Mach Learn (2016) 103:141–183
DOI 10.1007/s10994-016-5555-y

Online Passive-Aggressive Active learning

Jing Lu[1] · **Peilin Zhao**[2] · **Steven C. H. Hoi**[1]

Received: 24 March 2015 / Accepted: 23 February 2016 / Published online: 22 March 2016

Abstract We investigate online active learning techniques for online classification tasks. Unlike traditional supervised learning approaches, either batch or online learning, which often require to request class labels of each incoming instance, online active learning queries only a subset of informative incoming instances to update the classification model, aiming to maximize classification performance with minimal human labelling effort during the entire online learning task. In this paper, we present a new family of online active learning algorithms called Passive-Aggressive Active (PAA) learning algorithms by adapting the Passive-Aggressive algorithms in online active learning settings. Unlike conventional Perceptron-based approaches that employ only the misclassified instances for updating the model, the proposed PAA learning algorithms not only use the misclassified instances to update the classifier, but also exploit correctly classified examples with low prediction confidence. Specifically, we propose several variants of PAA algorithms to tackle three types of online learning tasks: binary classification, multi-class classification, and cost-sensitive classification. We give the mistake bounds of the proposed algorithms in theory, and conduct extensive experiments to evaluate the empirical performance of our techniques on both standard and large-scale datasets, in which the encouraging results validate the empirical effectiveness of the proposed algorithms.

Editors: Hang Li, Dinh Phung, Tru Cao, Tu-Bao Ho, and Zhi-Hua Zhou.

✉ Steven C. H. Hoi
chhoi@smu.edu.sg

Jing Lu
jing.lu.2014@phdis.smu.edu.sg

Peilin Zhao
zhaop@i2r.a-star.edu.sg

[1] School of Information Systems, Singapore Management University, Singapore 178902, Singapore

[2] Institute for Infocomm Research, A*STAR, Singapore 138632, Singapore

Keywords Online learning · Active learning · Passive-aggressive · Cost-sensitive classification · Multi-class classification

1 Introduction

Both online learning and active learning have been extensively studied in machine learning and data mining (Freund et al. 1997; McCallum and Nigam 1998; Balcan et al. 2006, 2007; Cesa-Bianchi and Lugosi 2006; Crammer et al. 2006; Castro and Nowak 2007; Zhao and Hoi 2010; Hoi et al. 2014). In a traditional online learning task (e.g., online classification), a learner is trained in a sequential manner to predict the class labels of a sequence of instances as accurately as possible. Specifically, at each round of a typical online learning task, the learner first receives an incoming instance, and then makes a prediction of its class label. After that, it is assumed to *always* receive the true class label from an oracle, which can be used to measure the loss incurred by the learner's prediction so as to update the learner if necessary. In many real-world applications especially for mining real-life sequential arriving data (e.g., spam email filtering), acquiring the true class labels from an oracle is often time-consuming and costly due to the unavoidable interaction between the learner and the environment. This has motivated the recent study of Online Active Learning (Cesa-Bianchi et al. 2006; Dasgupta et al. 2009; Cesa-Bianchi and Lugosi 2006; Sculley 2007), which explores active learning strategy in an online learning setting to avoid requiring to request class labels of all incoming instances.

A pioneering and state-of-the-art technique to online active learning is known as Label Efficient Perceptron (Cesa-Bianchi and Lugosi 2006) or Selective Sampling Perceptron (Cesa-Bianchi et al. 2006; Cavallanti et al. 2009), or called Perceptron-based Active Learning (Dasgupta et al. 2009). In particular, consider an online classification task, when a learner receives an incoming instance $\mathbf{x}_t$, the learner first makes a prediction $\hat{y}_t = \text{sign}(f(\mathbf{x}_t))$ where $f(\mathbf{x}_t) = \mathbf{w}_t \cdot \mathbf{x}_t$, and then uses a stochastic approach to decide whether it should query the class label or not, where the query probability is inversely proportional to the prediction confidence (e.g., the magnitude of the margin, i.e., $\delta/(\delta + |f(\mathbf{x}_t)|)$ where δ is a positive smoothing constant). If no class label is queried, the learner makes no update; otherwise, it acquires the true label y_t from the environment and follows the regular Perceptron algorithm to make update (i.e., the learner will update the model if and only if the instance is misclassified according to the true class label). We summarize the common Framework of Online Active learning algorithms in Algorithm 1.

In the above Perceptron-based active learning, if an incoming instance is predicted with low confidence by the current model, the learner very likely would query its class label. However, if the instance is correctly classified according to the acquired true label, this training instance will be discarded and never be used to update the learner according to the principle of the Perceptron algorithm. Clearly this is a critical limitation of wasting the effort of requesting class labels. To overcome this limitation, we present a new scheme for online active learning, i.e., the Passive-Aggressive Active (PAA) learning, which explores the principle of passive-aggressive learning (Crammer et al. 2006). It not only decides when the learner should make a query appropriately, but also attempts to fully exploit the potential of every queried instance for updating the classification model. To tackle different kinds of machine learning tasks, we propose several variants of the PAA algorithms, i.e. the PAA algorithm for online binary classification tasks, the Multi-class Passive-Aggressive Active (MPAA) learning algorithm for online multi-class classification tasks and the Cost-Sensitive

Algorithm 1 The Framework of Online Active learning

```
INITIALIZE : classifier $f_0$.
for $t = 1, \dots, T$ do
  Observe: $\mathbf{x}_t \in \mathbb{R}^d$, make prediction $\hat{y}_t$ based on $f_t(\mathbf{x}_t)$;
  Make a decision whether to query the label ($Z_t = 1$) or not ($Z_t = 0$);
  if $Z_t = 1$ then
    Query label $y_t$, and suffer loss $\ell_t(f_t)$;
    if $\ell_t(f_t) \neq 0$ then
      Update: calculate classifier $f_{t+1}$;
    else
      $f_{t+1} = f_t$
    end if
  else
    $f_{t+1} = f_t$
  end if
end for
```

Passive-Aggressive Active learning (CSPAA) algorithm for online binary classification with extremely unbalanced data. We theoretically analyse the mistake bounds of the proposed algorithms and conduct extensive experiments to examine their empirical performance, in which encouraging results show clear advantages of our algorithms over the baselines.

The rest of this paper is organized as follows. Section 2 reviews background and related work. Section 3 presents the proposed framework of Passive-Aggressive Active-learning (PAA) algorithms for online binary classification tasks and analyzes the mistake bounds of the proposed algorithms. Section 4 extends the proposed framework for online multi-class classification by presenting a family of multi-class PAA algorithms (MPAA). Section 5 extends the proposed learning framework to tackle cost-sensitive classification by presenting cost-sensitive PAA (CSPAA) algorithms. Section 6 discusses our empirical studies and the applications of our technique to large-scale real-world online learning tasks, and finally Sect. 7 concludes this work.

