Review on Efficient Bandit Algorithms for Online Multiclass Prediction

Xiaochen Zhu [1 2]

Abstract

This report provides a critical review of the paper "Efficient bandit algorithms for online multiclass prediction"(Kakade et al., 2008), which proposes an efficient algorithm, namely *Banditron* for multiclass online learning with linear hypotheses in a bandit setting where the learner can only receive partial feedback about whether its prediction is correct or not, instead of the full information feedback which also contains the correct label. We review the Banditron algorithm and related proofs and discuss the positive points and contributions from an advocate perspective. Then, we identify some limitations with the algorithm, namely: (i) the mistake bound is polynomial to the number of examples; (ii) it is hard to optimally tune the hyperparameter γ. Then, we propose an improvement of Banditron to tackle the latter issue and theoretically show the mistake bound of the improved Banditron algorithm. At last, we address the incompleteness of our work (due to limited time) and further research work required to complement the current result.

1. Overview

In convential multiclass machine learning (ML) settings, a learner aims to learn a model from a dataset $\mathcal{D} = \{(x_t, y_t) : t \in \{1, 2, \dots, T\}, x_t \in \mathbb{R}^d, y_t \in [k] = \{1, 2, \dots, k\}\}$ where d is the dimension of examples and k is the number of label classes. First, the learner initializes its model and make prediction $\hat{y}_t$ on example x_t. Then, it compares its own prediction $\hat{y}_t$ and the ground truth label y_t to update its own model. However, in real world scenarios, one may not always have the privilege to know the actual labels but is only aware of the correctness of its predictions. We refer to this scenario as bandit setting, where the learner only has access to indicator $\mathbf{1}[\hat{y}_t = y_t]$ instead of y_t. Naturally, ML in bandit setting is an online (the learner cannot test a batch of examples at a time) multiclass (knowing $\mathbf{1}[\hat{y}_t = y_t]$ is equivalent to knowing y_t for binary classification) classification problem, which has been extensively studied. The work of our interest, Kakade et al. (2008), studies the learning of linear predicators in the bandit setting and proposes a variant of Multiclass Perceptron (Crammer & Singer, 2003), namely Banditron. This section reviews the algorithm and gives a proof sketch on its mistake bound.

1.1. Multiclass Perceptron

Before introducing Banditron, we first discuss Multiclass Perceptron, the foundation of Banditron in the full information context (i.e. the learner receives ground truth labels y_t instead of bandit feedback). In Multiclass Perceptron, the learner learns a model $W \in \mathbb{R}^{k\times d}$, where each row vector W_j can be interpreted as a linear predictor $\theta_j \in \mathbb{R}^d$ as in the original binary Perceptron algorithm (Rosenblatt, 1958). On example $\mathbf{x}_t, y_t$, the learner predicts $\operatorname{argmax}_j(W\mathbf{x}_t)_j$, the index of the maximum prediction among all θ_j predictions. Similar to the binary Perceptron, Multicalss Perceptron updates W on prediction mistakes, penalizing the row vector of W making the wrong prediction and reinforcing the row vector of W that should have given the greatest output. That is, given that $\hat{y}_t = \operatorname{argmax}_j(W\mathbf{x}_t)_j$, then we have $\forall j \in [k]$,

$$W_j^{(t+1)} = \begin{cases} W_j^{(t)}, & \text{if } \hat{y}_t = y_t \\ \begin{cases} W_j^{(t)} - \mathbf{x}_t, & \text{if } j = \hat{y}_t \\ W_j^{(t)} + \mathbf{x}_t, & \text{if } j = y_t \end{cases}, & \text{if } \hat{y}_t \neq y_t \end{cases} \tag{1}$$

Rearranging Equation (1), the update rule of Multiclass Perceptron can be expressed as $W^{(t+1)} = W^{(t)} + U^{(t)}$, where for all $j \in [k]$,

$$U_r^{(t)} = \mathbf{x}_t(\mathbf{1}[y_t = r] - \mathbf{1}[\hat{y}_t = r]). \tag{2}$$

Despite its simplicity, Multiclass Perceptron enjoys favorable mistake bounds. Fink et al. (2006) prove that if $\|\mathbf{x}_t\| \leq 1$ for all t, the number of mistakes (i.e. updates) Multiclass Perceptron makes to reach any fixed model $W^* \in \mathbb{R}^{(k\times d)}$ is upper bounded by $L + D + \sqrt{LD}$, where

[1]Department of Computer Science, National University of Singapore [2]Department of Mathematics, National University of Singapore. Correspondence to: Xiaochen Zhu, A0194512H <xzhu@u.nus.edu>.

