations up to time $t-1$ and \mathbf{x}_t, y_t , that is, $\mathbf{x}_1, y_1, \tilde{y}_1, \dots, \mathbf{x}_{t-1}, y_{t-1}, \tilde{y}_{t-1}, \mathbf{x}_t, y_t$.

Let $[k]$ denote $\{1, \dots, k\}$, the set of possible labels. In our setting, learning proceeds in rounds:

For $t = 1, 2, \dots, T$:

1. The adversary presents an example $\mathbf{x}_t \in \mathbb{R}^d$ to the learner, and commits to a hidden label $y_t \in [k]$.
2. The learner predicts a label $\tilde{y}_t \sim \mathbf{p}_t$, where $\mathbf{p}_t \in \Delta^{k-1}$ is a probability distribution over $[k]$.
3. The learner receives the bandit feedback $\mathbf{1}[\tilde{y}_t \neq y_t]$.

The goal of the learner is to minimize the total number of mistakes, $M_T = \sum_{t=1}^T \mathbf{1}[\tilde{y}_t \neq y_t]$.

We will use linear predictors specified by a matrix $\mathbf{W} \in \mathbb{R}^{k \times d}$. The prediction is given by $\mathbf{W}(\mathbf{x}) = \arg \max_{i \in [k]} (\mathbf{W}\mathbf{x})_i$, where $(\mathbf{W}\mathbf{x})_i$ is the i th element of

the vector $\mathbf{W}\mathbf{x}$, corresponding to class i .

A useful notion to measure the performance of a competitor $\mathbf{U} \in \mathbb{R}^{k \times d}$ is the *multiclass hinge loss*

$$\ell(\mathbf{U}, (\mathbf{x}, y)) := \max_{i \neq y} [1 - (\mathbf{U}\mathbf{x})_y + (\mathbf{U}\mathbf{x})_i]_+, \quad (1)$$

where $[\cdot]_+ = \max(\cdot, 0)$.

3. A History of Loss Functions

As outlined in the introduction, a critical choice in obtaining strong theoretical guarantees is the choice of the loss function. In this section we introduce and motivate a family of multiclass loss functions.

In the full information setting, strong binary and multiclass mistake bounds are obtained through the use of the Perceptron algorithm (Rosenblatt, 1958). A common misunderstanding of the Perceptron algorithm is that it corresponds to a gradient descent procedure with respect to the (binary or multiclass) hinge loss. However, it is well known that the Perceptron simultaneously satisfies mistake bounds that depend on the cumulative hinge loss and also on the cumulative squared hinge loss, see for example Mohri & Ros-tamizadeh (2013). Note also that the squared hinge loss is not dominated by the hinge loss, so, depending on the data, one loss can be better than the other.

We show that the Perceptron algorithm satisfies an even stronger mistake bound with respect to the cumulative loss of any power of the multiclass hinge loss between 1 and 2.

Theorem 1. *On any sequence $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_T, y_T)$ with $\|\mathbf{x}_t\| \leq X$ for all $t \in [T]$, and any linear predictor $\mathbf{U} \in \mathbb{R}^{k \times d}$, the total number of mistakes M_T of the multiclass Perceptron satisfies, for any $q \in [1, 2]$,*

$$M_T \leq M_T^{1-\frac{1}{q}} L_{MH,q}^{\frac{1}{q}}(\mathbf{U}) + \|\mathbf{U}\|_F X \sqrt{2} \sqrt{M_T},$$

where $L_{MH,q}(\mathbf{U}) = \sum_{t=1}^T \ell(\mathbf{W}, (\mathbf{x}_t, y_t))^q$. In particular, it simultaneously satisfies the following:

$$M_T \leq L_{MH,1}(\mathbf{U}) + 2X^2 \|\mathbf{U}\|_F^2 + X \|\mathbf{U}\|_F \sqrt{2} \sqrt{L_{MH,1}(\mathbf{U})}$$

$$M_T \leq L_{MH,2}(\mathbf{U}) + 2X^2 \|\mathbf{U}\|_F^2 + X \|\mathbf{U}\|_F 2\sqrt{2} \sqrt{L_{MH,2}(\mathbf{U})}.$$

For the proof, see Appendix B.

A similar observation was done by Orabona et al. (2012) who proved a *logarithmic* mistake bound with respect to all loss functions in a similar family of functions smoothly interpolating between the hinge loss to the squared hinge loss. In particular, Orabona et al. (2012) introduced the following family of binary loss functions

$$\ell_{\eta}(x) := \begin{cases} 1 - \frac{2}{2-\eta}x + \frac{\eta}{2-\eta}x^2, & x \leq 1 \\ 0, & x > 1. \end{cases} \quad (2)$$

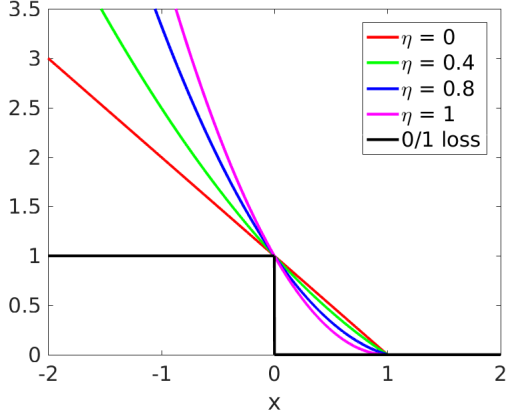


Figure 1. Plot of the loss functions in ℓ_η for different values of η .

where $0 \leq \eta \leq 1$. Note that $\eta = 0$ recovers the binary hinge loss, and $\eta = 1$ recovers the squared hinge loss. Meanwhile, for any $0 \leq \eta \leq 1$, $\ell_\eta(x) \leq \max\{\ell_0(x), \ell_1(x)\}$, and ℓ_η is an upper bound on 0-1 loss: $\mathbb{1}[x < 0] \leq \ell_\eta(x)$. See Figure 1 for a plot of the different functions in the family.

Here, we define a multiclass version of the loss in (2) as

$$\ell_\eta(\mathbf{U}, (\mathbf{x}, y)) := \ell_\eta \left((\mathbf{U}\mathbf{x})_y - \max_{i \neq y} (\mathbf{U}\mathbf{x})_i \right). \quad (3)$$

Hence, $\ell_0(\mathbf{U}, (\mathbf{x}, y)) = \ell(\mathbf{U}, (\mathbf{x}, y))$ is the classical multiclass hinge loss and $\ell_1(\mathbf{U}, (\mathbf{x}, y)) = \ell^2(\mathbf{U}, (\mathbf{x}, y))$ is the squared multiclass hinge loss.