2 Related work

Our work is closely related to three major categories of machine learning studies: online learning, online active learning and cost-sensitive classification. Below we briefly review some representative related work in each category.

2.1 Online learning

Online learning represents a family of efficient and scalable machine learning algorithms (Hoi et al. 2014; Rosenblatt 1958; Crammer and Singer 2003; Cesa-Bianchi et al. 2004; Crammer et al. 2006; Zhao et al. 2011; Wang et al. 2012). Unlike conventional batch learning methods that assume all training instances are available prior to the learning task, online learning repeatedly updates the predictive models sequentially, which is more appropriate for large-scale applications where training data often arrive sequentially. In literature, a variety of online learning methods have been proposed in machine learning. A classical online learning method is the Perceptron algorithm (Rosenblatt 1958; Freund and Schapire 1999), which updates the model by adding a new example with some constant weight into the current set of support vectors when the example is misclassified. Recently a lot of new online learning algorithms have been developed based on the criterion of maximum margin (Crammer and Singer 2003; Gentile 2001; Kivinen et al. 2001; Crammer et al. 2006; Li and Long 1999). One notable technique in this category is the online Passive-Aggressive (PA) learning method (Crammer et al. 2006), which updates the classification function when a new example is misclassified or its classification score does not exceed some predefined margin. PA algorithms have been proved as a very successful and popular online learning technique for solving many real-world applications. Finally, we note that there are also a number of emerging online learning algorithms proposed recently, such as second-order online learning (Crammer et al. 2008, 2009; Wang et al. 2012), which make more accurate predictions and often converge faster than first-order algorithms.

Most of the above existing online learning methods generally belong to supervised and passive learning. One major weakness of such supervised passive online learning methodology (Hoi et al. 2014) is the unrealistic assumption in that it assumes class labels of every incoming instance will be already requested or made available at the end of every iteration, which limits the application of online learning techniques for many real-world online learning tasks where class labels may not be always available or may be expensive to collect or request.

2.2 Online active learning

Online Active Learning algorithms emerge to address the main problem of conventional supervised online learning approach, i.e. the strong dependence on labeled data. The basic process of online active learning works in iterations. At each iteration, one unlabelled instance is presented to the learner, and the learner needs to decide whether to query its label. If the label is queried, then the learner can use the labelled instance to update the model, otherwise the model is kept unchanged.

Specifically, there are two kinds of settings for online active learning, selective sampling setting (Cavallanti et al. 2009; Cesa-Bianchi et al. 2009; Dekel et al. 2010; Orabona and Cesa-Bianchi 2011) and label efficient learning setting. We summarize their differences in several aspects. Firstly, in the selective sampling setting the instances are drawn randomly from a fixed distribution, while in the label efficient setting the instances can be generated adversarially. Secondly, the label efficient model must make predictions on those instance where the label is not requested, while the selective sampling models are concerned with the generalization error rather than the performance of the algorithm on the sequence of instances. Our work belongs to the second category. One of the most representative existing work in label efficient learning setting is the Label Efficient Perceptron algorithm, where the probability of querying the label is decided by the classification confidence. Following the similar setting, many variants of this algorithm were proposed including Adaptive Label Efficient Perceptron, Label Efficient Second-Order Perceptron (Cesa-Bianchi et al. 2006), Adaptive Label Efficient Second-Order Perceptron and Label Efficient Winnow (Cesa-Bianchi et al. 2006). Although extensively studied, the existing active learning algorithms still suffer from a serious limitation: the effort of querying a correctly classified instance is wasted due to the adoption of Perceptron update strategy.

In this work, we apply the popular PA algorithm to solve the online active learning task. Our work enjoys two advantages. On one hand, different from the regular PA setting which assumes every class label will be revealed, our approach queries the class labels of only a limited amount of incoming instances through active learning. On the other hand, our effective updating strategy fully exploits the potential of every queried instance and thus achieves a superior performance compared to existing active learning algorithms.

In addition, it is important to note that our work in online active learning is highly related to but different from active learning in data streams (Zhu et al. 2007, 2010; Žliobaitė et al. 2011). Both of them attempt to achieve the highest prediction accuracy while querying the fewest instance labels in the situation of sequentially arriving data. However, online active learning algorithms focus on the updating strategy when one new instance arrives and may discard past instances for efficient learning. While active learning in data streams must preserve the data distribution to detect the potential concept drift (Zliobaite et al. 2014).

2.3 Cost-sensitive classification

Cost-sensitive classification has been extensively studied in data mining and machine learning (Liu and Zhou 2006; Zadrozny et al. 2003; Zhu and Wu 2006). To address this problem, researchers have proposed a variety of cost-sensitive metrics. The well-known examples include the weighted sum of sensitivity and specificity (Jiawei and Kamber 2001; Brodersen et al. 2010), and the weighted misclassification cost that takes cost into consideration when measuring classification performance (Elkan 2001; Akbani et al. 2004). As a special case, when the weights are both equal to 0.5, the weighted sum of sensitivity and specificity is reduced to the well-known balanced accuracy (Brodersen et al. 2010). Although lots of algorithms have been proposed, most of them are in batch setting.

Recently a few existing works have attempted to address online cost-sensitive classification problems. Perceptron Algorithms with Uneven Margin (PAUM) (Li et al. 2002) is an extension of the Perceptron with Margin (PAM) algorithm (Krauth and Mézard 1987) where the classifier is updated whenever the classification margin is smaller than a predefined threshold. The PAUM algorithm achieves a cost-sensitive update by setting different thresholds for different class labels. Cost-Sensitive Passive-Aggressive (CPA) (Crammer et al. 2006) is proposed as a variant of PA algorithms, where the loss function is the sum of a margin based term and a constant depending on the mistake type. Although both of the above algorithms can achieve cost-sensitive updating, one main drawback is that they are not designed to optimize a cost-sensitive measurement directly. Recently some algorithms such as Cost-Sensitive Online Learning (Wang et al. 2014) and Online AUC maximization (Zhao et al. 2011) are proposed to address this drawback by directly solving an optimization problem that optimizes the target cost-sensitive measurements. Our work follows the idea of cost-sensitive learning but extends it to the active learning setting, which enjoys a great advantage in saving the labor of labeling huge amount of instances. Finally, we note that this journal article is based the extensions of our previous conference papers (Lu et al. 2014; Zhao and Hoi 2013).