Term paper submitted to Prof Vincent Tan in partial fulfillment of the requirements for *MA4270 Data Modeling and Computation*, November, 2021. Copyright 2021 by the author(s).

Algorithm 1 Banditron

```
Given: $\gamma \in (0,1)$
Initialize $W^{(1)} \leftarrow \mathbf{0} \in \mathbb{R}^{(k\times d)}$
for $t = 1$ to $T$ do
  $\hat{y}_t \leftarrow \text{argmax}_j (W^{(t)}\mathbf{x}_t)_j$
  Choose from {exploitation, exploration} with probability $(1-\gamma, \gamma)$
  if exploitation then
    $\tilde{y_t} = \hat{y}_t$
  else
    Randomly choose $\tilde{y_t}$ from $[k]$ uniformly
  end if
  $W^{(t+1)} \leftarrow W^{(t)}$
  if $\tilde{y_t} \neq y_t$ then
    Update $W^{(t+1)}_{\tilde{y_t}} \leftarrow W^{(t)}_{\tilde{y_t}} + \mathbf{x}_t/\text{P}(\tilde{y_t})$
    Update $W^{(t+1)}_{\hat{y}_t} \leftarrow W^{(t)}_{\hat{y}_t} - \mathbf{x}_t$
  else
    Update $W^{(t+1)}_{\hat{y}_t} \leftarrow W^{(t)}_{\hat{y}_t} - \mathbf{x}_t$
  end if
end for
```

L is the sum of hinge loss of W^* on all examples [1] and $D = 2\sum_{i=1}^{k}\sum_{j=1}^{d}(W^*_{ij})^2$, the *complexity*[2] of W^*. Note that this bound is not assumed with any linear separability of $\mathcal{D}$. On contrast, Crammer & Singer (2003) shows that if $\|\mathbf{x}_t\| \leq R$ for all t and there exists a W^* whose complexity $D \leq 1$ that separate all examples correctly with a margin γ (i.e. for all t, $W^*_{y_t}\mathbf{x}_t \geq \gamma/2$ and $W^*_i\mathbf{x}_t \leq \gamma/2$ for all $i \neq y_t$), the number of updates performed by Multiclass Perceptron is bounded by $2R/\gamma^2$. Since the focus of this review is Banditron, we will not prove these mistake bounds here.

1.2. The Banditron algorithm

Banditron, the main result of Kakade et al. (2008), adopts the idea of Multiclass Perceptron. However, it cannot reinforce the row vector W_{y_t} on mistakes because it does not receive ground truth labels y_t. Banditron addresses this issue by performing exploration rounds (predict based on a probability distribution) as well as exploitation rounds (predict based on its model W), which is a common idea in reinforcement learning. This is natural and intuitive because rather than being a supervised ML problem where the learner has access to a dataset with labels, the setting of ML with bandit feedback is more like reinforcement learning, where the learner has no information about the correct prediction, but only provided with a reward feedback on each of its actions, i.e. $\mathbf{1}[y_t = \hat{y}_t]$. The general idea of Banditron is that, on each example, the learner either explores with probability γ, where it randomly pick one label (i.e. each one with probability γ/k) to test, or explores with probability $1-\gamma$, where it predits $\hat{y} = \text{argmax}_j(W\mathbf{x}_t)_j$. Hence, on example $(\mathbf{x}_t, y_t)$, Banditron predicts $\hat{y}_t$ with probability $\gamma/k + 1 - \gamma$ and any other $i \neq \hat{y}_t$ with probability γ/k. Denote the probability of choosing $\tilde{y_t} = i$ as $\text{P}(i)$, the complete algorithm follows as Algorithm 1. If one wants a closed-form update policy $W^{(t+1)} = W^{(t)} + \tilde{U}^{(t)}$ using the style of (2), we have for all $r \in [k]$

$$\tilde{U}^{(t)}_r = \mathbf{x}_t\left(\frac{\mathbf{1}[y_t = \tilde{y_t}]\mathbf{1}[r = \tilde{y_t}]}{\text{P}(r)} - \mathbf{1}[r = \hat{y}_t]\right). \quad (3)$$

