Proceedings of Machine Learning Research 129:369–384, 2020 ACML 2020

Exact Passive-Aggressive Algorithms for Multiclass Classification Using Bandit Feedbacks

Maanik Arora MAANIK.ARORA@RESEARCH.IIIT.AC.IN
Machine Learning Lab, Kohli Center on Intelligent Systems, IIIT Hyderabad, India

Naresh Manwani NARESH.MANWANI@IIIT.AC.IN
Machine Learning Lab, Kohli Center on Intelligent Systems, IIIT Hyderabad, India

Editors: Sinno Jialin Pan and Masashi Sugiyama

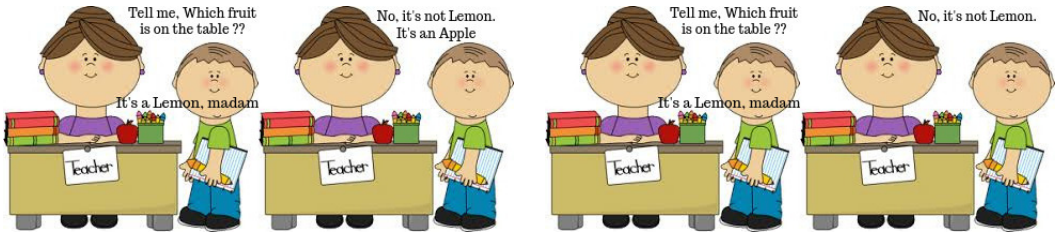
Abstract

In many real-life classification problems, we may not get exact class labels for training samples. One such example is bandit feedback in multiclass classification. In this setting, we only get to know whether our predicted label is correct or not. Due to which, we are left in uncertainty about the actual class label when we predict the wrong class. This paper proposes exact passive-aggressive online algorithms for multiclass classification under bandit feedback (EPABF). The proposed approach uses an exploration-exploitation strategy to guess the class label in every trial. To update the weights, we solve a quadratic optimization problem under multiple class separability constraints and find the exact solution. We do this by finding active constraints using the KKT conditions of the optimization problem. These constraints form a support set that determines the classes for which the weight vector needs to be updated. We propose three different variants of the weight update rule, which vary based on the aggressiveness to correct the mistake. These are called EPABF, EPABF-I, and EPABF-II. We also provide mistake bounds for the proposed EPABF, EPABF-I, and EPABF-II. Experiments demonstrated that our proposed algorithms perform better than other bandit feedback-based approaches and comparably to the full information approaches.

1. Introduction

Online learning of multiclass classifiers is an important and well-studied problem in machine learning. The goal is to classify examples into a set of classes. Digit recognition (Ma and Zhang, 2015), text classification (McCallum, 1999), recommendation systems (Li et al., 2016), optical character recognition (OCR), face recognition, tagging locations in vacation photos etc. are some of the well known applications of multiclass classifier.

Standard multiclass setting assumes that the exact (true) class labels of the training examples are given. This is called the full information case. However, in many real-life problems, getting true labels of the examples is a challenging task. Instead, we may get partial information about the class label. One such case is where we only know whether the predicted class label is the same as the true label. This is called *bandit feedback* setting (Sham M. Kakade, 2008). For example, when a user types a query in the recommendation system, she is presented with different links based on the user profile. A click on one of the link represents positive feedback, and no click corresponds to a negative response. On the other hand, the algorithm does not know what would have happened when other links been



(a) (b)

Figure 1: Example illustrating multiclass classification with (a) full information case and (b) bandit feedback case.

presented. Figure 1 presents the difference between the full information case and bandit information case using an example.

Consider the learning with bandit feedback as follows. The learner receives an instance $\mathbf{x}^t$ at trial t. The learner has to predict its label out of a set of K labels. Based on the current parameters, it predicts the label $\hat{y}^t$, where $\hat{y}^t \in [K]$. Then it receives a partial feedback $\mathbb{I}_{\{\hat{y}^t \neq y^t\}}$. Here y^t is the true label of $\mathbf{x}^t$. So if the prediction is correct, that is $\hat{y}^t = y^t$, the feedback is 1 and learner exactly knows that the true class is $\hat{y}^t$. On the other hand, if $\hat{y}^t \neq y^t$, the bandit feedback is 0 and the learner gets to know that $\hat{y}^t$ is not the true class. But, the learner still does not know the true class in this case. The performance of the learner is measured using the cumulative sum of mistakes in T trials, which is $\sum_{t=1}^{T} \mathbb{I}_{\{\hat{y}^t \neq y^t\}}$. Since the learning algorithm is unaware of the true label, it makes multiclass classification in the bandit setting harder than multiclass classification problems with a full information setting. In Sham M. Kakade (2008), authors extend the multiclass Perceptron (Crammer and Singer, 2003) to learn classifier using bandit feedback.

Inspired by passive-aggressive updates (Crammer et al., 2006), Zhong and Dauce (2015) propose bandit passive-aggressive algorithms (BPA) for multiclass classification using bandit feedbacks. Updates proposed in BPA are much more aggressive compared to Banditron (Sham M. Kakade, 2008). BPA shows better performance compared to Banditron (Zhong and Dauce, 2015). BPA finds the updates by solving a quadratic optimization problem in each trial. However, for multiclass classifier learning, it ignores the structure of multiple classes and reduces the problem to binary classification.

This paper proposes exact passive-aggressive algorithms under bandit feedback (EPABF) for online multiclass classification in the bandit setting. Our approach takes care of the issues in the method proposed in Zhong and Dauce (2015) by considering the separability conditions among different classes. At every trial, KKT optimality conditions are used to find the active constraints. These active constraints correspond to the support classes whose parameters will be updated. A similar approach was used for handling full information case (Matsushima et al., 2010). We observe that the proposed approaches perform better than the other bandit algorithms. Our key contributions in this paper are as follows.

1. We derive update rules for EPABF, EPABF-I, and EPABF-II under bandit feedback setting. EPABF algorithms update the parameters at trial t by minimizing convex

quadratic optimization problems. We find the exact solution to these optimization problems.

2. We propose a support class set finding algorithms for all the three variants and show their correctness.

3. We provide mistake bounds for all three variants EPABF, EPABF-I, and EPABF-II.

4. We perform simulations on different datasets and show the proposed algorithms' effectiveness compared to other bandit feedback algorithms and full information algorithms.