3 Passive-Aggressive Active learning

3.1 Problem formulation and background review

We first introduce the problem setting of a regular online binary classification task. Let $\{(\mathbf{x}_t, y_t)|\ t = 1, \ldots, T\}$ be a sequence of input patterns for online learning, where each instance $\mathbf{x}_t \in \mathbb{R}^d$ received at the tth trial is a vector of d dimension and $y_t \in \{-1, +1\}$ is its true class label. The goal of online binary classification is to learn a linear classifier $\hat{y}_t = \text{sign}(\mathbf{w}_t \cdot \mathbf{x}_t)$ where $\mathbf{w}_t \in \mathbb{R}^d$ is the weight vector at the tth round.

For the Perceptron algorithm (Rosenblatt 1958; Freund and Schapire 1999), a learner first receives an incoming instance $\mathbf{x}_t$ at tth round; it then makes a prediction based on the current classifier $\mathbf{w}_t$; finally the true class label y_t is disclosed. If the prediction is correct, i.e., $\hat{y}_t = y_t$, no update is applied to the learner; otherwise, Perceptron updates the model with the misclassified instance $(\mathbf{x}_t, y_t)$:

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + y_t \mathbf{x}_t$$

Unlike Perceptron that updates the model only when a misclassification occurs, the Passive-Aggressive (PA) algorithms (Crammer et al. 2006) make update whenever the loss function $\ell(\mathbf{w}_t; (\mathbf{x}_t, y_t))$ is nonzero, e.g., one can choose the hinge loss $\ell(\mathbf{w}_t; (\mathbf{x}_t, y_t)) = \max(0, 1 - y_t \mathbf{w}_t \cdot \mathbf{x}_t)$. In particular, PA algorithms update the model $\mathbf{w}_{t+1}$ by solving three variants of the optimization task:

$$\arg\min_{\mathbf{w}} F(\mathbf{w}) = \begin{cases} \frac{1}{2}\|\mathbf{w} - \mathbf{w}_t\|^2 \text{ s.t. } \ell_t(\mathbf{w}; (\mathbf{x}_t, y_t)) = 0, & \text{(PA)} \\ \frac{1}{2}\|\mathbf{w} - \mathbf{w}_t\|^2 + C\ell_t(\mathbf{w}; (\mathbf{x}_t, y_t)), & \text{(PA-I)} \\ \frac{1}{2}\|\mathbf{w} - \mathbf{w}_t\|^2 + C\ell_t(\mathbf{w}; (\mathbf{x}_t, y_t))^2, & \text{(PA-II)} \end{cases}$$

where $C > 0$ is a penalty cost parameter. The closed-form solutions can be derived for the above optimizations, i.e., $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + \tau_t y_t \mathbf{x}_t$, where the stepsize τ_t is computed respectively as follows:

$$\tau_t = \begin{cases} \ell_t(\mathbf{w}_t; (\mathbf{x}_t, y_t))/\|\mathbf{x}_t\|^2, & \text{(PA)} \\ \min(C, \ell_t(\mathbf{w}_t; (\mathbf{x}_t, y_t))/\|\mathbf{x}_t\|^2), & \text{(PA-I)} \\ \ell_t(\mathbf{w}_t; (\mathbf{x}_t, y_t))/(\|\mathbf{x}_t\|^2 + 1/(2C)). & \text{(PA-II)} \end{cases} \tag{1}$$

Thus, PA algorithms are more aggressive in updating models than Perceptron.

3.2 Passive-Aggressive Active learning algorithms

In this section, we aim to develop new algorithms for online active learning. Unlike conventional online learning (Rosenblatt 1958) and pool-based active learning (McCallum and Nigam 1998; Tong and Koller 2002), the key challenges to an online active learning task are two-fold: (i) when a learner should query the class label of an incoming instance, and (ii) when the class label is queried and disclosed, how to exploit the labeled instance to update the learner in an effective way. We propose Passive-Aggressive Active (PAA) learning to tackle the above challenges. In particular, the PAA algorithms adopt a simple yet effective randomized rule to decide whether the label of an incoming instance should be queried, and employ state-of-the-art PA algorithms to exploit the labeled instance for updating the online learner.

In particular, for an incoming instance $\mathbf{x}_t$ at the tth round, the PAA algorithm first computes its prediction margin, i.e.,

$$p_t = \mathbf{w}_t \cdot \mathbf{x}_t,$$

by the current classifier, and then decides if the class label should be queried according to a Bernoulli random variable $Z_t \in \{0, 1\}$ with probability equal to

$$\delta/(\delta + |p_t|),$$

where $\delta \geq 1$ is a smoothing parameter. Such an approach is similar to the idea of margin-based active learning (Tong and Koller 2002; Balcan et al. 2007) and has been adopted in other previous work (Cesa-Bianchi et al. 2006; Dasgupta et al. 2009). If the outcome $Z_t = 0$, the class label will not be queried and the learner is not updated; otherwise, the class label is queried and the outcome y_t is disclosed. Whenever the class label of an incoming instance is queried, the PAA algorithms will try the best effort to exploit the potential of this example for updating the learner. Specifically, it adopts the PA algorithms to update the linear classification model $\mathbf{w}_{t+1}$ according to Eq. (1). Based on the different updating strategies, we name the hard margin algorithm as PAA and the two soft margin algorithms as PAA-I and PAA-II. Clearly this is able to overcome the limitation of the Perceptron-based active learning algorithm that only updates the misclassified instances and wastes a large amount of correctly classified instances with low prediction confidence which can be potentially beneficial to improving the classifier. Finally, we summarize the detailed steps of the proposed PAA algorithms in Algorithm 2.