On correct predictions, if Banditron predicts by exploration, it reinforces the $\tilde{y_t}$-th row vector and penalize the $\hat{y}_t$-th row vector. If Banditron predicts correctly by exploitation, it adds $\mathbf{x}_t/\text{P}(\hat{y}_t) - \mathbf{x}_t$ to the $\hat{y}_t$-th row vector which is an reinforcement since $1/\text{P}(\hat{y}_t) > 1$. On wrong predictions, Banditron subtracts $\mathbf{x}_t$ from the $\hat{y}_t$-th row vector, to make the algorithm *unbiased* compared with Multiclass Perceptron, which is fomally defined in Lemma 1.

1.3. Mistake bound of Banditron

Kakade et al. (2008) proves a mistake bound similar to the $L + D + \sqrt{LD}$ bound for Multiclass Perceptron (Fink et al., 2006), stated in the following Theorem 1, whose concise sketch of proof will be discussed in this section.

Theorem 1 (Mistake bound of Banditron) *If $\|\mathbf{x}_t\| \leq 1$ for all $t \in [T]$, then for any fixed model W^*, the number of mistakes M made by Banditron satisifes*

$$\mathbb{E}(M) \leq L + \gamma T + 3\max\left\{\frac{kD}{\gamma}, \sqrt{D\gamma T}\right\} + \sqrt{\frac{kDL}{\gamma}}, \quad (4)$$

where L is the sum of hinge loss of W^* on all examples and D is the complexity of W^* as defined earlier.

To bound the expectation of M, the number of mistakes Banditron makes when approaching W^*, naturally one investigates the term $\mathbb{E}[\langle W^*, W^{(t)}\rangle]$ [3], which adopts the same idea used to prove the convergence bound of binary and multiclass Perceptron. The key ideas is to construct both an upper and a lower bound of $\langle W^*, W^{(t)}\rangle$. To do so, one needs the following lemmas, which can be proved by calculating the expectation w.r.t. $\tilde{y_t}$ using Equation (3).

Lemma 1 *Banditron is unbiased compared with Multiclass Perceptron, in the sense for each update, that the expectation of Banditron's update matrix $\tilde{U}^{(t)}$ is the same as Multiclass Perceptron's update matrix $\tilde{U}$, w.r.t. the random variable $\tilde{y_t}$, that is, $\mathbb{E}_t[\tilde{U}^{(t)}] = U^{(t)}$.*

[1] $L = \sum_{t=1}^{T}\max_{r\in[k]\setminus\{y_t\}}\max\{0, 1 - W^*_{y_t}\mathbf{x}_t + W^*_r\mathbf{x}_t\}$.

[2] $D = 2\|W^*\|_F^2$, twice the square of Frobenius norm of W^*.

[3] This is an extension of the conventionly inner product between vectors in the domain of matrices, i.e. $\langle A, B\rangle = \sum_i\sum_j A_{ij}B_{ij}$.

Lemma 2 *The expectation (w.r.t. $\tilde{y}_t$) of the squared Frobenius norm of $\tilde{U}^{(t)}$ is bounded as $\mathbb{E}_t[\|\tilde{U}^{(t)}\|_F^2] \leq 2\|\mathbf{x}_t\|^2 \left(k/\gamma \mathbf{1}[y_t \neq \hat{y}_t] + \gamma \mathbf{1}[y_t = \hat{y}_t]\right)$.*