Our algorithm has a $\tilde{O}(\frac{1}{\eta}\sqrt{T})$ regret bound that holds simultaneously for all loss functions in this family, with η in a range that ensure that $(\mathbf{U}\mathbf{x})_i - (\mathbf{U}\mathbf{x})_j \leq \frac{2-\eta}{\eta}$, $i, j \in [k]$. We also show that there exists a setting of the parameters of the algorithm that gives a mistake upper bound of $\tilde{O}((L^*T)^{1/3} + \sqrt{T})$, where L^* is the cumulative hinge loss of the competitor, which is never worse than the best bounds in Kakade et al. (2008).

4. Second Order Banditron Algorithm

This section introduces our algorithm for bandit multiclass online learning, called Second Order Banditron Algorithm (SOBA), described in Algorithm 1.

SOBA makes a prediction using the γ -greedy strategy: At each iteration t , with probability $1 - \gamma$, it predicts $\hat{y}_t = \arg \max_{i \in [k]} (\mathbf{W}_t \mathbf{x}_t)_i$; with the remaining probability γ , it selects a random action in $[k]$. As discussed in Kakade et al. (2008), randomization is essential for designing bandit multiclass learning algorithms. If we deterministically output a label and make a mistake, then it is hard to make an update since we do not know the identity of y_t . However,

Algorithm 1 Second Order Banditron Algorithm (SOBA)

Input: Regularization parameter $a > 0$, exploration parameter $\gamma \in [0, 1]$.

- 1: Initialization: $\mathbf{W}_1 = \mathbf{0}$, $\mathbf{A}_0 = a\mathbf{I}$, $\boldsymbol{\theta}_0 = \mathbf{0}$
- 2: **for** $t = 1, 2, \dots, T$ **do**
- 3: Receive instance $\mathbf{x}_t \in \mathbb{R}^d$
- 4: $\hat{y}_t = \arg \max_{i \in [k]} (\mathbf{W}_t \mathbf{x}_t)_i$
- 5: Define $\mathbf{p}_t = (1 - \gamma)\mathbf{e}_{\hat{y}_t} + \frac{\gamma}{k}\mathbf{1}_k$
- 6: Randomly sample \tilde{y}_t according to \mathbf{p}_t
- 7: Receive bandit feedback $\mathbb{1}[\tilde{y}_t \neq y_t]$
- 8: Initialize update indicator $n_t = 0$
- 9: **if** $\tilde{y}_t = y_t$ **then**
- 10: $\tilde{y}_t = \arg \max_{i \in [k] \setminus \{y_t\}} (\mathbf{W}_t \mathbf{x}_t)_i$
- 11: $\mathbf{g}_t = \frac{1}{p_{t, \tilde{y}_t}} (\mathbf{e}_{\tilde{y}_t} - \mathbf{e}_{y_t}) \otimes \mathbf{x}_t$
- 12: $\mathbf{z}_t = \sqrt{p_{t, \tilde{y}_t}} \mathbf{g}_t$
- 13: $m_t = \frac{\langle \mathbf{W}_t, \mathbf{z}_t \rangle^2 + 2\langle \mathbf{W}_t, \mathbf{g}_t \rangle}{1 + \mathbf{z}_t^T \mathbf{A}_{t-1}^{-1} \mathbf{z}_t}$
- 14: **if** $m_t + \sum_{s=1}^{t-1} n_s m_s \geq 0$ **then**
- 15: Turn on update indicator $n_t = 1$
- 16: **end if**
- 17: **end if**
- 18: Update $\mathbf{A}_t = \mathbf{A}_{t-1} + n_t \mathbf{z}_t \mathbf{z}_t^T$
- 19: Update $\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} - n_t \mathbf{g}_t$
- 20: Set $\mathbf{W}_{t+1} = \text{mat}(\mathbf{A}_t^{-1} \boldsymbol{\theta}_t)$
- 21: **end for**

Remark: matrix \mathbf{A}_t is of dimension $kd \times kd$, and vector $\boldsymbol{\theta}_t$ is of dimension kd ; in line 20, the matrix multiplication results in a kd dimensional vector, which is reshaped to matrix \mathbf{W}_{t+1} of dimension $k \times d$.

if randomization is used, we can estimate y_t and perform online stochastic mirror descent type updates (Bubeck & Cesa-Bianchi, 2012).

SOBA keeps track of two model parameters: cumulative Perceptron-style updates $\boldsymbol{\theta}_t = -\sum_{s=1}^t n_s \mathbf{g}_s \in \mathbb{R}^{kd}$ and corrected covariance matrix $\mathbf{A}_t = a\mathbf{I} + \sum_{s=1}^t n_s \mathbf{z}_s \mathbf{z}_s^T \in \mathbb{R}^{kd \times kd}$. The classifier \mathbf{W}_t is computed by matricizing over the matrix-vector product $\mathbf{A}_{t-1}^{-1} \boldsymbol{\theta}_{t-1} \in \mathbb{R}^{kd}$. The weight vector $\boldsymbol{\theta}_t$ is standard in designing online mirror descent type algorithms (Shalev-Shwartz, 2011; Bubeck & Cesa-Bianchi, 2012). The matrix \mathbf{A}_t is standard in designing online learning algorithms with adaptive regularization (Cesa-Bianchi et al., 2005; Crammer et al., 2009; McMahan & Streeter, 2010; Duchi et al., 2011; Orabona et al., 2015). The algorithm updates its model ($n_t = 1$) only when the following conditions hold simultaneously: (1) the predicted label is correct ($\tilde{y}_t = y_t$), and (2) the ‘‘cumulative regularized negative margin’’ ($\sum_{s=1}^{t-1} n_s m_s + m_t$) is positive if this update were performed. Note that when the predicted label is correct we know the identity of the true label.

As we shall see, the set of iterations where $n_t = 1$ includes all iterations where $\tilde{y}_t = y_t \neq \hat{y}_t$. This fact is cru-

cial to the mistake bound analysis. Furthermore, there are some iterations where $\tilde{y}_t = y_t = \hat{y}_t$ but we still make an update. This idea is related to “online passive-aggressive algorithms” (Crammer et al., 2006; 2009) in the full information setting, where the algorithm makes an update even when it predicts correctly but the margin is too small.

Let’s now describe our algorithm more in details. Throughout, suppose all the examples are ℓ_2 -bounded: $\|\mathbf{x}_t\|_2 \leq X$.