Algorithm 2 Passive-Aggressive Active learning algorithms (**PAA**)

```
INPUT : penalty parameter $C > 0$ and smoothing parameter $\delta \geq 1$.
INITIALIZATION : $\mathbf{w}_1 = (0, \ldots, 0)^\top$.
for $t = 1, \ldots, T$ do
  observe: $\mathbf{x}_t \in \mathbb{R}^d$, set $p_t = \mathbf{w}_t \cdot \mathbf{x}_t$, and predict $\hat{y}_t = \text{sign}(p_t)$;
  draw a Bernoulli random variable $Z_t \in \{0, 1\}$ of parameter $\delta/(\delta + |p_t|)$;
  if $Z_t = 1$ then
    query label $y_t \in \{-1, +1\}$, and suffer loss $\ell_t(\mathbf{w}_t) = \max(0, 1 - y_t \mathbf{w}_t \cdot \mathbf{x}_t)$;
    compute $\tau_t$ according to Eq. (1), and $\mathbf{w}_{t+1} = \mathbf{w}_t + \tau_t y_t \mathbf{x}_t$;
  else
    $\mathbf{w}_{t+1} = \mathbf{w}_t$;
  end if
end for
```

3.3 Analysis of mistake bounds for the PAA algorithms

In this section, we aim to theoretically analyze the mistake bounds of the proposed PAA algorithms. Before presenting the mistake bounds, we begin by presenting a technical lemma which would facilitate the proofs in this section. With this lemma, we could then derive the loss and mistake bounds for the three variants of PAA algorithm. For convenience, we introduce the following notation: $\mathcal{M} = \{t | t \in [T], \hat{y}_t \neq y_t\}$, and $\mathcal{L} = \{t | t \in [T], \hat{y}_t = y_t, \ell_t(\mathbf{w}_t; (\mathbf{x}_t, y_t)) > 0\}$, where $[T]$ denotes $\{1, 2, \ldots, T\}$.

Lemma 1 *Let $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_T, y_T)$ be a sequence of input instances, where $\mathbf{x}_t \in \mathbb{R}^d$ and $y_t \in \{-1, +1\}$ for all t. Let τ_t be the stepsize parameter for either of the three PAA variants as given in Eq. (1). Then, the following bound holds for any $\mathbf{w} \in \mathbb{R}^d$ and any $\alpha > 0$*

$$\sum_{t=1}^{T} Z_t 2\tau_t \big[L_t(\alpha - |p_t|) + M_t(\alpha + |p_t|)\big] \leq \alpha^2 \|\mathbf{w}\|^2 + \sum_{t=1}^{T} \tau_t^2 \|\mathbf{x}_t\|^2 + \sum_{t=1}^{T} 2\alpha\tau_t \ell_t(\mathbf{w}),$$

where $M_t = \mathbb{I}_{(t \in \mathcal{M})}$, $L_t = \mathbb{I}_{(t \in \mathcal{L})}$, $\mathbb{I}$ is the indicator function.

The detailed proof of Lemma 1 can be found in "Appendix 1".

Based on Lemma 1, we first derive the expected mistake bound for the PAA algorithm in the separable case. We assume there exists some $\mathbf{w}$ such that $y_t(\mathbf{w} \cdot \mathbf{x}_t) \geq 1, \forall t \in [T]$.

Theorem 1 *Let $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_T, y_T)$ be a sequence of input instances, where $\mathbf{x}_t \in \mathbb{R}^d$ and $y_t \in \{-1, +1\}$ and $\|\mathbf{x}_t\| \leq R$ for all t. For any vector $\mathbf{w}$ that satisfies $\ell_t(\mathbf{w}) = 0$ for all t, assuming $\delta \geq 1$, the expected number of mistakes made by the PAA algorithm on this sequence of examples is bounded by*

$$\mathbb{E}\left[\sum_{t=1}^{T} M_t\right] \leq \mathbb{E}\left[\sum_{t=1}^{T} M_t \ell_t(\mathbf{w}_t)\right] \leq \frac{R^2}{4}\left(\delta + \frac{1}{\delta} + 2\right)\|\mathbf{w}\|^2.$$

By setting $\delta = 1$*, we can obtain the best upper bound as follows:*

$$\mathbb{E}\left[\sum_{t=1}^{T} M_t\right] \leq \mathbb{E}\left[\sum_{t=1}^{T} M_t \ell_t(\mathbf{w}_t))\right] \leq R^2\|\mathbf{w}\|^2.$$

Proof Since $\ell_t(\mathbf{w}) = 0, \forall t \in [T]$, according to Lemma 1, we have

$$\sum_{t=1}^{T} Z_t 2\tau_t \big[L_t(\alpha - |p_t|) + M_t(\alpha + |p_t|)\big] \leq \alpha^2\|\mathbf{w}\|^2 + \sum_{t=1}^{T} \tau_t^2 \|\mathbf{x}_t\|^2.$$

Further, the above inequality can be reformulated as:

$$\begin{aligned}
\alpha^2\|\mathbf{w}\|^2 &\geq \sum_{t=1}^{T} Z_t 2\tau_t \left[L_t\left(\alpha - |p_t|\right) + M_t\left(\alpha + |p_t|\right)\right] - \sum_{t=1}^{T} \tau_t^2\|\mathbf{x}_t\|^2 \\
&= \sum_{t=1}^{T} Z_t 2\tau_t \left[L_t\left(\alpha - |p_t| - \frac{\tau_t}{2}\|\mathbf{x}_t\|^2\right) + M_t\left(\alpha + |p_t| - \frac{\tau_t}{2}\|\mathbf{x}_t\|^2\right)\right] \\
&= \sum_{t=1}^{T} Z_t 2\tau_t \left[L_t\left(\alpha - |p_t| - \frac{\ell_t(\mathbf{w}_t)}{2}\right) + M_t\left(\alpha + |p_t| - \frac{\ell_t(\mathbf{w}_t)}{2}\right)\right] \\
&= \sum_{t=1}^{T} Z_t 2\tau_t \left[L_t\left(\alpha - |p_t| - \frac{1 - y_t p_t}{2}\right) + M_t\left(\alpha + |p_t| - \frac{1 - y_t p_t}{2}\right)\right] \\
&= \sum_{t=1}^{T} Z_t 2\tau_t \left[L_t\left(\alpha - |p_t| - \frac{1 - |p_t|}{2}\right) + M_t\left(\alpha + |p_t| - \frac{1 + |p_t|}{2}\right)\right] \\
&= \sum_{t=1}^{T} L_t Z_t 2\tau_t\left(\alpha - \frac{1 + |p_t|}{2}\right) + \sum_{t=1}^{T} M_t Z_t 2\tau_t\left(\alpha - \frac{1 - |p_t|}{2}\right).
\end{aligned}$$

Plugging $\alpha = \frac{\delta+1}{2}, \delta \geq 1$ into the above inequality results in

$$\left(\frac{1+\delta}{2}\right)^2 \|\mathbf{w}\|^2 \geq \sum_{t=1}^{T} M_t Z_t \tau_t(\delta + |p_t|),$$

since when $L_t = 1, |p_t| \in [0, 1), (\alpha - \frac{1+|p_t|}{2}) = \frac{\delta - |p_t|}{2} > 0$, and $(\alpha - \frac{1-|p_t|}{2}) = \frac{\delta + |p_t|}{2}$.