2. Related Work

Learning Under Full Information Setting

In the full information case, the learner knows the exact labels of all the examples in the training set. Popular batch learning algorithms for multiclass learning are discussed in (Crammer and Singer, 2002; Hsu and Lin, 2002; Ou and Murphey, 2007). In the online setting, we get instances sequentially, so the algorithm has to make predictions even when learning continuously. When it makes the wrong prediction, then it updates the parameters of the classifier else; it does not change the parameters. Perceptron is one of the earliest online algorithms for learning binary classifiers (Rosenblatt, 1960). Extensions of Perceptron algorithm for multiclass is proposed in Crammer and Singer (2003); Fink et al. (2006). Cesa-Bianchi et al. (2005) proposed a second-order Perceptron algorithm, which uses a second-order derivative in updating the parameters. Passive-aggressive learning is another framework in which the algorithm makes more aggressive updates to achieve the smallest loss on current misclassified example (Crammer et al., 2006) by solving a constrained quadratic optimization problem in each trial. However, for multiclass classifier learning, it ignores the structure of multiple classes and reduces the binary classification problem. Matsushima et al. (2010) proposed an exact passive-aggressive approach for the multiclass classification problem, which addresses the above issue. It first determines the set of classes called support classes when an input is received. Then the weight vectors are updated corresponding to all the support classes.

Learning Under Bandit Feedback

In many practical situations, the learner does not know the exact class label of the training examples. Instead, it may just observe some partial feedback. One such feedback is whether the predicted label is the same as the actual label. This is called bandit feedback. Several classification algorithms exist that address the bandit feedback setting. Banditron (Sham M. Kakade, 2008) extends the Perceptron algorithm to deal with the bandit feedback. Banditron uses an exploitation-exploration scheme proposed in Auer et al. (2003). When it updates, it replaces the gradient of the loss function with an unbiased estimator. When the data is linearly separable, the expected number of mistakes made by Banditron is $O(\sqrt{T})$ in T rounds. In the general case, the expected number of mistakes of Banditron is $O\left(T^{2=3}\right)$. Another bandit algorithm, named NEWTRON (Elad Hazan, 2011), is based

on the online Newton method. It uses strongly convex loss objective function (adding regularization term with the loss function) and Follow-The-Regularized-Leader (FTRL) strategy to achieve $O(\log T)$ regret bound in the best case and $O\left(T^{2/3}\right)$ regret bound in the worst case. Second-order Perceptron is also extended in bandit feedback setting by Crammer and Gentile (2011). It uses upper-con dence bounds (UCB) (Auer et al., 2002) based approach to make trade-o between exploration and exploitation and achieves regret bound of $O\left(\sqrt{T}\log(T)\right)$.

Recently, Beygelzimer et al. (2019) proposed an algorithm based on Kernel Perceptron. Authors proposed e cient online multiclass linear classi cation algorithms with bandit feedback when the data is linearly separable by a margin. They showed that their algorithm achieves a near-optimal bound of $O(K^{-1})$ under strong linear separability condition (Beygelzimer et al., 2019). Among the second-order algorithms, (Beygelzimer et al., 2017) proposed a second-order algorithm with $O(\frac{1}{\gamma}\sqrt{T})$ regret for the bandit online multiclass problem.

Zhong and Dauce (2015) extended the passive-aggressive online multiclass learning approach proposed in (Crammer et al., 2006) in the bandit feedback setting. They solved a constraint optimization problem in each trial to update the parameters which inherit similar issues as in the method proposed in (Crammer et al., 2006) (i.e., they solve a relaxed version of the quadratic optimization problem). In this paper, we use the ideas presented in Matsushima et al. (2010) to address the issues in BPA (Zhong and Dauce, 2015). We propose an exact passive-aggressive approach for multiclass classi ers under bandit feedback.

3. Passive-Aggressive Online Learning Under Full Information

The passive-aggressive online learning of multiclass classi er is proposed in Crammer et al. (2006). The learning is performed in a sequence of trials. At trial t, the learner is presented an example $x_t \in R^d$ and is required to predict the label out of a set of K labels. Let $[K] = \{1,\ldots,K\}$ denote the set of labels. Let $w_1^t,\ldots,w_K^t$ be the parameters of the multiclass classi er at the beginning of trial t. The label predicted by the algorithm at round t is given by, $\hat{y}_t = \arg\max_{j\in[K]}(w_j^t \cdot x^t)$. Then the algorithm observes the correct label y^t. The loss incurred by the algorithm is as follows.

$$l_{hinge}^t = \max[0, 1 - w_{y^t}^t \cdot x^t + \max_{j\neq y^t} w_j^t \cdot x^t] \tag{1}$$

where l_{hinge} is hinge loss for multiclass classi cation and is a convex surrogate of 0-1 loss Crammer et al. (2006). The passive-aggressive approach (Crammer et al., 2006) proposes three di erent ways to update the parameters as follows.

1. PA: In this approach, parameters $w_1^{t+1},\ldots,w_K^{t+1}$ are found such that they are closest to $w_1^t,\ldots,w_K^t$ and the loss incurred on x^t becomes zero.

$$w_1^{t+1},\ldots,w_K^{t+1} = \arg\min_{w_1,\ldots,w_K} \frac{1}{2}\sum_{j=1}^{K} \|w_j - w_j^t\|^2$$
$$\text{s.t.: } \max[0, 1 - w_{y^t} \cdot x^t + \max_{j\neq y^t} w_j \cdot x^t] = 0$$

2. PA-I: PA makes very aggressive updates as it tries to find the parameters which incur zero loss on x^t. To relax that, PA-I allows some error. Thus, PA-I makes a trade-off between the closeness to $w_1^t, \ldots, w_K^t$ and aggressiveness to minimize the loss on example x^t. C is the hyper-parameter which controls the aggressiveness.

$$w_1^{t+1}, \ldots, w_K^{t+1} = \arg\min_{w_1, \ldots, w_K} \frac{1}{2} \sum_{j=1}^{K} \| w_j - w_j^t \|^2 + C \xi$$
$$\text{s.t.} \quad \xi \geq 0, \; w_{y^t} \cdot x^t - \max_{j \neq y^t} w_j \cdot x^t \geq 1$$

3. PA-II: PA-II also allows some error, but it minimizes the square of the loss in contrast to PA-I. Thus, it finds new parameters as follows.