As outlined above, we associate a time-varying regularizer $R_t(\mathbf{W}) = \frac{1}{2} \|\mathbf{W}\|_{\mathbf{A}_t}^2$, where $\mathbf{A}_t = a\mathbf{I} + \sum_{s=1}^t n_s \mathbf{z}_s \mathbf{z}_s^T$ is a $kd \times kd$ matrix and

$$\mathbf{z}_t = \sqrt{p_{t,y_t}} \mathbf{g}_t = \frac{1}{\sqrt{p_{t,y_t}}} (\mathbf{e}_{\tilde{y}_t} - \mathbf{e}_{y_t}) \otimes \mathbf{x}_t.$$

Note that this time-varying regularizer is constructed by scaled versions of the updates \mathbf{g}_t . This is critical, because in expectation this becomes the correct regularizer. Indeed, it is easy to verify that, for any $\mathbf{U} \in \mathbb{R}^{k \times d}$,

$$\begin{aligned} \mathbb{E}_{t-1}[\mathbb{1}[y_t = \tilde{y}_t] \mathbf{g}_t] &= (\mathbf{e}_{\tilde{y}_t} - \mathbf{e}_{y_t}) \otimes \mathbf{x}_t, \\ \mathbb{E}_{t-1}[\mathbb{1}[y_t = \tilde{y}_t] \langle \mathbf{U}, \mathbf{z}_t \rangle^2] &= \langle \mathbf{U}, (\mathbf{e}_{\tilde{y}_t} - \mathbf{e}_{y_t}) \otimes \mathbf{x}_t \rangle^2. \end{aligned}$$

In words, this means that in expectation the regularizer contains the outer products of the updates, that in turn promote the correct class and demotes the wrong one. We stress that it is impossible to get the same result with the estimator proposed in Kakade et al. (2008). Also, the analysis is substantially different from the Online Newton Step approach (Hazan et al., 2007) used in Hazan & Kale (2011).

In reality, we do not make an update in all iterations in which $\tilde{y}_t = y_t$, since the algorithm need to maintain the invariant that $\sum_{s=1}^t m_s n_s \geq 0$, which is crucial to the proof of Lemma 2. Instead, we prove a technical lemma that gives an explicit form on the expected update $n_t \mathbf{g}_t$ and expected regularization $n_t \mathbf{z}_t \mathbf{z}_t^T$. Define

$$\begin{aligned} q_t &:= \mathbb{1} \left[\sum_{s=1}^{t-1} n_s m_s + m_t \geq 0 \right], \\ h_t &:= \mathbb{1}[\hat{y}_t \neq y_t] + q_t \mathbb{1}[\hat{y}_t = y_t]. \end{aligned}$$

Lemma 1. For any $\mathbf{U} \in \mathbb{R}^{kd}$,

$$\begin{aligned} \mathbb{E}_{t-1}[n_t \langle \mathbf{U}, \mathbf{g}_t \rangle] &= h_t \langle \mathbf{U}, (\mathbf{e}_{y_t} - \mathbf{e}_{\tilde{y}_t}) \otimes \mathbf{x}_t \rangle, \\ \mathbb{E}_{t-1}[n_t \langle \mathbf{U}, \mathbf{z}_t \rangle^2] &= h_t \langle \mathbf{U}, (\mathbf{e}_{y_t} - \mathbf{e}_{\tilde{y}_t}) \otimes \mathbf{x}_t \rangle^2. \end{aligned}$$

The proof of Lemma 1 is deferred to the end of Subsection 4.1.

Our last contribution is to show how our second order algorithm satisfies a mistake bound for an entire family of

loss functions. Finally, we relate the performance of the algorithm that predicts \hat{y}_t to the γ -greedy algorithm.

Putting all together, we have our expected mistake bound for SOBA.²

Theorem 2. *SOBA has the following expected upper bound on the number of mistakes, M_T , for any $\mathbf{U} \in \mathbb{R}^{k \times d}$ and any $0 < \eta \leq \min(1, \frac{2}{2 \max_i \|\mathbf{u}_i\|_{X+1}})$,*

$$\begin{aligned} \mathbb{E}[M_T] &\leq L_\eta(\mathbf{U}) + \frac{a\eta}{2-\eta} \|\mathbf{U}\|_F^2 \\ &\quad + \frac{k}{\gamma\eta(2-\eta)} \sum_{t=1}^T \mathbb{E}[\mathbf{z}_t^T \mathbf{A}_t^{-1} \mathbf{z}_t] + \gamma T, \end{aligned}$$

where $L_\eta(\mathbf{U}) := \sum_{t=1}^T \ell_\eta(\mathbf{U}, (\mathbf{x}_t, y_t))$ is the cumulative η -loss of the linear predictor \mathbf{U} , and $\{\mathbf{u}_i\}_{i=1}^k$ are rows of \mathbf{U} .

In particular, setting $\gamma = O(\sqrt{\frac{k^2 d \ln T}{T}})$ and $a = X^2$, we have

$$\mathbb{E}[M_T] \leq L_\eta(\mathbf{U}) + O\left(X^2 \|\mathbf{U}\|_F^2 + \frac{k}{\eta} \sqrt{dT \ln T}\right).$$

Note that, differently from previous analyses (Kakade et al., 2008; Crammer & Gentile, 2013; Hazan & Kale, 2011), we do not need to assume a bound on the norm of the competitor, as in the full information Perceptron and Second Order Perceptron algorithms. In Appendix A, we also present an adaptive variant of SOBA that sets exploration rate γ_t dynamically, which achieves a regret bound within a constant factor of that using optimal tuning of γ .

We prove Theorem 2 in the next Subsection, while in Subsection 4.2 we prove a mistake bound with respect to the hinge loss, that is not fully covered by Theorem 2.

4.1. Proof of Theorem 2

Throughout the proofs, \mathbf{U} , \mathbf{W}_t , \mathbf{g}_t , and \mathbf{z}_t ’s should be thought of as $kd \times 1$ vectors. We first show the following lemma. Note that this is a statement over any sequence and no expectation is taken.

Lemma 2. For any $\mathbf{U} \in \mathbb{R}^{kd}$, with the notation of Algorithm 1, we have:

$$\begin{aligned} &\sum_{t=1}^T n_t \left(2 \langle \mathbf{U}, -\mathbf{g}_t \rangle - \langle \mathbf{U}, \mathbf{z}_t \rangle^2 \right) \\ &\leq a \|\mathbf{U}\|_F^2 + \sum_{t=1}^T n_t \mathbf{g}_t^T \mathbf{A}_t^{-1} \mathbf{g}_t. \end{aligned}$$

²Throughout the paper, expectations are taken with respect to the randomization of the algorithm.

Proof. First, from line 14 of Algorithm 1, it can be seen (by induction) that SOBA maintains the invariant that

$$\sum_{s=1}^t n_s m_s \geq 0. \quad (4)$$

We next reduce the proof to the regret analysis of online least squares problem. For iterations where $n_t = 1$, define $\alpha_t = \frac{1}{\sqrt{p_{t,y_t}}}$ so that $\mathbf{g}_t = \alpha_t \mathbf{z}_t$. From the algorithm, $\mathbf{A}_t = aI + \sum_{s=1}^t n_s \mathbf{z}_s \mathbf{z}_s^T$, and \mathbf{W}_t is the ridge regression solution based on data collected in time 1 to $t-1$, i.e. $\mathbf{W}_t = \mathbf{A}_{t-1}^{-1}(-\sum_{s=1}^{t-1} n_s \mathbf{g}_s) = \mathbf{A}_{t-1}^{-1}(-\sum_{s=1}^{t-1} n_s \alpha_s \mathbf{z}_s)$.