First to lower bound $\mathbb{E}[\langle W^*, W^{(T+1)}\rangle]$, we have

$$\begin{aligned}
&\mathbb{E}[\langle W^*, W^{(T+1)}\rangle] = \sum_{t=1}^{T} \mathbb{E}[\langle W^*, W^{(t+1)}\rangle] - \mathbb{E}[\langle W^*, W^{(t)}\rangle] \\
&= \sum_{t=1}^{T} \mathbb{E}[\langle W^*, W^{(t+1)} - W^{(t)}\rangle] = \sum_{t=1}^{T} \mathbb{E}[\langle W^*, \tilde{U}^{(t)}\rangle] \\
&= \sum_{t=1}^{T} \mathbb{E}[\langle W^*, U^{(t)}\rangle] \qquad \text{(by Lemma 1)} \\
&\geq \sum_{t=1}^{T} \mathbb{E}[\mathbf{1}[\hat{y}_t \neq y_t]] - \ell_{\text{hinge}}(W^*, (\mathbf{x}_t, y_t)) = \mathbb{E}[\hat{M}] - L,
\end{aligned} \tag{5}$$

where $\hat{M} = \sum_{t=1}^{T} \mathbf{1}[\hat{y}_t \neq y_t]$. The last inequality is true because $\ell_{\text{hinge}}(W^*, (\mathbf{x}_t, y_t)) \geq \mathbf{1}[\hat{y}_t \neq y_t] - \langle W^*, U^{(t)}\rangle$, derived from the definition of multiclass hinge loss.

Then, one wants to upper bound $\mathbb{E}[\langle W^*, W^{(T+1)}\rangle]$, using Cauchy-Schwartz and Jensen's inequalities, we have

$$\begin{aligned}
&\mathbb{E}[\langle W^*, W^{(T+1)}\rangle] \leq \mathbb{E}[\|W^*\|_F \|W^{(T+1)}\|_F] \\
&\leq \sqrt{\frac{D\mathbb{E}[\|W^{(T+1)}\|_F^2]}{2}} \qquad \text{(since } \sqrt{\cdot} \text{ is concave)}.
\end{aligned} \tag{6}$$

Since $\|W^{(t+1)}\|_F^2 = \|W^{(t)}\|_F^2 + \langle W^{(t)}, \tilde{U}^{(t)}\rangle + \|\tilde{U}^{(t)}\|_F^2$ for all $t \in [T]$, we then have

$$\begin{aligned}
&\mathbb{E}[\|W^{(T+1)}\|_F^2] = \sum_{t=1}^{T} (\mathbb{E}[\langle W^{(t)}, \tilde{U}^{(t)}\rangle] + \mathbb{E}[\|\tilde{U}^{(t)}\|_F^2]) \\
&= \sum_{t=1}^{T} (\mathbb{E}[\langle W^{(t)}, U^{(t)}\rangle] + \mathbb{E}[\|\tilde{U}^{(t)}\|_F^2]) \quad \text{(by Lemma 1)} \\
&\leq \sum_{t=1}^{T} \mathbb{E}[\|\tilde{U}^{(t)}\|_F^2] \\
&\leq \sum_{t=1}^{T} \mathbb{E}[\frac{2k}{\gamma}\mathbf{1}[y_t \neq \hat{y}_t] + 2\gamma \mathbf{1}[y_t = \hat{y}_t]] \quad \text{(by Lemma 2)} \\
&= \frac{2k}{\gamma}\mathbb{E}[\hat{M}] + 2\gamma T.
\end{aligned} \tag{7}$$

Substituting (7) into (6), we have

$$\begin{aligned}
\mathbb{E}[\hat{M}] - L &\leq \mathbb{E}[\langle W^*, W^{(T+1)}\rangle] \\
&\leq \sqrt{\frac{Dk\mathbb{E}[\hat{M}]}{\gamma}} + \sqrt{D\gamma T}.
\end{aligned} \tag{8}$$

Therefore, one can give an upper bound of $\mathbb{E}[\hat{M}]$ from (8) and because Banditron explores no more than γT rounds, we have

$$\mathbb{E}[M] \leq \mathbb{E}[\hat{M}] + \gamma T \tag{9}$$

$$\leq L + \gamma T + 3\max\left\{\frac{kD}{\gamma}, \sqrt{D\gamma T}\right\} + \sqrt{\frac{kDL}{\gamma}}. \tag{10}$$

2. Contributions

From the perspective of an advocate, this section discusses some of the contributions and positive points about Kakade et al. (2008).