In addition, combining the fact $\tau_t = \ell_t(\mathbf{w}_t)/\|\mathbf{x}_t\|^2 \geq \ell_t(\mathbf{w}_t)/R^2$ with the above inequality concludes:

$$\left(\frac{1+\delta}{2}\right)^2 \|\mathbf{w}\|^2 \geq \frac{1}{R^2}\sum_{t=1}^{T} M_t Z_t \ell_t(\mathbf{w}_t)(\delta + |p_t|).$$

Taking expectation with the above inequality results in

$$\frac{1}{R^2}\mathbb{E}\left[\delta\sum_{t=1}^{T}M_t\ell_t(\mathbf{w}_t)\right] = \mathbb{E}\left[\frac{1}{R^2}\sum_{t=1}^{T}M_t\ell_t(\mathbf{w}_t)(\delta+|p_t|)\mathbb{E}Z_t\right]$$
$$= \mathbb{E}\left[\frac{1}{R^2}\sum_{t=1}^{T}M_t\ell_t(\mathbf{w}_t)Z_t(\delta+|p_t|)\right] \le \left(\frac{1+\delta}{2}\right)^2\|\mathbf{w}\|^2.$$

□

Remark Since, the above theorem holds for any $\mathbf{w}$ such that $\forall t$, $\ell_t(\mathbf{w}) = 0$, we obtain the tightest bound when $\mathbf{w} = \mathbf{w}^*$, where

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} ||\mathbf{w}||_2^2,$$
$$s.t. \min_{t\in[T]} y_t\mathbf{w}^\top\mathbf{x}_t \ge 1.$$

We can draw two observations as follows. When the norm of instances diverse greatly, i.e. $\min_t ||\mathbf{x}_t|| \ll R$, to guarantee the zero loss of the smallest $\mathbf{x}_t$, the optimal $||\mathbf{w}^*||_2^2$ should be extremely large, which indicates a loose theoretical bound. Thus a proper data pre-processing scheme, for example scaling all instance vectors to the same norm, will help improve performance. However, adopting two different data scaling scheme, for example, scaling $||\mathbf{x}_t|| = R$ and $||\mathbf{x}_t|| = cR$, $c \in \mathbb{R}^+$, will not affect the performance. That is because when changing $\mathbf{x}_t$ to $c\mathbf{x}_t$ (R to cR), $\mathbf{w}^*$ is changed to $\mathbf{w}^*/c$, which makes no change to the theorem.

The above mistake bound indicates that the expected number of mistakes is proportional to the upper bound of the instances norm R and inversely proportional to the margin $1/\|\mathbf{w}\|^2$, which is consistent with existing research (Crammer et al. 2006). One disadvantage of the above theorem is the linear separable assumption, since real world datasets are usually not separable. To solve this problem, we present the expected mistake bound for the PAA-I algorithm, which is suitable for the non-separable problem.

Theorem 2 *Let $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_T, y_T)$ be a sequence of examples where $\mathbf{x}_t \in \mathbb{R}^d$ and $y_t \in \{-1, +1\}$ and $\|\mathbf{x}_t\| \le R$ for all t. Assuming $\delta \ge 1$, for any vector $\mathbf{w} \in \mathbb{R}^d$, the expected number of prediction mistakes made by PAA-I on this sequence of examples is bounded from above by*

$$\mathbb{E}\left[\sum_{t=1}^{T}M_t\right] \le \beta\left\{\left(\frac{\delta+1}{2}\right)^2\|\mathbf{w}\|^2 + (\delta+1)C\sum_{t=1}^{T}\ell_t(\mathbf{w})\right\},$$

where $\beta = \frac{1}{\delta}\max\{\frac{1}{C}, R^2\}$ and C is the aggressiveness parameter for PAA-I. Setting $\delta = 1$ leads to the following bound

$$\mathbb{E}\left[\sum_{t=1}^{T}M_t\right] \le \max\left\{\frac{1}{C}, R^2\right\}\left\{\|\mathbf{w}\|^2 + 2C\sum_{t=1}^{T}\ell_t(\mathbf{w})\right\}.$$

Setting $\delta = \sqrt{1 + \frac{4C\sum_{t=1}^{T}\ell_t(\mathbf{w})}{\|\mathbf{w}\|^2}}$ *leads to the following bound*

$$\mathbb{E}\left[\sum_{t=1}^{T} M_t\right]$$
$$\leq \max\left\{\frac{1}{C}, R^2\right\}\left\{\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{t=1}^{T}\ell_t(\mathbf{w}) + \frac{1}{2}\|\mathbf{w}\|\sqrt{\|\mathbf{w}\|^2 + 4C\sum_{t=1}^{T}\ell_t(\mathbf{w})}\right\}.$$

Proof According to Lemma 1, we have

$$\alpha^2\|\mathbf{w}\|^2 + \sum_{t=1}^{T} 2\alpha\tau_t\ell_t(\mathbf{w})$$
$$\geq \sum_{t=1}^{T} Z_t 2\tau_t\left[L_t\left(\alpha - |p_t| - \frac{\tau_t}{2}\|\mathbf{x}_t\|^2\right) + M_t\left(\alpha + |p_t| - \frac{\tau_t}{2}\|\mathbf{x}_t\|^2\right)\right]$$
$$\geq \sum_{t=1}^{T} L_t Z_t 2\tau_t\left(\alpha - \frac{1+|p_t|}{2}\right) + \sum_{t=1}^{T} M_t Z_t 2\tau_t\left(\alpha - \frac{1-|p_t|}{2}\right).$$

Similar to Theorem 1, plugging $\alpha = \frac{\delta+1}{2}$, $\delta \geq 1$ into the above inequality will result in

$$\left(\frac{\delta+1}{2}\right)^2\|\mathbf{w}\|^2 + \sum_{t=1}^{T}(\delta+1)\tau_t\ell_t(\mathbf{w}) \geq \sum_{t=1}^{T} M_t Z_t\tau_t(\delta + |p_t|).$$

Since $\tau_t \geq \min\{C, \frac{1}{R^2}\}$ holds when $M_t = 1$, the above inequality implies:

$$\left(\frac{\delta+1}{2}\right)^2\|\mathbf{w}\|^2 + \sum_{t=1}^{T}(\delta+1)\tau_t\ell_t(\mathbf{w}) \geq \min\left\{C, \frac{1}{R^2}\right\}\sum_{t=1}^{T} M_t Z_t(\delta + |p_t|).$$

Taking expectation with the above equality and re-arranging the result conclude the theorem. □

This theorem shows that the number of expected mistakes is bounded by a weighted sum of the model complexity $\|\mathbf{w}\|^2$ and the cumulative loss $\sum_{t=1}^{T}\ell_t(\mathbf{w})$ suffered by it. Finally, we give the mistake bound of the PAA-II algorithm in the following theorem.