By per-step analysis in online least squares, (see, e.g., Orabona et al., 2012)(See Lemma 6 for a proof), we have that if an update is made at iteration t , i.e. $n_t = 1$, then

$$\begin{aligned} & \frac{1}{2}(\langle \mathbf{W}_t, \mathbf{z}_t \rangle + \alpha_t)^2 (1 - \mathbf{z}_t^T \mathbf{A}_t^{-1} \mathbf{z}_t) - \frac{1}{2}(\langle \mathbf{U}, \mathbf{z}_t \rangle + \alpha_t)^2 \\ & \leq \frac{1}{2} \|\mathbf{U} - \mathbf{W}_t\|_{\mathbf{A}_{t-1}}^2 - \frac{1}{2} \|\mathbf{U} - \mathbf{W}_{t+1}\|_{\mathbf{A}_t}^2. \end{aligned}$$

Otherwise $n_t = 0$, in which case we have $\mathbf{W}_{t+1} = \mathbf{W}_t$ and $\mathbf{A}_{t+1} = \mathbf{A}_t$.

Denoting by $k_t = 1 - \mathbf{z}_t^T \mathbf{A}_t^{-1} \mathbf{z}_t$, by Sherman-Morrison formula, $k_t = \frac{1}{1 + \mathbf{z}_t^T \mathbf{A}_{t-1}^{-1} \mathbf{z}_t}$. Summing over all rounds $t \in [T]$ such that $n_t = 1$,

$$\begin{aligned} & \frac{1}{2} \sum_{t=1}^T n_t [(\langle \mathbf{W}_t, \mathbf{z}_t \rangle + \alpha_t)^2 k_t - (\langle \mathbf{U}, \mathbf{z}_t \rangle + \alpha_t)^2] \\ & \leq \frac{1}{2} \|\mathbf{U}\|_{\mathbf{A}_0}^2 - \frac{1}{2} \|\mathbf{U} - \mathbf{W}_{T+1}\|_{\mathbf{A}_T}^2 \leq \frac{a}{2} \|\mathbf{U}\|_F^2. \end{aligned}$$

We also have by definition of m_t ,

$$\begin{aligned} & (\langle \mathbf{W}_t, \mathbf{z}_t \rangle + \alpha_t)^2 k_t - (\langle \mathbf{U}, \mathbf{z}_t \rangle + \alpha_t)^2 \\ & = m_t - 2 \langle \mathbf{U}, \mathbf{g}_t \rangle - \langle \mathbf{U}, \mathbf{z}_t \rangle^2 - \alpha_t^2 \mathbf{z}_t^T \mathbf{A}_t^{-1} \mathbf{z}_t. \end{aligned}$$

Putting all together and using the fact that $\sum_{t=1}^T n_t m_t \geq 0$, we have the stated bound. \square

We can now prove the following mistake bound for the prediction \hat{y}_t , defined as $\hat{M}_T := \sum_{t=1}^T \mathbf{1}[\hat{y}_t \neq y_t]$.

Theorem 3. For any $\mathbf{U} \in \mathbb{R}^{k \times d}$, and any $0 < \eta \leq \min(1, \frac{2}{2 \max_i \|\mathbf{u}_i\|_{X+1}})$, the expected number of mistakes committed by \hat{y}_t can be bounded as

$$\begin{aligned} \mathbb{E} [\hat{M}_T] & \leq L_\eta(\mathbf{U}) + \frac{a\eta \|\mathbf{U}\|_F^2}{2-\eta} + \frac{k \sum_{t=1}^T \mathbb{E}[n_t \mathbf{z}_t^T \mathbf{A}_t^{-1} \mathbf{z}_t]}{\gamma \eta (2-\eta)} \\ & \leq L_\eta(\mathbf{U}) + \frac{a\eta \|\mathbf{U}\|_F^2}{2-\eta} + \frac{dk^2 \ln \left(1 + \frac{2TX^2}{adk}\right)}{\gamma \eta (2-\eta)}, \end{aligned}$$

where $L_\eta(\mathbf{U}) := \sum_{t=1}^T \ell_\eta(\mathbf{U}, (\mathbf{x}_t, y_t))$ is the η -loss of the linear predictor \mathbf{U} .

Proof. Using Lemma 2 with $\eta \mathbf{U}$, we get that

$$\begin{aligned} & \sum_{t=1}^T n_t \left(2\eta \langle \mathbf{U}, -\mathbf{g}_t \rangle - \eta^2 \langle \mathbf{U}, \mathbf{z}_t \rangle^2 \right) \\ & \leq a\eta^2 \|\mathbf{U}\|_F^2 + \sum_{t=1}^T n_t \mathbf{g}_t^T \mathbf{A}_t^{-1} \mathbf{g}_t. \end{aligned}$$

Taking expectations, using Lemma 1 and the fact that $\frac{1}{p_{t,y_t}} \leq \frac{k}{\gamma}$ and that \mathbf{A}_t is positive definite, we have

$$\begin{aligned} 0 & \leq -\mathbb{E} \left[\sum_{t=1}^T h_t \cdot 2\eta \langle \mathbf{U}, (\mathbf{e}_{y_t} - \mathbf{e}_{\hat{y}_t}) \otimes \mathbf{x}_t \rangle \right] \\ & + \mathbb{E} \left[\sum_{t=1}^T h_t \cdot \eta^2 (\langle \mathbf{U}, (\mathbf{e}_{y_t} - \mathbf{e}_{\hat{y}_t}) \otimes \mathbf{x}_t \rangle)^2 \right] \quad (5) \\ & + a\eta^2 \|\mathbf{U}\|_F^2 + \frac{k}{\gamma} \sum_{t=1}^T \mathbb{E}[n_t \mathbf{z}_t^T \mathbf{A}_t^{-1} \mathbf{z}_t]. \end{aligned}$$

Add the terms $\eta(2-\eta) \mathbb{E} \left[\sum_{t=1}^T h_t \right]$ to both sides and divide both sides by $\eta(2-\eta)$, to have

$$\begin{aligned} \mathbb{E} \left[\sum_{t=1}^T h_t \right] & \leq \mathbb{E} \left[\sum_{t=1}^T h_t f(\langle \mathbf{U}, (\mathbf{e}_{y_t} - \mathbf{e}_{\hat{y}_t}) \otimes \mathbf{x}_t \rangle) \right] \\ & + \frac{a\eta}{2-\eta} \|\mathbf{U}\|_F^2 + \frac{k}{\gamma \eta (2-\eta)} \sum_{t=1}^T \mathbb{E}[n_t \mathbf{z}_t^T \mathbf{A}_t^{-1} \mathbf{z}_t], \end{aligned}$$