$$w_1^{t+1}, \ldots, w_K^{t+1} = \arg\min_{w_1, \ldots, w_K} \frac{1}{2} \sum_{j=1}^{K} \| w_j - w_j^t \|^2 + C \xi^2$$
$$\text{s.t.} \quad w_{y^t} \cdot x^t - \max_{j \neq y^t} w_j \cdot x^t \geq 1$$

Exact solutions of these three approaches are proposed in Matsushima et al. (2010). The update equations proposed in Matsushima et al. (2010) are as follows.

$$w_v^{t+1} = w_v^t - \tau_v^t (1 - I_{\{v = y^t\}}) x^t + I_{\{v = y^t\}} \sum_{v \neq y^t} \tau_v^t x^t \qquad v = 1, \ldots, K$$

where

$$\tau_v^t = \begin{cases} \frac{1}{\|x^t\|^2} - I_v^t & \frac{1}{|S^t|+1} \sum_{j \in S^t} I_j^t \; ; & \text{PA} \\ \frac{1}{\|x^t\|^2} - I_v^t & \max\left(\frac{\sum_{j \in S^t} I_j^t}{|S^t|} + \frac{C}{|S^t|} \|x^t\|^2 ; \frac{\sum_{j \in S^t} I_j^t}{|S^t|+1} \right) \; ; & \text{PA-I} \\ \frac{I_v^t}{\|x^t\|^2} - \frac{\|x^t\|^2 + \frac{1}{2C}}{(|S^t|+1)\|x^t\|^2 + \frac{|S^t|}{2C}} \sum_{j \in S^t} I_j^t ; & & \text{PA-II} \end{cases}$$

In the above, S^t denotes the support set which contains the indices of active constraints (Matsushima et al. (2010)). Exact passive-aggressive updates proposed in Matsushima et al. (2010) are shown to perform better than the updates proposed Crammer et al. (2006). However, passive-aggressive approach in Matsushima et al. (2010); Crammer et al. (2006) is developed under the full information case. That is, the algorithm knows the exact label y^t of x^t at every trial t. In this paper, we will propose the exact passive-aggressive approach under bandit feedback.

4. Proposed Approach: Exact Passive Aggressive Learning Under Bandit Feedback

Here again, learning happens in online fashion. At every trial t, an example $x^t \in R^d$ is presented to the algorithm. Let $w_1^t, \ldots, w_K^t$ be the parameters of the multiclass classifier at the beginning of the trial t, where K is the total number of classes. Then the algorithm

computes the label $\hat{y}^t = \arg\max_{j \in [K]} w_j^t . x^t$. Before asking for the bandit feedback, we form a probability distribution over different classes as follows.

$$P^t(r) = (1 - \beta) I_{[r = \hat{y}^t]} + \frac{\beta}{K}$$

Here $\beta \in (0, 0.5)$ is exploration parameter and $I_{[r=\hat{y}^t]}$ is the Indicator function. We randomly sample a label y_p^t from $P^t(r)$. We predict the sampled label y_p^t. Then, we ask for the bandit feedback $I_{[y_p^t = y^t]}$, which means if $y_p^t = y^t$, the feedback is 1, else it is 0. Thus, if $y_p^t \neq y^t$, the algorithm does not know which of the remaining $K - 1$ labels is the true label. The objective of the learning algorithm is to minimize the number of prediction mistakes, i.e. $\sum_{t=1}^{T} I_{[y_p^t \neq y^t]}$. However, we can't minimize $I_{[y_p^t \neq y^t]}$ as it is not continuous. Thus, we propose a new loss functions as follows.

We know that the l_{hinge}^t (Eq. (1)) is a convex surrogate of the 0-1 loss. However, one can easily see that

$$l_{hinge}^t = \max(0, 1 - a_t w_{y_p^t}^t . x^t + \max_{r \neq y^t} w_r^t . x^t) \leq \sum_{r=1}^{K} \max(0, 1 - a_t w_{y_p^t}^t . x^t + w_r^t . x^t) = \sum_{r=1}^{K} l_r^t .$$

Thus, $\sum_{r=1}^{K} l_r^t$ also forms a convex surrogate for 0-1 loss.

We use $\theta_r^t = \max(0, 1 - a_t w_{y_p^t}^t . x^t + w_r^t . x^t)$ as an estimator for l_r^t in presence of bandit feedback, where $a_t = I_{[y_p^t = y^t]} / P(y_p^t)$. The total loss incurred at time t is given as $\sum_{r=1}^{K} \theta_r^t$. Note that each of θ_r^t is a random variable as it depends on y_p^t. We observe the following.

$$\begin{aligned} E[\theta_r^t] &= E[\max(0, 1 - a_t w_{y_p^t}^t . x^t + w_r^t . x^t)] \leq \max(0, E[1 - w_{y_p^t}^t . x^t a_t + w_r^t . x^t]) \\ &\leq \max(0, 1 - w_{y^t}^t . x^t + w_r^t . x^t) = l_r^t \end{aligned}$$

$E[\sum_{r=1}^{K} \theta_r^t]$ upper bounds $\sum_{r \neq y^t} l_r^t$. Thus, minimizing $\sum_{r=1}^{K} \theta_r^t$ is an appropriate objective at every trial. We now derive the update equations under exact passive aggressive framework using bandit feedback. As discussed earlier, there are three variants of the passive-aggressive approach, namely, PA, PA-I, and PA-II. We name them as EPABF (Exact Passive Aggressive approach under Bandit Feedback), EPABF-I and EPABF-II respectively.