2.1. As an unbiased variant of Multiclass Perceptron

First of all, like its foundation Multiclass Perceptron, Banditron is an intuitive and simple algorithm that naturally makes sense. While Lemma 1 guarantees that the expected update matrix in Banditron is the same as Multiclass Perceptron, Banditron enjoys some of the advantages of Multiclass Perceptron such as the favorable performance (demonstrated in Kakade et al. (2008) empirically) and mistake bound (as proved in Section 1.3). As one of the earliest paper studying multiclass linear classification with bandit feedback, Kakade et al. (2008) enables further research to study this area more thoroughly. In short, the importance of the Banditron algorithm w.r.t. the bandit setting should be similar to that of the Perceptron (Rosenblatt, 1958) w.r.t. the full information setting.

2.2. Mistake bound w.r.t. arbitrary W^*

The approach Kakade et al. (2008) uses to provide a mistake bound for Banditron is similar to Fink et al. (2006), which is given w.r.t. any fixed model matrix W^* without any further assumptions on either the separability of $\mathcal{D}$ or the loss of W^*. In contrast, the mistake bounds for Perceptron and Multiclass Perceptron derived in Rosenblatt (1958); Crammer & Singer (2003) are made under the assumption of linear separability with a known margin and the existence of a zero-loss solution W^*. However, when running the algorithms, one cannot guarantee neither the existence of a solution nor the margin for linear separability, while the bounds in Kakade et al. (2008) and Fink et al. (2006) enables anyone to know the upper bound of mistakes when approaching any arbitrary goal W^*. Hence, the bound given by Kakade et al. (2008) is more generalized.

While it is not possible to derive the general case mistake boundary from the linear separable case, we can easily derive some mistake bound on the separable case from the general bound given by Kakade et al. (2008). Assume that $\mathcal{D}$ is linearly separable and W^* is a zero-loss solution, one can easily derive that $\mathbb{E}[M]$ is bounded by $O(\sqrt{T})$ by substituting $L = 0$ in Equation (10).

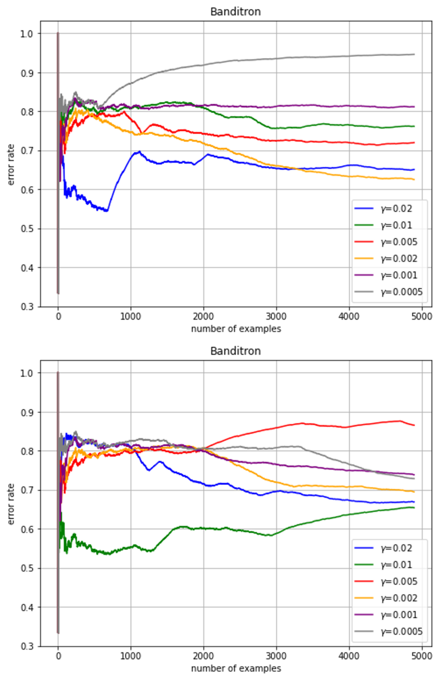

Figure 1. Two runs of Banditron with various γ

Algorithm 2 Banditron with variable γ

```
Given: $\gamma_1, \gamma_2, \ldots, \gamma_T \in (0, 1)$
Initialize $W^{(1)} \leftarrow \mathbf{0} \in \mathbb{R}^{(k \times d)}$
for $t = 1$ to $T$ do
    $\hat{y}_t \leftarrow \text{argmax}_j (W^{(t)} \mathbf{x}_t)_j$
    $\gamma \leftarrow \gamma_t$
    Choose from $\{\text{exploitation}, \text{exploration}\}$ with probability $(1 - \gamma, \gamma)$
    if exploitation then
        $\tilde{y}_t = \hat{y}_t$
    else
        Randomly choose $\tilde{y}_t$ from $[k]$ uniformly
    end if
    $W^{(t+1)} \leftarrow W^{(t)}$
    if $\tilde{y}_t \neq y_t$ then
        Update $W^{(t+1)}_{\tilde{y}_t} \leftarrow W^{(t)}_{\tilde{y}_t} + \mathbf{x}_t / \text{P}(\tilde{y}_t)$
        Update $W^{(t+1)}_{\hat{y}_t} \leftarrow W^{(t)}_{\hat{y}_t} - \mathbf{x}_t$
    else
        Update $W^{(t+1)}_{\hat{y}_t} \leftarrow W^{(t)}_{\hat{y}_t} - \mathbf{x}_t$
    end if
end for
```

3. Limitations

From the perspective of a critic, this section discusses some of the limitations and negative points about the Kakade et al. (2008).