where $f(z) := 1 - \frac{2}{2-\eta}z + \frac{\eta}{2-\eta}z^2$. Taking a close look at the function f , we observe that the two roots of the quadratic function are 1 and $\frac{2-\eta}{\eta}$, respectively. Setting $\eta \leq 1$, the function is negative in $(1, \frac{2-\eta}{\eta}]$ and positive in $(-\infty, 1]$. Additionally, if $0 < \eta \leq \frac{2}{2 \max_i \|\mathbf{u}_i\|_{X+1}}$, then for all $i, j \in [k]$, $\langle \mathbf{U}, (\mathbf{e}_i - \mathbf{e}_j) \otimes \mathbf{x}_t \rangle \leq \frac{2-\eta}{\eta}$. Therefore, we have that

$$\begin{aligned} & f(\langle \mathbf{U}, (\mathbf{e}_{y_t} - \mathbf{e}_{\hat{y}_t}) \otimes \mathbf{x}_t \rangle) \\ & = f(\langle \mathbf{U} \mathbf{x}_t, \mathbf{y}_t - \mathbf{e}_{\hat{y}_t} \rangle) \\ & \leq \ell_\eta(\langle \mathbf{U} \mathbf{x}_t, \mathbf{y}_t - \mathbf{e}_{\hat{y}_t} \rangle) \\ & \leq \ell_\eta \left(\langle \mathbf{U} \mathbf{x}_t, \mathbf{y}_t - \max_{r \neq y_t} \langle \mathbf{U} \mathbf{x}_t, \mathbf{e}_r \rangle \rangle \right) = \ell_\eta(\mathbf{U}, (\mathbf{x}_t, y_t)). \end{aligned}$$

where the first equality is from algebra, the first inequality is from that $f(\cdot) \leq \ell_\eta(\cdot)$ in $(-\infty, \frac{2-\eta}{\eta}]$, the second inequality is from that $\ell_\eta(\cdot)$ is monotonically decreasing.

Putting together the two constraints on η , and noting that $\hat{M}_T \leq \sum_{t=1}^T h_t$, we have the first bound.

The second statement follows from Lemma 3 below. \square

Lemma 3. *If $d \geq 1$, $k \geq 2$, $T \geq 2$, then*

$$\sum_{t=1}^T \mathbb{E}[n_t z_t^T \mathbf{A}_t^{-1} z_t] \leq dk \ln \left(1 + \frac{2X^2 T}{a dk} \right).$$

Specifically, if $a = X^2$, the right hand side is $\leq dk \ln T$.

Proof. Observe that

$$\begin{aligned} \sum_{t=1}^T n_t z_t^T \mathbf{A}_t^{-1} z_t &\leq \ln \frac{|\mathbf{A}_T|}{|\mathbf{A}_0|} \\ &\leq dk \ln \left(1 + \frac{2X^2 \sum_{t=1}^T \frac{\mathbb{1}[\tilde{y}_t = y_t]}{p_{t,y_t}}}{a dk} \right), \end{aligned}$$

where the first inequality is a well-known fact from linear algebra (e.g. Hazan et al., 2007, Lemma 11). Given that the \mathbf{A}_T is $kd \times kd$, the second inequality comes from the fact that $|\mathbf{A}_T|$ is maximized when all its eigenvalues are equal to $\frac{\text{tr}(\mathbf{A}_T)}{dk} = a + \frac{\sum_{t=1}^T n_t \|z_t\|^2}{dk} \leq a + \frac{2X^2 \sum_{t=1}^T \frac{\mathbb{1}[\tilde{y}_t = y_t]}{p_{t,y_t}}}{dk}$. Finally, using Jensen's inequality, we have that,

$$\sum_{t=1}^T \mathbb{E}[n_t z_t^T \mathbf{A}_t^{-1} z_t] \leq dk \ln \left(1 + \frac{2X^2 T}{a dk} \right).$$

If $a = X^2$, then the right hand side is $dk \ln(1 + \frac{2T}{dk})$, which is at most $dk \ln T$ under the conditions on d, k, T . \square

Proof of Theorem 2. Observe that by triangle inequality, $\mathbb{1}[\tilde{y}_t \neq y_t] \leq \mathbb{1}[\tilde{y}_t \neq \hat{y}_t] + \mathbb{1}[y_t \neq \hat{y}_t]$. Summing over t , taking expectation on both sides, we conclude that

$$\mathbb{E}[M_T] \leq \mathbb{E}[\hat{M}_T] + \gamma T. \quad (6)$$

The first statement follows from combining the above inequality with Theorem 3.

For the second statement, first note that from Theorem 3, and Equation (6), we have

$$\begin{aligned} \mathbb{E}[M_T] &\leq L_\eta(\mathbf{U}) + \frac{a\eta \|\mathbf{U}\|_F^2}{2-\eta} + \frac{dk^2 \ln \left(1 + \frac{2TX^2}{a dk} \right)}{\gamma\eta(2-\eta)} + \gamma T \\ &\leq L_\eta(\mathbf{U}) + X^2 \|\mathbf{U}\|_F^2 + \frac{2dk^2 \ln T}{\gamma\eta} + \gamma T, \end{aligned}$$

where the second inequality is from that $\eta \leq 1$, and Lemma 3 with $a = X^2$. The statement is concluded by the setting of $\gamma = O(\sqrt{\frac{k^2 d \ln T}{T}})$. \square

Proof of Lemma 1. We show the lemma in two steps. Let $G_t := q_t \cdot \mathbb{1}[y_t = \tilde{y}_t = \hat{y}_t]$, and $H_t := \mathbb{1}[y_t = \tilde{y}_t \neq \hat{y}_t]$.

First, we show that $n_t = G_t + H_t$. Recall that SOBA maintains the invariant (4), hence $\sum_{s=1}^{t-1} n_s m_s \geq 0$. From line 14 of SOBA, we see that $n_t = 1$ only if $\tilde{y}_t = y_t$. Now consider two cases:

- $y_t = \tilde{y}_t \neq \hat{y}_t$. In this case, $\bar{y}_t = \hat{y}_t$, therefore $\langle \mathbf{W}_t, \mathbf{g}_t \rangle \geq 0$, making $m_t \geq 0$. This implies that $\sum_{s=1}^{t-1} n_s m_s + m_t \geq 0$, guaranteeing $n_t = 1$.
- $y_t = \tilde{y}_t = \hat{y}_t$. In this case, n_t is set to 1 if and only if $q_t = 1$, i.e. $\sum_{s=1}^{t-1} n_s m_s + m_t \geq 0$.

This gives that $n_t = G_t + H_t$.