4.1. EPABF Updates

In EPABF approach, after receiving the bandit feedback $I_{[y_p^t = y_t]}$, the new parameters of the classifier are found as follows. If $\sum_{r=1}^{K} \theta_r^t = 0$, then we do not update the parameters. On the other hand, if the $\sum_{r=1}^{K} \theta_r^t > 0$, then we find the new parameters by solving the following optimization problem.

$$\begin{aligned} w_1^{t+1}, \ldots, w_K^{t+1} &= \arg\min_{w_1, \ldots, w_K} \frac{1}{2} \sum_{v=1}^{K} \|w_v - w_v^t\|^2 \quad s.t. \quad \sum_{r=1}^{K} \theta_r^t = 0 \\ &= \arg\min_{w_1, \ldots, w_K} \frac{1}{2} \sum_{v=1}^{K} \|w_v - w_v^t\|^2 \quad s.t. \quad a_t w_{y_p^t}^t . x^t - w_r . x^t \geq 1, \forall r \in [K] \end{aligned}$$

This is a quadratic optimization problem with K linear constraints (one constraint corresponding to each class. Let S^t denotes the support class set which contain indices of active constraint. Final update equations for EPABF are as follows.

$$w_r^{t+1} = w_r^t - \tau_r^t x^t + a_t \mathbb{I}_{[r=\hat{y}^t]} \sum_{i \in S^t} \tau_i^t x^t; \quad r = 1 \ldots K$$

where

$$\tau_r^t = \begin{cases} \frac{1}{\|x^t\|^2} \left(\beta_r^t + \frac{a_t \beta_{\hat{y}^t}^t}{1 + |S^t| a_t^2} - \frac{a_t^2 \sum_{i \in S^t} \beta_i^t}{1 + |S^t| a_t^2} \right) & r = \hat{y}^t \in S^t \\ \frac{1}{\|x^t\|^2} \left(\beta_r^t - \frac{a_t^2}{a_t^2 |S^t| + 1} \sum_{i \in S^t} \beta_i^t \right) & r = \hat{y}^t \notin S^t \end{cases} \tag{2}$$

The derivation of the step sizes is given in the supplementary file. Thus, we can determine the step sizes of τ_r^t and obtain a complete update rule if the support class set S^t is known. The overall approach of learning classifier using EPABF is described in Algorithm 1.

Algorithm 1 Exact Passive-Aggressive Approach Under Bandit Feedback (EPABF)

Input: $\in (0, 0.5)$, Training Set Q
Initialize: $w_1^0, \ldots, w_K^0$
for $t = 1, \ldots, T$ **do**
  Receive $x^t \in Q$
  Set $\hat{y}^t = \arg\max_{r \in [K]} (w_r^t \cdot x^t)$
  for $r = 1, \ldots, K$ **do**
    Define $P^t(r) = (1 -) \mathbb{I}_{[r=\hat{y}^t]} + \frac{}{K}$
  end for
  Sample $\tilde{y}^t \sim P^t$. Receive feedback $\mathbb{I}_{[\tilde{y}^t = y_t]}$.
  for $r = 1, \ldots, K$ **do**
    Compute $\beta_r^t = [1 - a_t w_{\tilde{y}^t}^t \cdot x^t + w_r^t \cdot x^t]_+$
  end for
  $S^t = \text{SCA}(\beta_1^t, \ldots, \beta_K^t, a_t)$
  if $\tilde{y}^t \in S^t$ **then**
    for $r \in S^t$ **do**
      $\tau_r^t = \frac{\beta_r^t}{\|x^t\|^2} + \frac{\|x^t\|^2 a_t \beta_{\tilde{y}^t}^t}{1 + |S^t| a_t^2} - \frac{\|x^t\|^2 a_t^2 \sum_{i \in S^t} \beta_i^t}{1 + |S^t| a_t^2}$
    end for
  else if $\tilde{y}^t \notin S^t$ **then**
    for $r \in S^t$ **do**
      $\tau_r^t = \frac{1}{\|x^t\|^2} \left(\beta_r^t - \frac{a_t^2 \sum_{i \in S^t} \beta_i^t}{1 + |S^t| a_t^2} \right)$
    end for
  end if
  Set $\tau_r^t = 0, \forall r \notin S^t$
  for $r = 1 \ldots K$ **do**
    $w_r^{t+1} = w_r^t - \tau_r^t x^t + a_t \mathbb{I}_{[r=\hat{y}^t]} \sum_{i \in S^t} \tau_i^t x^t$
  end for
end for
Output: $w_1^{T+1}, \ldots, w_K^{T+1}$

Algorithm 2 Support Class Set Finding Algorithm for EPABF (SCA)

Input: $\beta_1^t, \ldots, \beta_K^t, a_t$
Initialize: $S^t :=$
Sort $\beta_1^t \ldots \beta_K^t$ in descending order. Let $(1) \ldots (K)$ be the sorted order.
for $j = 1 \ldots K$ **do**
  if $\tilde{y}^t \in S^t$ **then**
    if $\frac{a_t^2}{1 + |S^t| a_t^2} \sum_{k \in S^t} \beta_{(k)}^t < \beta_{(j)}^t$ **then**
      $S^t = S^t \cup (j)$
    end if
  else if $\tilde{y}^t \notin S^t$ **then**
    if $\frac{a_t^2}{1 + |S^t| a_t^2 - a_t} \sum_{k \in S^t} \beta_{(k)}^t < \beta_{(j)}^t + \frac{a_t \beta_{\tilde{y}^t}^t}{1 + |S^t| a_t^2 - a_t}$ **then**
      $S^t = S^t \cup (j)$
    end if
  end if
end for
Output: S^t

Determining Support Class Set S^t The EPABF algorithm assumes requires the support class set S^t as an input. Following theorem states the necessary and sufficient condition for any class belonging to the support class set.

Theorem 1 Assume that at trial t, $S^t \neq \emptyset$. Let (k) be the k-th class in the sorted order of $\beta_1^t, \ldots, \beta_K^t$. The necessary and sufficient condition for (k) to be in set S^t is as follows.

$$\beta_{(k)}^t > \begin{cases} \dfrac{a_t^2 \sum_{j=1}^{k-1} \beta_{(j)}^t}{1+(k-1)a_t^2 - a_t} & \dfrac{a_t \beta_{y^t}^t}{1+(k-1)a_t^2 - a_t} \quad y^t \in S^t \\ \dfrac{a_t^2 \sum_{j=1}^{k-1} \beta_{(j)}^t}{1+(k-1)a_t^2} & y^t \notin S^t \end{cases}$$

The proof of this theorem is given in supplementary file. Theorem 1 allows us to algorithmically determine the support classes S^t. We simply sort classes according to β_r^t values in descending order, and the top J classes in the sorted list are the support classes (where J is the largest k that satisfies the Theorem 1 statement). Support class set finding algorithm (SCA) is described in Algorithm 2.