Theorem 3 *Let* $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_T, y_T)$ *be a sequence of examples where* $\mathbf{x}_t \in \mathbb{R}^d$ *and* $y_t \in \{-1, +1\}$ *and* $\|\mathbf{x}_t\| \leq R$ *for all* t*. Then, for any vector* $\mathbf{w} \in \mathbb{R}^d$*, assuming* $\delta \geq 1$*, the expected number of prediction mistakes made by PAA-II on this sequence of examples is bounded from above by,*

$$\mathbb{E}\left[\sum_{t=1}^{T} M_t\right] \leq \left\{R^2 + \frac{1}{2C}\right\}\frac{1}{\delta}\left\{\left(\frac{\delta+1}{2}\right)^2\|\mathbf{w}\|^2 + 2C\left(\frac{\delta+1}{2}\right)^2\sum_{t=1}^{T}\ell_t(\mathbf{w})^2\right\},$$

where C *is the aggressiveness parameter for PAA-II. By setting* $\delta = 1$*, we can further have*

$$\mathbb{E}\left[\sum_{t=1}^{T} M_t\right] \leq \left\{R^2 + \frac{1}{2C}\right\}\left\{(\|\mathbf{w}\|^2 + 2C\sum_{t=1}^{T}\ell_t(\mathbf{w})^2\right\}.$$

Proof Define

$$\mathcal{O} = \alpha^2\|\mathbf{w}\|^2 + \sum_{t=1}^{T}\tau_t^2\|\mathbf{x}_t\|^2 + \sum_{t=1}^{T}2\alpha\tau_t\ell_t(\mathbf{w}),$$

$$\mathcal{P} = \sum_{t=1}^{T}\alpha\left(\frac{\tau_t}{\sqrt{2C\alpha}} - \sqrt{2C\alpha}\ell_t(\mathbf{w})\right)^2,$$

$$\mathcal{Q} = \alpha^2\|\mathbf{w}\|^2 + \sum_{t=1}^{T}\tau_t^2\left(\|\mathbf{x}_t\|^2 + \frac{1}{2C}\right) + \sum_{t=1}^{T}2C\alpha^2\ell_t(\mathbf{w})^2,$$

then it is easy to verify that $\mathcal{O} \leq \mathcal{O} + \mathcal{P} = \mathcal{Q}$.

Combining $\mathcal{O} \leq \mathcal{Q}$ with Lemma 1, we have the following

$$\sum_{t=1}^{T}(L_tZ_t2\tau_t(\alpha - |p_t|) + M_tZ_t2\tau_t(\alpha + |p_t|) \leq \mathcal{Q}.$$

Furthermore, the above formulation can be reformulated as:

$$\begin{aligned}
&\alpha^2\|\mathbf{w}\|^2 + \sum_{t=1}^{T}2C\alpha^2\ell_t(\mathbf{w})^2 \\
&\geq \sum_{t=1}^{T}Z_t2\tau_t\Big[L_t(\alpha - |p_t|) + M_t(\alpha + |p_t|)\Big] - \tau_t^2\left(\|\mathbf{x}_t\|^2 + \frac{1}{2C}\right) \\
&= \sum_{t=1}^{T}L_tZ_t2\tau_t\left(\alpha - \frac{1+|p_t|}{2}\right) + \sum_{t=1}^{T}M_tZ_t2\tau_t\left(\alpha - \frac{1-|p_t|}{2}\right).
\end{aligned}$$

Similar to Theorem 1, plugging $\alpha = \frac{\delta+1}{2}$, $\delta \geq 1$ into the above inequality results in

$$\left(\frac{\delta+1}{2}\right)^2\|\mathbf{w}\|^2 + \sum_{t=1}^{T}2C\left(\frac{\delta+1}{2}\right)^2\ell_t(\mathbf{w})^2 > \sum_{t=1}^{T}M_tZ_t\tau_t(\delta + |p_t|).$$

Taking expectation with the above inequality and using $\tau_t \geq 1/\{R^2 + \frac{1}{2C}\}$, will conclude the theorem. □

Remark As proven in previous work (Cesa-Bianchi et al. 2006), the expected mistake bounds for active learning perceptron, which in our notation, could be expressed as:

$$\mathbb{E}\left[\sum_{t=1}^{T}M_t\right] \leq \frac{(2\delta + R^2)^2}{8\delta}\|\mathbf{w}\|^2 + \left(1 + \frac{R^2}{2\delta}\right)\sum_{t=1}^{T}\ell_t(\mathbf{w}).$$

By setting $\delta = 1$, they further have

$$\mathbb{E}\left[\sum_{t=1}^{T}M_t\right] \leq \frac{(2 + R^2)^2}{8}\|\mathbf{w}\|^2 + \left(1 + \frac{R^2}{2}\right)\sum_{t=1}^{T}\ell_t(\mathbf{w}).$$

We could find that generally speaking, the bounds are similar and it depends on the parameters to determine which is better. This is similar to the comparison between the PA bound (Cesa-Bianchi and Lugosi 2006; Crammer et al. 2006) and Perceptron bound (Freund and Schapire

1999). However, the bound for Percetron Based Active learning has a $R^4\|w\|^2$ order term, which may make it inferior to ours.

One problem in the above theorem is that the value of δ must be larger than 1, which may result in too many requests. To fix this issue, we propose the following theorem that can resolve this limitation.

Theorem 4 *Let $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_T, y_T)$ be a sequence of examples where $\mathbf{x}_t \in \mathbb{R}^d$ and $y_t \in \{-1, +1\}$ and $\|\mathbf{x}_t\| \leq R$ for all t. Assume there exists a vector $\mathbf{w}$ such that $\ell_t(\mathbf{w}) = 0$ for all t. For the PAA algorithm, if change the parameter for the Bernoulli distribution to $\delta/(\delta + 1 + |p_t|)$ and $\delta \geq 0$, then its expected number of prediction mistakes on this sequence is bounded by*

$$\mathbb{E}\left[\sum_{t=1}^{T} M_t)\right] \leq \mathbb{E}\left[\sum_{t=1}^{T} M_t \ell_t(\mathbf{w}_t)\right] \leq R^2\left(\frac{\delta}{4} + \frac{1}{\delta} + 1\right)\|\mathbf{w}\|^2.$$

When setting $\delta = 2$, we get the best upper bound

$$\mathbb{E}\left[\sum_{t=1}^{T} M_t\right] \leq \mathbb{E}\left[\sum_{t=1}^{T} M_t \ell_t(\mathbf{w}_t))\right] \leq 2R^2\|\mathbf{w}\|^2.$$