3.1. Dependency of mistake bound on T

Note that the mistake bound given by Kakade et al. (2008) (Theorem 1) is polynomial large w.r.t. the number of examples/updates T. Such dependency on T does not exist on the mistake bounds in Multiclass Perceptron and has made Banditron's mistake bounds undesirable with a long time horizon. Beygelzimer et al. (2019) addresses this issue in the m-linearly separable case by proprosing an algorithm that achieve near-optimal mistake bound of $O(k/\sqrt{m})$.

3.2. Tuning of hyperparameter γ

Another limitation of Banditron is the difficulty in tuning of hyperparameter, namely the exploration-exploitation trade-off parameter γ. First, because of the randomness of the algorithm, the performance (i.e. error rate) of Banditron with different γ values have high variation and can be dramastically different in each run. This has been demonstrated in Figure 3, where the same algorithm with the same set of γ values is executed on the dataset of Wine quality (Cortez et al., 2009), which is a non-linearly separable dataset with $k = 7, d = 11, T = 4898$. We can see that different runs of the same parameter may result in dramastically different performance (i.e. in the first run, the best γ seems to be 0.002 while in the second run, the best γ seems to be much larger at either 0.02 or 0.01). Even one can run the algorithm for multiple times and compare the average performance to choose γ, the high variantion of the performance will make such hyperparameter tuning less effective.

Another issue regarding tuning of γ is that, the optimal γ that can minimize error rate varies a lot on different datasets, the γ chosen from the known dataset may not be applicable for predicting actual user behavior, while the learner cannot directly trains Banditron with user to with several γ to choose the best one (because in bandit setting, the learner cannot choose the examples it sees or sees the same example over and over again, otherwise the learner can just trial and error every label until receiving a positive feedback). Hence, the choice of γ seems to be difficult in real world applications.

4. Extension

In this section, we introduce some extensions of Banditron theoretically and empirically, focusing on the use of variable γ in Banditron. First, there are several issues with the constant γ value in Banditron:

1. As discussed in Section 3.2 constant γ will have big impact on the performance and make it harder to tune;

2. With constant γ, Banditron uses the same exploitation-exploration stategy even after a long time horizon, when the matrix W should be able to reach a very

small loss and Banditron should value exploitation more than exploration. Instead, Banditron still explores with probability γ. This is especially undesirable under separability.

Motivated by these issues, we study the use of variable γ instead of constant γ in Banditron. If Banditron can still work under variable γ, then we can design better and more flexible update policies, such as exploration more in earlier iterations while exploitation more in later ones. Moreover, the sequence of γ is potentially *learnable* to minimize the error rate and therefore there's no need to tune the parameter.

The Banditron with variable γ is given as Algorithm 2, and we can prove the following mistake bound for this algorithm.

Theorem 2 (Banditron with variable γ) *If $\|\mathbf{x}_t\| \le 1$ for all $t \in [T]$, then for any fixed model W^*, the number of mistakes M made by Algorithm 2 satisifes*

$$\mathbb{E}[M] \le L + \gamma_{\text{sum}} + 3\max\{\frac{kD}{\gamma_{\min}}, \sqrt{D\gamma_{\text{sum}}}\} + \sqrt{\frac{kDL}{\gamma_{\min}}} \tag{11}$$

where $\gamma_{\min} = \min\{\gamma_1, \dots, \gamma_T\}, \gamma_{\text{sum}} = \sum_{t=1}^T \gamma_t$ and k, L, D have been defined previously.

We hereby provide a proof sketch of the stated theorem by modifying the original proof in Section 1.3.