4.2. EPABF-I Updates

EPABF-I finds new parameters which are closest to the previous parameters and achieve minimum loss on the current example. This is done by solving the following problem.

$$\begin{aligned} w_1^{t+1} \ldots w_K^{t+1} = \arg\min_{w_1,\ldots,w_K} & \frac{1}{2}\sum_{v=1}^{K} \|w_v - w_v^t\|^2 + C\sum_{v=1}^{K} \xi_v \\ \text{s.t.:} \quad & a_t w_{y^t}^t x^t - w_r^t x^t \geq 1 - \xi_r, \quad \xi_r \geq 0, \quad r \in [K] \end{aligned} \tag{3}$$

Here $C > 0$ is a parameter to the trade-off between the two parts of the objective function. The larger the value of C, the focus is more on minimizing the loss. The resulting parameter update equations are as follows.

$$w_r^{t+1} = w_r^t - \tau_r^t x^t + a_t I_{\{r=y^t\}} \sum_{i \in S^t} \tau_i^t x^t; \quad r = 1 \ldots K$$

where

$$\tau_r^t = \begin{cases} \min\left(C, \dfrac{1}{\|x^t\|^2}\left(\beta_r^t + \dfrac{a_t \beta_{y^t}^t}{a_t^2 |S^t| + 1 - a_t}\left(\dfrac{a_t^2 \sum_{v \in S^t} \beta_v^t}{a_t^2 |S^t| + 1 - a_t}\right)\right)\right) & y^t \in S^t \\ \min\left(C, \dfrac{1}{\|x^t\|^2}\left(\beta_r^t - \dfrac{a_t^2 \sum_{v \in S^t} \beta_v^t}{a_t^2 |S^t| + 1}\right)\right) & y^t \notin S^t \end{cases}$$

A complete derivation of the update equation and τ_r^t is given in the supplementary file. To determine the support set S^t, EPABF-I uses a slightly different approach compared to Algorithm 2. EPABF-I uses an iterative process to find the values of τ_r^t until all the values converge. Support class algorithm (SCA-I) for EPABF-I is given in Algorithm 3.

4.3. EPABF-II Updates

EPABF-II algorithm also makes a trade-o between the closeness to the current weight vector and aggressiveness to minimize the loss on the current instance. But in EPABF-II we minimize square of the loss while considering the trade-o .

$$w_1^{t+1} ::: w_K^{t+1} = \arg\min_{w_1 ::: w_K} \frac{1}{2} \sum_{v=1}^{K} \| w_v - w_v^t \|^2 + C \sum_{v=1}^{K} \xi_v^2$$
$$\text{s:t: } a_t w_{y^t}^t x^t - w_r^t x^t \geq 1 - \delta_{y^t r} - \xi_r, \; r \in [K]$$

Algorithm 3 Support Class Set Finding Algorithm for EPABF-I (SCA-I)

Input: $\hat{y}_1^t, \ldots, \hat{y}_K^t, a_t$
Initialize: $S^t :=$
Sort $\hat{y}_1^t ::: \hat{y}_K^t$ in descending order. Let $(1) ::: (K)$ be the sorted order.
while $\tau_1^t, \ldots, \tau_K^t$ do not converge do
  for $r = 1 ::: K$ do
    if $r \in S^t$ then
      $\tau_{(r)}^t = \min\left(C, \frac{1}{\|x^t\|^2}\left(\hat{y}_{(r)} - \frac{a_t \hat{y}_{y^t}}{a_t^2 |S^t| + 1} - \frac{a_t^2 \sum_{v \in S^t} \hat{y}_v}{a_t^2 |S^t| + 1}\right)\right)$
    else if $r \neq y^t$ then
      $\tau_{(r)}^t = \min\left(C, \frac{1}{\|x^t\|^2}\left(\hat{y}_{(r)} - \frac{a_t^2 \sum_{v \in S^t} \hat{y}_v}{a_t^2 |S^t| + 1}\right)\right)$
    end if
    if $\tau_{(r)}^t > 0$ then
      $S^t = S^t \cup (r)$
    else
      $S^t = S^t - (r)$
      $\tau_{(r)}^t = 0$
    end if
  end for
end while
Output: S^t

Algorithm 4 Support Class Set Finding Algorithm for EPABF-II (SCA-II)

Input: $\hat{y}_1^t, \ldots, \hat{y}_K^t, a_t$
Initialize: $S^t :=$
Sort $\hat{y}_1^t ::: \hat{y}_K^t$ in descending order. Let $(1) ::: (K)$ be the sorted order.
for $j = 1 ::: K$ do
  if $y^t \in S^t$ then
    if $\frac{a_t^2}{a_t^2 |S^t| + 1 + \frac{1}{2C\|x^t\|^2}} \sum_{k \in S^t} \hat{y}_{(k)} < \hat{y}_{(j)}$ then
      $S^t = S^t \cup (j)$
    end if
  else if $y^t \notin S^t$ then
    if $\frac{a_t^2 \sum_{k \in S^t} \hat{y}_{(k)} - a_t \hat{y}_{y^t}}{a_t^2 |S^t| + 1 - a_t + \frac{1}{2C\|x^t\|^2}} < \hat{y}_{(j)}$ then
      $S^t = S^t \cup (j)$
    end if
  end if
end for
Output: S^t

Parameter $C > 0$ is used to make trade o between the two parts of the objective function. For larger value of C, more emphasis is given to minimize the square of the loss. The resulting parameter update equations are as follows.

$$w_r^{t+1} = w_r^t - \tau_r^t x^t + a_t \mathbb{I}_{\{r = y^t\}} \sum_{i \in S^t} \tau_i^t x^t, \; r = 1 ::: K$$

where

$$\tau_r^t = \begin{cases} \dfrac{1}{\|x^t\|^2 \left(1 + \frac{1}{2C\|x^t\|^2}\right)} \left(\theta_r^t + \dfrac{a_t \theta_{y^t}^t - a_t^2 \sum_{j \in S^t} \theta_j^t}{a_t^2 |S^t| + 1 - a_t + \frac{1}{2C\|x^t\|^2}}\right) & y^t \in S^t \\ \dfrac{1}{\|x^t\|^2 \left(1 + \frac{1}{2C\|x^t\|^2}\right)} \left(\theta_r^t - \dfrac{a_t^2 \sum_{j \in S^t} \theta_j^t}{a_t^2 |S^t| + 1 + \frac{1}{2C\|x^t\|^2}}\right) & y^t \notin S^t \end{cases}$$

The derivations of the step size is given in supplementary file. Using this step size, we get following method to determine the support class.