Proof As proven in Theorem 1,

$$\alpha^2\|\mathbf{w}\|^2 + \sum_{t=1}^{T} 2\alpha\tau_t\ell_t(\mathbf{w}) \geq \sum_{t=1}^{T} Z_t 2\tau_t\left[L_t\left(\alpha - \frac{1+|p_t|}{2}\right) + M_t\left(\alpha - \frac{1-|p_t|}{2}\right)\right].$$

Plugging $\alpha = \frac{\delta}{2} + 1$, $\delta \geq 0$ into the above inequality results in

$$\left(\frac{\delta}{2} + 1\right)^2\|\mathbf{w}\|^2 > \sum_{t=1}^{T} M_t Z_t \tau_t(\delta + 1 + |p_t|),$$

since, when $L_t = 1$, $|p_t| \in [0, 1)$, $(\alpha - \frac{1+|p_t|}{2}) = \frac{\delta+1-|p_t|}{2} > 0$, and $(\alpha - \frac{1-|p_t|}{2}) = \frac{\delta+1+|p_t|}{2}$. Taking expectation with the above inequality and using $\tau_t \geq \ell_t(\mathbf{w}_t)/R^2$ will conclude the theorem. □

Remark Note this theorem demonstrates that for a new sampling probability $\frac{\delta}{\delta+1+|p_t|}$, the expected number of mistakes of the PAA algorithm is bounded. Similar property also holds for PAA-I and PAA-II. It's easy to get the corresponding bounds using the facts that $\tau_t \geq \min\{C, \frac{1}{R^2}\}$ for PAA-I and $\tau_t \geq 1/\{R^2 + \frac{1}{2C}\}$ for PAA-II when $M_t = 1$. We omit the theorems since it is similar to Theorem 4.

4 Extension to multi-class online classification

In this section, we will generalize the PAA algorithms to solve online multi-class classification tasks.

4.1 Problem formulation and background review

We first introduce the problem setting of the multi-class classification problem. Let $\{(\mathbf{x}_t, y_t)|\ t = 1, \ldots, T\}$ be a sequence of input patterns for online learning, where each instance $\mathbf{x}_t \in \mathbb{R}^d$ received at the tth trial is a vector of d dimension and $y_t \in Y = \{1, \ldots, k\}$ is its true class label. We adopt the multi-prototype model in Crammer et al. (2006). The classifier is made up of k weight vectors $\mathbf{w}^r \in \mathbb{R}^d$, $r \in Y$, where each vector corresponds to one class label. During the prediction period of the tth iteration, the classifier first generates a sequence of k prediction scores for all the class labels:

$$(\mathbf{w}_t^1 \cdot \mathbf{x}_t, \ldots, \mathbf{w}_t^r \cdot \mathbf{x}_t, \ldots, \mathbf{w}_t^k \cdot \mathbf{x}_t).$$

Then, by comparing the above scores, it picks the class label with the largest score as the prediction,

$$\hat{y}_t = \arg\max_{r \in Y} \mathbf{w}_t^r \cdot \mathbf{x}_t. \tag{2}$$

We further define s_t as the irrelevant class with the highest prediction score:

$$s_t = \arg\max_{r \in Y, r \neq y_t} \mathbf{w}_t^r \cdot \mathbf{x}_t.$$

The margin with respect to the hypothesis in the tth iteration is defined to be the gap between the prediction score of class y_t and s_t:

$$\gamma_t = \mathbf{w}_t^{y_t} \cdot \mathbf{x}_t - \mathbf{w}_t^{s_t} \cdot \mathbf{x}_t.$$

Obviously, in a correct prediction, the margin $\gamma_t > 0$. In the max-score multi-class Perceptron algorithm (Crammer and Singer 2003), the classifier is only updated when a prediction mistake occurs, i.e. $\hat{y}_t \neq y_t$; otherwise, Perceptron updates the model with the misclassified instance $(\mathbf{x}_t, y_t)$:

$$\mathbf{w}_{t+1}^{y_t} \leftarrow \mathbf{w}_t^{y_t} + \mathbf{x}_t,$$
$$\mathbf{w}_{t+1}^{s_t} \leftarrow \mathbf{w}_t^{s_t} - \mathbf{x}_t.$$

Unlike Perceptron that updates the model only when a misclassification occurs, the Multi-class Passive-Aggressive (MPA) algorithms (Crammer et al. 2006) will also updates the classifier when the prediction is correct while the margin is not large enough. Specifically, MPA algorithms will update the model when the hinge loss is nonzero, where the hinge loss is defined as,

$$\ell_t(\overline{\mathbf{w}}_t) = \max(0, 1 - \gamma_t),$$

in which $\overline{\mathbf{w}}_t$ denotes the set of all k weight vectors in the classifier.

If the hinge loss is positive, then multi-class PA algorithms will update the model $\overline{\mathbf{w}}_{t+1}$ by solving three variants of the following optimization objectives:

$$\arg\min_{\overline{\mathbf{w}}} F(\overline{\mathbf{w}}) = \begin{cases} \frac{1}{2}\sum_{r=1}^k \|\mathbf{w}^r - \mathbf{w}_t^r\|^2 \text{ s.t. } \ell_t(\overline{\mathbf{w}}; (\mathbf{x}_t, y_t)) = 0, & \text{(MPA)} \\ \frac{1}{2}\sum_{r=1}^k \|\mathbf{w}^r - \mathbf{w}_t^r\|^2 + C\ell_t(\overline{\mathbf{w}}; (\mathbf{x}_t, y_t)), & \text{(MPA-I)} \\ \frac{1}{2}\sum_{r=1}^k \|\mathbf{w}^r - \mathbf{w}_t^r\|^2 + C\ell_t(\overline{\mathbf{w}}; (\mathbf{x}_t, y_t))^2, & \text{(MPA-II)} \end{cases}$$

where $C > 0$ is a penalty cost parameter. Luckily, the above optimizations enjoy closed-form solutions as follows,

$$\begin{aligned} \mathbf{w}_{t+1}^{y_t} &\leftarrow \mathbf{w}_t^{y_t} + \tau_t \mathbf{x}_t, \\ \mathbf{w}_{t+1}^{s_t} &\leftarrow \mathbf{w}_t^{s_t} - \tau_t \mathbf{x}_t, \end{aligned} \tag{3}$$

where the stepsize τ_t is computed respectively as follows:

$$\tau_t = \begin{cases} \ell_t(\overline{\mathbf{w}}_t; (\mathbf{x}_t, y_t))/(2\|\mathbf{x}_t\|^2), & \text{(MPA)} \\ \min(C, \ell_t(\overline{\mathbf{w}}_t; (\mathbf{x}_t, y_t))/(2\|\mathbf{x}_t\|^2)), & \text{(MPA-I)} \\ \ell_t(\overline{\mathbf{w}}_t; (\mathbf{x}_t, y_t))/(2\|\mathbf{x}_t\|^2 + 1/(2C)). & \text{(MPA-II)} \end{cases} \tag{4}$$

These update rules generally implies that larger losses will result in larger learning rates.