First, note that although the γ values in different rounds are different, the update policy in each round is still the same with the original Banditron of fixed γ. Therefore, Lemma 1 and 2 are still true (other than replacing γ with γ_t in Lemma 2) given that the expectation is taken w.r.t. the choice of $\tilde{y}_t$ in a single round with one γ. Therefore Equation (5) should still be true while Equation (7) now becomes

$$\begin{aligned}
&\mathbb{E}[\|W^{(T+1)}\|_F^2] \\
&\le \sum_{t=1}^T \mathbb{E}[\frac{2k}{\gamma_t}\mathbf{1}[y_t \ne \hat{y}_t] + 2\gamma_t \mathbf{1}[y_t = \hat{y}_t]] \quad \text{(by Lemma 2)} \\
&= \sum_{t=1}^T \mathbb{E}\left[\left(\frac{2k}{\gamma_t} - 2\gamma_t\right)\mathbf{1}[y_t \ne \hat{y}_t] + 2\gamma_t\right] \\
&= 2\gamma_{\text{sum}} + \sum_{t=1}^T \mathbb{E}\left[\left(\frac{2k}{\gamma_t} - 2\gamma_t\right)\mathbf{1}[y_t \ne \hat{y}_t]\right] \\
&\le 2\gamma_{\text{sum}} + \sum_{t=1}^T \mathbb{E}\left[\left(\frac{2k}{\gamma_{\min}} - 2\gamma_{\min}\right)\mathbf{1}[y_t \ne \hat{y}_t]\right] \\
&\le 2\gamma_{\text{sum}} + \frac{2k}{\gamma_{\min}}\mathbb{E}[\hat{M}].
\end{aligned} \tag{12}$$

Therefore, similar to Equation (8), we now have

$$\begin{aligned}
\mathbb{E}[\hat{M}] - L &\le \mathbb{E}[\langle W^*, W^{(T+1)}\rangle] \\
&\le \sqrt{\frac{Dk\mathbb{E}[\hat{M}]}{\gamma_{\min}}} + \sqrt{D\gamma_{\text{sum}}}.
\end{aligned} \tag{13}$$

Hence, one can derive an upper bound of $\mathbb{E}[\hat{M}]$ from Equation (13). Since our algorithm in expectation runs γ_{sum} exploration rounds, we have

$$\begin{aligned}
&\mathbb{E}[M] \le \mathbb{E}[\hat{M}] + \gamma_{\text{sum}} \\
&\le L + \gamma_{\text{sum}} + 3\max\{\frac{kD}{\gamma_{\min}}, \sqrt{D\gamma_{\text{sum}}}\} + \sqrt{\frac{kDL}{\gamma_{\min}}}.
\end{aligned} \tag{14}$$

5. Further research

Therorem 2 enables Banditron to run with variable γ values, which can be applied to design better update policies for Banditron. I have experimented several (intuitive) update policies on the datasets of wine quality and Iris (Cortez et al., 2009; Dua & Graff, 2017), including:

- a decreasing sequence of γ, preferring exploration first and then preferring exploitation in later rounds;
- adaptively modify γ using the history accuracy of exploitation rounds and exploration rounds;
- set an initial γ and a learning rate η, adaptively update γ at each round by either $\gamma = \gamma + \eta$ or $\gamma = \gamma - \eta$, depending on the history performance of exploitation and exploration.

Unfortunately, none of these hypotheses consistently outperform the constant-γ Banditron. However, we believe that Banditron with various γ is still promising because it allows more flexibility than vanilla Banditron and introduces the possibility of automation (the algorithm adaptively updates its γ). Further research can be done to better study this area and propose working solutions to learn the variable γ sequence.

References

Beygelzimer, A., Pál, D., Szorenyi, B., Thiruvenkatachari, D., Wei, C.-Y., and Zhang, C. Bandit multiclass linear classification: Efficient algorithms for the separable case. In *International Conference on Machine Learning*, pp. 624–633. PMLR, 2019.

Cortez, P., Cerdeira, A., Almeida, F., Matos, T., and Reis, J. Modeling wine preferences by data mining from physicochemical properties, 2009. URL `https://archive.ics.uci.edu/ml/datasets/wine+quality`.

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

Dua, D. and Graff, C. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Fink, M., Shalev-Shwartz, S., Singer, Y., and Ullman, S. Online multiclass learning by interclass hypothesis sharing. In *Proceedings of the 23rd international conference on Machine learning*, pp. 313–320, 2006.

Kakade, S. M., Shalev-Shwartz, S., and Tewari, A. Efficient bandit algorithms for online multiclass prediction. In *Proceedings of the 25th international conference on Machine learning*, pp. 440–447, 2008.

Rosenblatt, F. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.