Determining Support Class Set S^t Following theorem states the necessary and sufficient condition for any class belonging to the support class set.

Theorem 2 Assume that at trial t, $S^t \neq \emptyset$. Let (k) be the k-th class in the sorted order of $\theta_1^t, \ldots, \theta_K^t$. The necessary and sufficient condition for (k) belongs to S^t is as follows.

$$\theta_{(k)}^t > \begin{cases} \dfrac{a_t^2 \sum_{j=1}^{k-1} \theta_{(j)}^t - a_t \theta_{y^t}^t}{a_t^2 (k-1) + 1 - a_t + \frac{1}{2C\|x^t\|^2}} & y^t \in S^t \\ \dfrac{a_t^2}{a_t^2 (k-1) + 1 + \frac{1}{2C\|x^t\|^2}} \sum_{j=1}^{k-1} \theta_{(j)}^t & y^t \notin S^t \end{cases}$$

The proof of this theorem can be found in the supplementary file.

5. Mistake Bound Analysis

Here, we find the mistake bounds for the proposed algorithms EPABF, EPABF-I and EPABF-II. Note that, $E[\sum_{v=1}^K \theta_v^t] \leq \sum_{v=1}^K l_v^t$ and $\sum_{v=1}^K l_v^t \leq l_{hinge}^t \leq l_{0-1}^t$. Thus, $\sum_{t=1}^T \sum_{v=1}^K E[\theta_v^t]$ upper bounds the number of mis-classifications over a sequence of examples. Similarly, $E[\sum_{v=1}^K (\theta_v^t)^2] \leq \sum_{v=1}^K (E[\theta_v^t])^2 \leq \sum_{v=1}^K (l_v^t)^2 \leq l_{0-1}^t$. Thus, $\sum_{t=1}^T \sum_{v=1}^K E[(\theta_v^t)^2]$ also upper bounds the number of mis-classifications over a sequence of examples.

Theorem 3 (Mistake Bound of EPABF) Let $x^1 \ldots x^T$ be the sequence of examples presented to EPABF. Let $u_1, \ldots, u_K$ be the parameters of an arbitrary linear classifier. Let $\theta_r^t = [1 - a_t u_{y^t} x^t + u_r x^t]_+$, $r \in [K]$. Then, EPABF algorithm satisfies following mistake bound.

$$\sum_{t=1}^T \sum_{v=1}^K E[(\theta_v^t)^2] \leq R^t \sum_{v=1}^K \|u_v\|^2 + 4 \sqrt{\sum_{t=1}^T \sum_{v=1}^K E[(\theta_v^t)^2]}^2$$

Where $R = \max_{t \in [T]} \|x^t\|_2$ and $= \frac{K^3}{2} + 1$.

The proof of the theorem can be found in supplementary le. The regret bound increases with K (number of classes). In the ideal case $l_v^t = 0; 8v \in [K]; 8t \in [T]$), the regret bound reduces to $R\sqrt{\sum_{v=1}^{K} \|u_v\|^2}$, which is similar to the regret achieved in the full information case (Matsushima et al., 2010).

Theorem 4 (Mistake Bound of EPABF-I) Let $x^1 ::: x^T$ be the sequence of examples presented to EPABF-I. Let $u_1; ::: ; u_K$ be the parameters of an arbitrary linear classi er. Let $\hat{e}_r^t = [1 - a_t u_{y^t} \cdot x^t + u_r \cdot x^t]_+ ; r \in [K]$. Let K be the total number of classes and $R = \max_{t \in [T]} \|x^t\|_2$. Then, EPABF-I achieves the following mistake bound.

$$\frac{1}{R^2}\left(1 - \frac{1}{K}\right)\sum_{t=1}^{T}\left(\sum_{v=1}^{K}\hat{e}_v^t\right)^2 + \frac{1}{2}\sqrt{\sum_{v=1}^{K}\|u_v\|^2\left(TKR^2 + \frac{TK^4R^2}{2} + T\right)}$$

The proof of the theorem can be found in supplementary le. The general regret bound is of order $O(\sqrt{T})$ which is at par with the majority of the bandit feedback algorithms. The regret bound increases with increase in number of classes. In the ideal case $l_v^t = 0; 8v \in [K]; 8t \in [T]$), the bound reduces to $\frac{1}{2}\sqrt{\sum_{v=1}^{K}\|u_v\|^2\left(TKR^2 + \frac{TK^4R^2}{2} + T\right)}$.

Theorem 5 (Mistake Bound of EPABF-II) Let $x^1 ::: x^T$ be the sequence of examples presented to EPA. Let $u_1; ::: ; u_K$ be the parameters of an arbitrary linear classi er. Let $\hat{e}_r^t = [1 - a_t u_{y^t} \cdot x^t + u_r \cdot x^t]_+ ; r \in [K]$ and $R = \max_{t \in [T]} \|x^t\|_2$. Then, EPABF-II algorithm satis es following mistake bound.

$$\sum_{t=1}^{T}\left(\sum_{v=1}^{K}\hat{e}_v^t\right)^2 \le M\left(\sum_{v=1}^{K}\|u_v\|^2 + 2CR^2\sum_{t=1}^{T}\left(\sum_{v=1}^{K}\hat{e}_v^t\right)^2\right)$$

where $M = \dfrac{R^2 + \frac{1}{2C}}{2K + \frac{1}{C}}$

The proof of the theorem can be found in supplementary le. In the ideal case when $l_v^t = 0; 8v \in [K]; 8t \in [T]$, the bound reduces to $M\sum_{v=1}^{K}\|u_v\|^2$.

6. Experiments

In this section, we provide experimental results to show the e ectiveness of the proposed approach on synthetic and UCI datasets (Dua and Gra (2017)).

Datasets Used We report experimental results on two synthetic datasets and 5 UCI datasets (Dua and Gra (2017)). Synthetic datasets are generated as follows.

1. SynSep: It is a 10-class, 100-dimensional data set of 1,00,000 examples generated as follows. We randomly generate vectors from $[-1;1]^{100}$. We use $u_1;\ldots;u_{10} \in R^{100}$ as the discriminant vectors for 10 classes. All the entries of u_i are 1 except for indices $i + 10j;\ j = 0:::9$. Entries at indices $i + 10j;\ (j = 0:::9)$ are -1. Class label for a given example x is assigned as $y_x = \arg\max_{i \in \{1,\ldots,10\}} (u_i : x)$. This data is linearly separable.