Second, we have the following two equalities:

$$\begin{aligned} &\mathbb{E}_{t-1} [H_t \langle \mathbf{U}, \mathbf{g}_t \rangle] \\ &= \mathbb{E}_{t-1} \left[\frac{\mathbb{1}[\tilde{y}_t = y_t]}{p_{t,y_t}} \mathbb{1}[\hat{y}_t \neq y_t] \langle \mathbf{U}, (\mathbf{e}_{y_t} - \mathbf{e}_{\tilde{y}_t}) \otimes \mathbf{x}_t \rangle \right] \\ &= \mathbb{1}[\hat{y}_t \neq y_t] \langle \mathbf{U}, (\mathbf{e}_{y_t} - \mathbf{e}_{\tilde{y}_t}) \otimes \mathbf{x}_t \rangle, \end{aligned}$$

$$\begin{aligned} &\mathbb{E}_{t-1} [G_t \langle \mathbf{U}, \mathbf{g}_t \rangle] \\ &= \mathbb{E}_{t-1} \left[\frac{\mathbb{1}[\tilde{y}_t = y_t]}{p_{t,y_t}} \mathbb{1}[\hat{y}_t = y_t] q_t \langle \mathbf{U}, (\mathbf{e}_{y_t} - \mathbf{e}_{\tilde{y}_t}) \otimes \mathbf{x}_t \rangle \right] \\ &= \mathbb{1}[\hat{y}_t \neq y_t] q_t \langle \mathbf{U}, (\mathbf{e}_{y_t} - \mathbf{e}_{\tilde{y}_t}) \otimes \mathbf{x}_t \rangle. \end{aligned}$$

The first statement follows from adding up the two equalities above.

The proof for the second statement is identical, except replacing $\langle \mathbf{U}, (\mathbf{e}_{y_t} - \mathbf{e}_{\tilde{y}_t}) \otimes \mathbf{x}_t \rangle$ with $\langle \mathbf{U}, (\mathbf{e}_{y_t} - \mathbf{e}_{\tilde{y}_t}) \otimes \mathbf{x}_t \rangle^2$. \square

4.2. Fall-Back Analysis

The loss function ℓ_η is an interpolation between the hinge and the squared hinge losses. Yet, the bound becomes vacuous for $\eta = 0$. Hence, in this section we show that SOBA also guarantees a $\tilde{O}((L_0(\mathbf{U})T)^{1/3} + \sqrt{T})$ mistake bound w.r.t. $L_0(\mathbf{U})$, the multiclass hinge loss of the competitor, assuming $L_0(\mathbf{U})$ is known. Thus the algorithm achieves a mistake guarantee no worse than the sharpest bound implicit in Kakade et al. (2008).

Theorem 4. *Set $a = X^2$ and denote by M_T the number of mistakes done by SOBA. Then SOBA has the following guarantees:³*

1. *If $L_0(\mathbf{U}) \geq (\|\mathbf{U}\|_F^2 + 1)\sqrt{dk^2 X^2 T \ln T}$, then with parameter setting $\gamma = \min(1, (\frac{dk^2 X^2 L_0(\mathbf{U}) \ln T}{T^2})^{1/3})$, one has the following expected mistake bound:*

$$\begin{aligned} \mathbb{E}[M_T] &\leq L_0(\mathbf{U}) \\ &\quad + O\left(\|\mathbf{U}\|_F (dk^2 X^2 L_0(\mathbf{U}) T \ln T)^{1/3}\right). \end{aligned}$$

³Assuming the knowledge of $\|\mathbf{U}\|_F$ it would be possible to reduce the dependency on $\|\mathbf{U}\|_F$ in both bounds. However such assumption is extremely unrealistic and we prefer not to pursue it.

2. If $L_0(\mathbf{U}) < (\|\mathbf{U}\|_F^2 + 1)\sqrt{dk^2 X^2 T \ln T}$, then with parameter setting $\gamma = \min(1, (\frac{dk^2 X^2 \ln T}{T})^{1/2})$, one has the following expected mistake bound:

$$\mathbb{E}[M_T] \leq L_0(\mathbf{U}) + O\left(k(\|\mathbf{U}\|_F^2 + 1)X\sqrt{dT \ln T}\right).$$

where $L_0(\mathbf{U}) := \sum_{t=1}^T \ell_0(\mathbf{U}, (\mathbf{x}_t, y_t))$ is the hinge loss of the linear classifier \mathbf{U} .

Proof. Recall that \hat{M}_T the mistakes made by \hat{y}_t , that is $\sum_{t=1}^T \mathbb{1}[\hat{y}_t \neq y_t]$. Adding to both sides of (5) the term $\eta \mathbb{E}[\sum_{t=1}^T h_t]$ and dividing both sides by η , and plugging $a = X^2$, we get that for all $\eta > 0$,

$$\begin{aligned} \mathbb{E}\left[\sum_{t=1}^T h_t\right] &\leq \mathbb{E}\left[\sum_{t=1}^T h_t \cdot (1 - \langle \mathbf{U}, (\mathbf{e}_{y_t} - \mathbf{e}_{\hat{y}_t}) \otimes \mathbf{x}_t \rangle)\right. \\ &\quad \left. + \sum_{t=1}^T h_t \cdot \frac{\eta}{2} \langle \mathbf{U}, (\mathbf{e}_{\hat{y}_t} - \mathbf{e}_{y_t}) \otimes \mathbf{x}_t \rangle^2\right] \\ &\quad + \frac{\eta X^2}{2} \|\mathbf{U}\|_F^2 + \frac{dk^2}{2\gamma\eta} \ln T \\ &\leq \mathbb{E}\left[\sum_{t=1}^T \ell_0(\mathbf{U}, (\mathbf{x}_t, y_t)) + \left(\sum_{t=1}^T h_t + \frac{1}{2}\right) \cdot \eta \|\mathbf{U}\|_F^2 X^2\right] \\ &\quad + \frac{dk^2}{2\gamma\eta} \ln T. \end{aligned}$$

where the first inequality uses Lemma 3, the second inequality is from Cauchy-Schwarz that $\langle \mathbf{U}, (\mathbf{e}_{y_t} - \mathbf{e}_{\hat{y}_t}) \otimes \mathbf{x}_t \rangle \leq \|\mathbf{U}\|_F \cdot \|(\mathbf{e}_{y_t} - \mathbf{e}_{\hat{y}_t}) \otimes \mathbf{x}_t\| \leq \|\mathbf{U}\|_F \sqrt{2X}$ and that $(1 - \langle \mathbf{U}, (\mathbf{e}_{y_t} - \mathbf{e}_{\hat{y}_t}) \otimes \mathbf{x}_t \rangle) \leq \ell(\mathbf{U}, (\mathbf{x}_t, y_t))$.