2. SynNonSep: This dataset is constructed in the same way as SynSep except that we introduce 5% label noise. This makes the data set linearly non-separable.

The UCI datasets (Dua and Gra , 2017) used are Ecoli, Abalone, Satimage, Iris and USPS. Abalone dataset is not balanced as there is asymmetry in its class distribution, so we divided the Rings attribute into 4 intervals as 1-7, 8-14, 15-21, 22-29. Thus, we reduce it to 4 class classi cation problem.

Benchmark Algorithms: We present experimental comparisons of the proposed algorithms (EPABF, EPABF-I, and EPABF-II) with Banditron (Sham M. Kakade (2008)), Bandit Passive-Aggressive (BPA) (Zhong and Dauce (2015)). We also compare the proposed approaches with PA, PA-I, and PA-II algorithms (Matsushima et al. (2010)), which are full information based algorithms. This means PA, PA-I, and PA-II use exact class labels for training. We do this to see how much performance degradation happens due to bandit feedback.

Experimental results are shown in Figures 4. For every dataset, we have ran each algorithm for 100,000 iterations. We have plotted the error curves by averaging the error rates(ratio of incorrectly classi ed and number of rounds) over 100 di erent runs. The nal plots for each dataset have the average instantaneous error rate on the Y-axis and the number of trials on the X-axis.

(a) (b) (c)

Figure 2: Converged Error Rates versus and and C for Iris Dataset. Note that and C are on the log scale.

Choosing Optimal and C for the Proposed Algorithms: EPABF algorithm takes is a user-defined parameter. Similarly, EPABF-I and EPABF-II require and C values as input. To choose the best values of these hyper-parameters, we use the following approach. We explain it for Iris dataset. Figure 2 shows the trend of converged error rate with gamma $\log_2(\ )$ and $\log_2(C)$ for EPABF algorithms for Iris dataset. We have chosen the best value of and C for which the minimum error rate is achieved. We found the optimal values of and C for all the datasets using the same method. The optimal values of C and for different datasets are shown below in the Tables 1, 2 and 3. The final results shown are based on taking these optimal hyper-parameters values. We use the same approach for selecting the hyper-parameters for the benchmark algorithms also.

Table 1: EPABF

Dataset	
SynSep	0.0001
SynNonSep	0.001
Ecoli	0.04
Abalone	0.008
Satimage	0.065
Iris	0.035
USPS	0.06

Table 2: EPABF-I

Dataset	C	
SynSep	0.1	0.0003
SynNonSep	0.1	0.001
Ecoli	0.01	0.05
Abalone	0.01	0.02
Satimage	0.1	0.06
Iris	0.03	0.025
USPS	0.1	0.1

Table 3: EPABF-II

Dataset	C	
SynSep	0.1	0.0001
SynNonSep	0.1	0.0001
Ecoli	0.01	0.04
Abalone	0.01	0.03
Satimage	0.1	0.06
Iris	0.03	0.04
USPS	0.1	0.1

Comparison of EPABF algorithms with Other Bandit Algorithms: Figure 3 presents comparison results of the proposed EPABF and its variants with other bandit feedback based algorithms (i.e., BPA and Banditron). We observe that the proposed EPABF variants outperform Banditron by a significant margin in terms of converged error rates for all the datasets. Compared to BPA, the proposed algorithms perform significantly better except for the Ecoli, USPS, and Iris datasets. For Ecoli, USPS, and Iris datasets, proposed approaches still perform better than BPA though marginally. For Iris and Satimage, EPABF variants take more time to converge compared to BPA and Banditron but achieve a lower error rate.

We see that from SynSep to SynNonSep, the performances of BPA and Banditron drop by huge amount compared to EPABF variants.

Comparison with Full Information Case: Figure 4 presents comparison results of the proposed algorithms with full information algorithms, namely, PA, PA-I, and PA-II. For Ecoli and USPS datasets, the proposed EPABF algorithms perform comparably to the full information counterparts. For Iris, Satimage, and USPS datasets, the full information algorithms perform better than the proposed approach.

On the other hand, for SynSep, SynNonSep, and Abalone datasets, the proposed bandit algorithms perform better than the full information based algorithms. A similar kind of behavior was observed in Zhong and Dauce (2015). This might happen because of the following reason. In the proposed algorithms, we are using an exploration-exploitation scheme. Unlike the full information algorithms, we do not always choose the best-predicted label, which is still selecting the best labels. Thus, over a broad set of examples, the actual

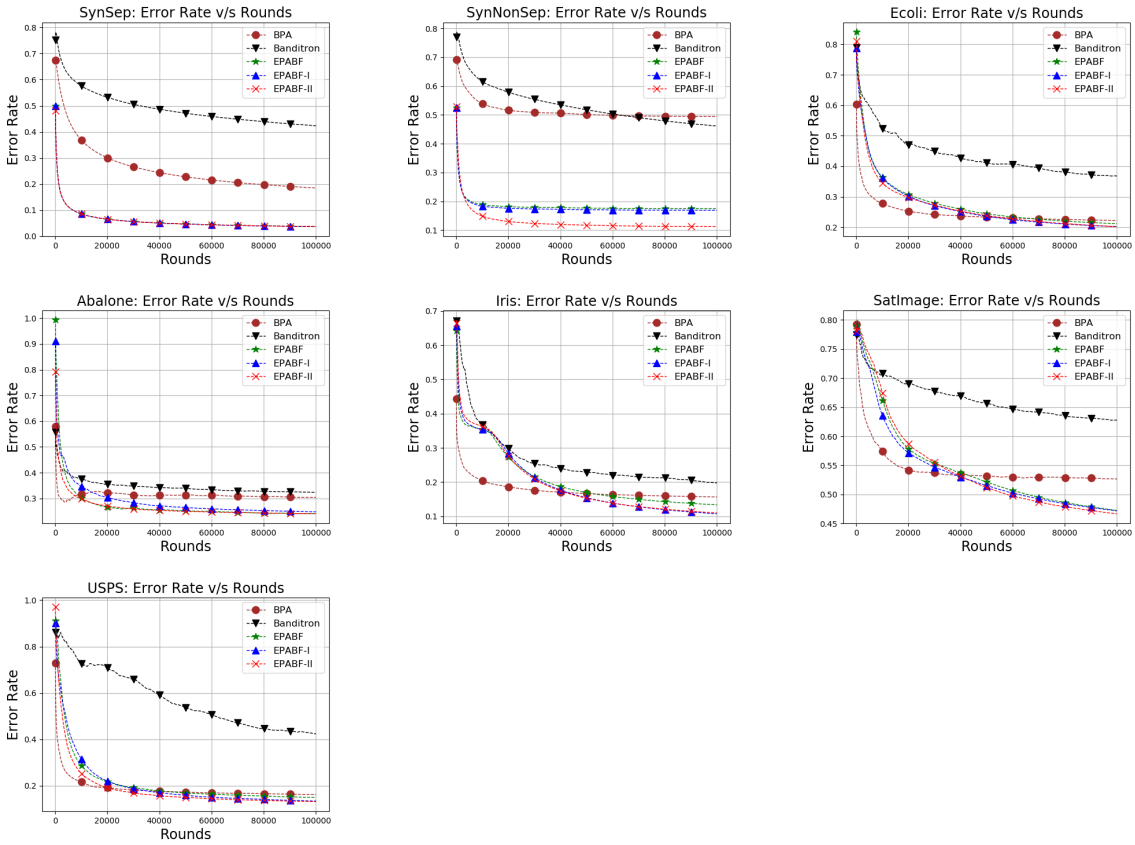


Figure 3: Performance comparison of EPABF, EPABF-I and EPABF-II with Bechmark Bandit Algorithms (BPA and Banditron).

labels match the labels predicted by EPABFs. This can lead to better performances of EPABF algorithms compared to full information based algorithms.

7. Conclusion

This paper proposed three variants of passive-aggressive online algorithms for multiclass classification using bandit feedback, namely EPABF, EPABF-I, and EPABF-II. We used exploration-exploitation to guess the best label of an example. To update the parameters, we solve a constrained optimization problem in each trial. The constraints are used to capture the separability conditions of the multiclass problem. Finally, we update the parameter of those classes whose corresponding constraints are active. These classes are called support classes. We also provided regret bounds for all three variants. We also offered an experimental comparison of all the algorithms on various datasets. The results showed that our EPABF algorithms perform better than existing bandit feedback based algorithms and perform comparably to the full feedback based algorithms on most of the given datasets.

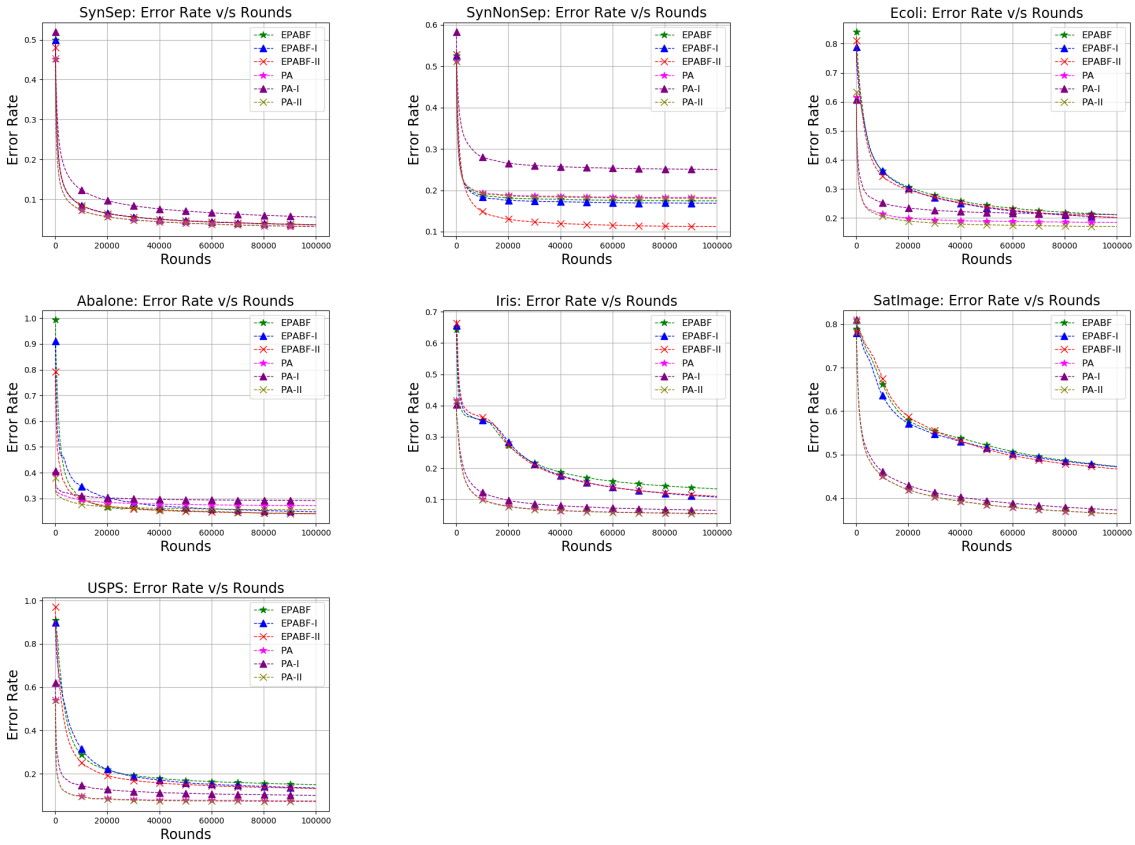


Figure 4: Performance comparison of EPABF, EPABF-I and EPABF-II with full information algorithms (PA, PA-I and PA-II).

References

Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Mach. Learn.*, 47(2-3):235–256, May 2002.

Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM J. Comput.*, 32(1):48–77, January 2003.

Alina Beygelzimer, Francesco Orabona, and Chicheng Zhang. Efficient online bandit multiclass learning with $\tilde{O}(\sqrt{T})$ regret. 02 2017.

Alina Beygelzimer, David Pal, Balazs Szorenyi, Devanathan Thiruvenkatachari, Chen-Yu Wei, and Chicheng Zhang, editors. *Bandit Multiclass Linear Classification: Efficient Algorithms for the Separable Case*, ICML, 2019.

Nicolò Cesa-Bianchi, Alex Conconi, and Claudio Gentile. A second-order perceptron algorithm. *SIAM Journal on Computing*, 34:640–668, January 2005.