Taking $\eta = \frac{\frac{dk^2}{2\gamma} \ln T}{\|\mathbf{U}\|_F^2 (\mathbb{E}[\sum_{t=1}^T h_t] + \frac{1}{2}) X^2}$, we have

$$\begin{aligned} \mathbb{E}\left[\sum_{t=1}^T h_t\right] &\leq L_0(\mathbf{U}) \\ &\quad + \sqrt{\|\mathbf{U}\|_F^2 \left(\mathbb{E}\left[\sum_{t=1}^T h_t\right] + \frac{1}{2}\right) \frac{dk^2 X^2}{2\gamma} \ln T} \\ &\leq L_0(\mathbf{U}) + \sqrt{\|\mathbf{U}\|_F^2 \left(\mathbb{E}\left[\sum_{t=1}^T h_t\right] + \frac{1}{2}\right) \frac{dk^2 X^2}{\gamma} \ln T}, \end{aligned}$$

where the last inequality is due to the elementary inequality $\sqrt{c+d} \leq \sqrt{c} + \sqrt{d}$, and the setting of $a = X^2$. Solving the inequality and using the fact that $\mathbb{E}[M_T] \leq \mathbb{E}[\hat{M}_T] + \gamma T \leq \mathbb{E}[\sum_{t=1}^T h_t] + \gamma T$, we have

$$\begin{aligned} \mathbb{E}[M_T] &\leq L_0(\mathbf{U}) + \gamma T \\ &\quad + O\left(\frac{dk^2 \|\mathbf{U}\|_F^2 X^2 \ln T}{\gamma} + \sqrt{L_0(\mathbf{U}) \frac{dk^2 \|\mathbf{U}\|_F^2 X^2 \ln T}{\gamma}}\right). \end{aligned}$$

The theorem follows from Lemma 5 in Appendix B, taking $U = \|\mathbf{U}\|_F^2$, $H = dk^2 X^2 \ln T$, $L = L_0(\mathbf{U})$. \square

5. Empirical Results

We tested SOBA to empirically validate the theoretical findings. We used three different datasets from Kakade et al. (2008): `SynSep`, `SynNonSep`, `Reuters4`. The first two are synthetic, with 10^6 samples in \mathbb{R}^{400} and 9 classes. `SynSep` is constructed to be linearly separable, while `SynNonSep` is the same dataset with 5% random label noise. `Reuters4` is generated from the RCV1 dataset (Lewis et al., 2004), extracting the 665,265 examples that have exactly one label from the set $\{\text{CCAT}, \text{ECAT}, \text{GCAT}, \text{MCAT}\}$. It contains 47,236 features. We also report the performance on `Covtype` from LibSVM repository.⁴ We report averages over 10 different runs.

SOBA, as the Newtron algorithm, has a quadratic complexity in the dimension of the data, while the Banditron and the Perceptron algorithm are linear. Following the long tradition of similar algorithms (Crammer et al., 2009; Duchi et al., 2011; Hazan & Kale, 2011; Crammer & Gentile, 2013), to be able to run the algorithm on large datasets, we have implemented an approximated diagonal version of SOBA, named SOBAdiag. It keeps in memory just the diagonal of the matrix A_t . Following Hazan & Kale (2011), we have tested only algorithms designed to work in the fully adversarial setting. Hence, we tested the Banditron and the PNewtron, the diagonal version of the Newtron algorithm in Hazan & Kale (2011). The multiclass Perceptron algorithm was used as a full-information baseline.

In the experiments, we only changed the exploration rate γ , leaving fixed all the other parameters the algorithms might have. In particular, for the PNewtron we set $\alpha = 10$, $\beta = 0.01$, and $D = 1$, as in Hazan & Kale (2011). In SOBA, a is fixed to 1 in all the experiments. We explore the effect of the exploration rate γ in the first row of Figure 5. We see that the PNewtron algorithm,⁵ thanks to the exploration based on the softmax prediction, can achieve very good performance for a wide range of γ .

It is important to note that SOBAdiag has good performance on all four datasets for a value of γ close to 1%. For bigger values, the performance degrades because the best possible error rate is lower bounded by $\frac{k-1}{k}\gamma$ due to explo-

⁴<https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/>

⁵We were unable to make the PNewtron work on `Reuters4`. For any setting of γ the error rate is never better than 57%. The reason might be that the dataset RCV1 has 47,236 features, while the one reported in Kakade et al. (2008); Hazan & Kale (2011) has 346,810, hence the optimal setting of the 3 other parameters of PNewtron might be different. For this reason we prefer not to report the performance of PNewtron on `Reuters4`.

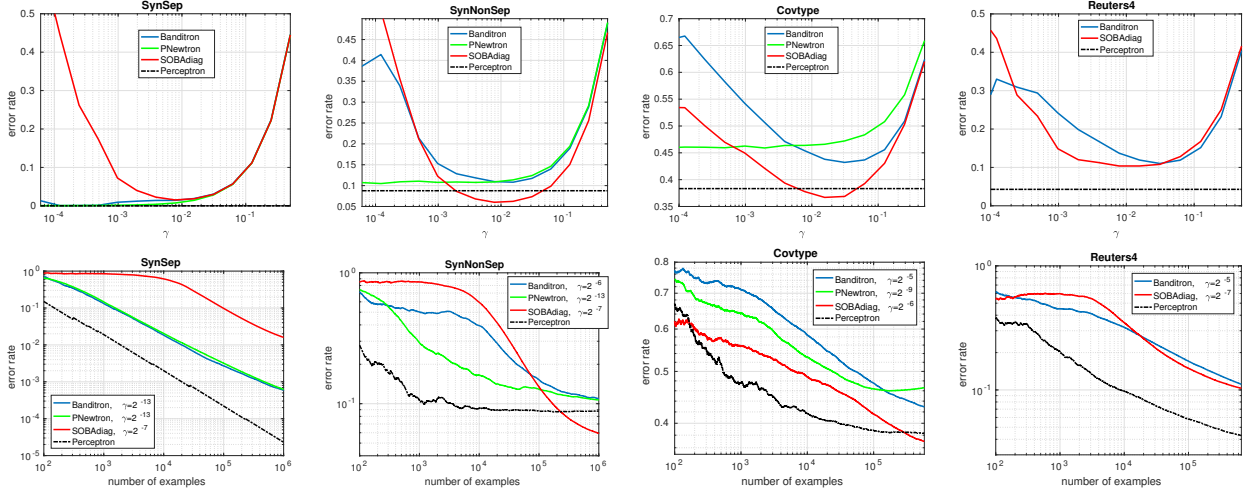


Figure 2. Error rates vs. the value of the exploration rate γ (top row) and vs. the number examples (bottom row). The x-axis is logarithmic in all the plots, while the y-axis is logarithmic in the plots in the second row. Figure best viewed in colors.

ration. For smaller values of exploration, the performance degrades because the algorithm does not update enough. In fact, SOBA updates only when $\hat{y}_t = y_t$, so when γ is too small the algorithms does not explore enough and remains stuck around the initial solution. Also, SOBA requires an initial number of updates to accumulate enough negative terms in the $\sum_t n_t m_t$ in order to start updating also when \hat{y}_t is correct but the margin is too small.

The optimal setting of γ for each algorithm was then used to generate the plots in the second row of Figure 5, where we report the error rate over time. With the respective optimal setting of γ , we note that the performance of PNewtron does not seem better than the one of the Multiclass Perceptron algorithm, and on par or worse to the Banditron’s one. On the other hand, SOBAdiag has the best performance among the bandits algorithms on 3 datasets out of 4.

The first dataset, *SynSep*, is separable and with their optimal setting of γ , all the algorithms converge with a rate of roughly $O(\frac{1}{T})$, as can be seen from the log-log plot, but the bandit algorithms will not converge to zero error rate, but to $\frac{k-1}{k}\gamma$. However, SOBA has an initial phase in which the error rate is high, due to the effect mentioned above.

On the second dataset, *SynNonSep*, SOBAdiag outperforms all the other algorithms (including the full-information Perceptron), achieving an error rate close to the noise level of 5%. This is due to SOBA being a second-order algorithm, while the Perceptron is a first-order algorithm. A similar situation is observed on *Covtype*. On the last dataset, *Reuters4*, SOBAdiag achieves performance better than the Banditron.

6. Discussion and Future Work

In this paper, we study the problem of online multiclass learning with bandit feedback. We propose SOBA, an algorithm that achieves a regret of $\tilde{O}(\frac{1}{\eta}\sqrt{T})$ with respect to η -loss of the competitor. This answers a COLT open problem posed by (Abernethy & Rakhlin, 2009). Its key ideas are to apply a novel adaptive regularizer in a second order online learning algorithm, coupled with updates only when the predictions are correct. SOBA is shown to have competitive performance compared to its precedents in synthetic and real datasets, in some cases even better than the full-information Perceptron algorithm. There are several open questions we wish to explore:

1. Is it possible to design efficient algorithms with mistake bounds that depend on the loss of the competitor, i.e. $\mathbb{E}[M_T] \leq L_\eta(\mathbf{U}) + \tilde{O}(\sqrt{kdL_\eta(\mathbf{U})} + kd)$? This type of bound occurs naturally in the full information multiclass online learning setting, (see e.g. Theorem 1), or in multi-armed bandit setting, e.g. (Neu, 2015).
2. Are there efficient algorithms that have a finite mistake bound in the separable case? (Kakade et al., 2008) provides an algorithm that performs enumeration and plurality vote to achieve a finite mistake bound in the finite dimensional setting, but unfortunately the algorithm is impractical. Notice that it is easy to show that in SOBA \hat{y}_t makes a logarithmic number of mistakes in the separable case, with a constant rate of exploration, yet it is not clear how to decrease the exploration over time in order to get a logarithmic number of mistakes for \hat{y}_t .

Acknowledgments. We thank Claudio Gentile for suggesting the original plan of attack for this problem, and thank the anonymous reviewers for thoughtful comments.

References

- Abernethy, J. and Rakhlin, A. An efficient bandit algorithm for \sqrt{T} -regret in online multiclass prediction? In *COLT*, 2009.
- Agarwal, A., Hsu, D., Kale, S., Langford, J., Li, L., and Schapire, R. E. Taming the monster: a fast and simple algorithm for contextual bandits. *ICML*, 2014.
- Auer, P., Cesa-Bianchi, N., and Gentile, C. Adaptive and self-confident on-line learning algorithms. *J. Comput. Syst. Sci.*, 64(1):48–75, 2002.
- Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. E. The nonstochastic multiarmed bandit problem. *SIAM J. Comput.*, 32(1):48–77, January 2003.
- Bubeck, S. and Cesa-Bianchi, N. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *FnTML*, 5(1):1–122, 2012.
- Cesa-Bianchi, N. and Lugosi, G. *Prediction, learning, and games*. Cambridge University Press, 2006.
- Cesa-Bianchi, N., Conconi, A., and Gentile, C. A second-order Perceptron algorithm. *SIAM Journal on Computing*, 34(3):640–668, 2005.
- Crammer, K. and Gentile, C. Multiclass classification with bandit feedback using adaptive regularization. *Machine learning*, 90(3):347–383, 2013.
- Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., and Singer, Y. Online passive-aggressive algorithms. *JMLR*, 7(Mar):551–585, 2006.
- Crammer, K., Kulesza, A., and Dredze, M. Adaptive regularization of weight vectors. In *NIPS*, pp. 414–422, 2009.
- Duchi, J., Hazan, E., and Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *JMLR*, 12(Jul):2121–2159, 2011.
- Duda, R. O. and Hart, P. E. *Pattern classification and scene analysis*. John Wiley, 1973.
- Dudík, M., Hsu, D. J., Kale, S., Karampatziakis, N., Langford, J., Reyzin, L., and Zhang, T. Efficient optimal learning for contextual bandits. In *UAI 2011*, pp. 169–178, 2011.
- Hazan, E. and Kale, S. Newtron: an efficient bandit algorithm for online multiclass prediction. In *NIPS*, pp. 891–899, 2011.
- Hazan, E., Agarwal, A., and Kale, S. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.
- Kakade, S. M., Shalev-Shwartz, S., and Tewari, A. Efficient bandit algorithms for online multiclass prediction. In *ICML*, pp. 440–447. ACM, 2008.
- Kivinen, J. and Warmuth, M. Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, 132(1):1–63, January 1997.
- Langford, J. and Zhang, T. The epoch-greedy algorithm for multi-armed bandits with side information. In *NIPS 20*, pp. 817–824. 2008.
- Lewis, D. D., Yang, Y., Rose, T. G., and Li, F. RCV1: A new benchmark collection for text categorization research. *JMLR*, 5(Apr):361–397, 2004.
- McMahan, H Brendan and Streeter, Matthew. Adaptive bound optimization for online convex optimization. *COLT*, 2010.
- Mohri, M. and Rostamizadeh, A. Perceptron mistake bounds. *arXiv preprint arXiv:1305.0208*, 2013.
- Neu, G. First-order regret bounds for combinatorial semi-bandits. In *COLT*, pp. 1360–1375, 2015.
- Orabona, F., Cesa-Bianchi, N., and Gentile, C. Beyond logarithmic bounds in online learning. In *AISTATS*, pp. 823–831, 2012.
- Orabona, F., Crammer, K., and Cesa-Bianchi, N. A generalized online mirror descent with applications to classification and regression. *Machine Learning*, 99(3):411–435, 2015.
- Rakhlin, A. and Sridharan, K. BISTRO: An efficient relaxation-based method for contextual bandits. In *ICML*, 2016.
- Rosenblatt, F. The Perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386–407, 1958.
- Shalev-Shwartz, S. Online learning and online convex optimization. *FnTML*, 4(2):107–194, 2011.
- Syrkkanis, Vasilis, Krishnamurthy, Akshay, and Schapire, Robert. Efficient algorithms for adversarial contextual learning. In *ICML*, pp. 2159–2168, 2016a.
- Syrkkanis, Vasilis, Luo, Haipeng, Krishnamurthy, Akshay, and Schapire, Robert E. Improved regret bounds for oracle-based adversarial contextual bandits. In *NIPS*, pp. 3135–3143, 2016b.
- Valizadegan, H., Jin, R., and Wang, S. Learning to trade off between exploration and exploitation in multiclass bandit prediction. In *KDD*, pp. 204–212. ACM, 2011.

Wang, S., Jin, R., and Valizadegan, H. A potential-based framework for online multi-class learning with partial feedback. In *AISTATS*, pp. 900–907, 2010.