4.2 Multi-class Passive-Aggressive Active learning algorithms (MPAA)

In this section, we aim to develop a group of new algorithms for online active multi-class classification tasks, termed as the Multi-class Passive-Aggressive Active learning (MPAA) algorithms. Firstly, a similar but different stochastic rule is adopted in deciding whether to query the label of a certain instance. This rule will be introduced in the later paragraphs. If a label is queried, the update rules of MPAA algorithms simply follow those of the MPA algorithms introduced in the previous section.

Now we will introduce the randomized rule for querying labels. As stated in the PAA algorithms, the probability of querying a label, $Pr(Z_t = 1)$ should be inversely proportional to the margin of the model on the current instance (with a smoothing parameter δ), which is considered as a kind of confidence of the model. However, in multi-class setting, the margin of the model on the current example is not available, since the label is not disclosed when the probability is computed.

To solve this problem, we introduce the label with second largest prediction score:

$$\tilde{y}_t = \arg\max_{r \in Y, r \neq \hat{y}_t} \mathbf{w}_t^r \cdot \mathbf{x}_t,$$

and propose a different confidence score

$$p_t = \mathbf{w}_t^{\hat{y}_t} \cdot \mathbf{x}_t - \mathbf{w}_t^{\tilde{y}_t} \cdot \mathbf{x}_t, \tag{5}$$

which is the gap between the prediction scores of predicted label $\hat{y}_t$ and the second label $\tilde{y}_t$. Note that $p_t \geq 0$ holds for all cases. The relation between the confidence value p_t and the margin γ_t has two cases: 1) if the prediction is correct, i.e., $\hat{y}_t = y_t$ and $s_t = \tilde{y}_t$, then $p_t = \gamma_t$; 2) if the prediction is incorrect, then it is easy to check $p_t \leq |\gamma_t|$. Given this confidence value, the probability of querying a label is set as

$$Pr(Z_t = 1) = \frac{\delta}{\delta + p_t},$$

where $\delta > 0$ is a smooth parameter. This probability is larger than $\frac{\delta}{\delta + |\gamma_t|}$, which will facilitate the theoretical analysis later.

Finally, we summarize the detailed steps of the proposed MPAA algorithms in Algorithm 3.

4.3 Analysis of mistake bounds for the MPAA algorithms

In this section, we aim to theoretically analyze the mistake bounds of the proposed MPAA algorithms.

Theorem 5 *Let $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_T, y_T)$ be a sequence of input instances, where $\mathbf{x}_t \in \mathbb{R}^d$ and $y_t \in Y$ and $\|\mathbf{x}_t\| \leq R$ for all t. For any classifier $\overline{\mathbf{w}}$ such that $\ell_t(\overline{\mathbf{w}}) = 0$ for all t, assuming $\delta \leq 1$ the expected number of mistakes made by the MPAA algorithm on this sequence of*

Algorithm 3 Multi-class Passive-Aggressive Active learning algorithms (**MPAA**)

```
INPUT : penalty parameter $C > 0$ and smoothing parameter $\delta \geq 1$.
INITIALIZATION : $\mathbf{w}_1^r = (0, \dots, 0)^\top$, for each $r \in Y$.
for $t = 1, \dots, T$ do
  observe: $\mathbf{x}_t \in \mathbb{R}^d$, predict $\hat{y}_t$ as Equation(2) and set $p_t$ as Equation (5).
  draw a Bernoulli random variable $Z_t \in \{0, 1\}$ of parameter $\delta/(\delta + p_t)$;
  if $Z_t = 1$ then
    query label $y_t \in \{1, \dots, k\}$, and suffer loss $\ell_t(\overline{\mathbf{w}_t}) = \max(0, 1 - \gamma_t)$;
    compute $\tau_t$ according to equation (4), and update the model as Equation (3)
  else
    $\overline{\mathbf{w}_{t+1}} = \overline{\mathbf{w}_t}$;
  end if
end for
```

examples is bounded by

$$\mathbb{E}\left[\sum_{t=1}^{T} M_t\right] \leq \mathbb{E}\left[\sum_{t=1}^{T} M_t \ell_t(\overline{\mathbf{w}}_t))\right] \leq \frac{R^2}{2}\left(\delta + \frac{1}{\delta} + 2\right)\sum_{r=1}^{k} \|\mathbf{w}^r\|^2.$$

By setting $\delta = 1$, *we can obtain the best upper bound as follows:*

$$\mathbb{E}\left[\sum_{t=1}^{T} M_t\right] \leq \mathbb{E}\left[\sum_{t=1}^{T} M_t \ell_t(\mathbf{w}_t)\right] \leq 2R^2 \sum_{r=1}^{k} \|\mathbf{w}^r\|^2.$$

The proof can be found in "Appendices 2 and 3". Similarly, we can get the mistake bounds for the other variants of MPAA algorithms. Because it is easy, we skip it for conciseness.

5 Extension to cost-sensitive online classification

In this section, we further extend the PAA algorithms to deal with highly imbalanced binary classification tasks, where instead of simply maximizing the accuracy, we should further consider some cost-sensitive evaluation metrics.

5.1 Problem formulation and cost-sensitive classification review

For binary classification, the result of each prediction for an instance can be classified into four cases: (1) $TruePositive$ (TP) if $\hat{y}_t = y_t = +1$; (2) $FalsePositive$ (FP) if $\hat{y}_t = +1$ and $y_t = -1$; (3) $TrueNegative$ (TN) if $\hat{y}_t = y_t = -1$; and (4) $FalseNegative$ (FN) if $\hat{y}_t = -1$ and $y_t = +1$. We now consider a sequence of training examples $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_T, y_T)$ for online learning. Then, we denote by $\mathcal{M}$ to be the set of indexes that correspond to the trial of misclassification:

$$\mathcal{M} = \{t | y_t \neq \text{sign}(\mathbf{w}_t \cdot \mathbf{x}_t), \forall t \in [T]\}.$$

Similarly, we denote by $\mathcal{M}_p = \{t | t \in \mathcal{M} \text{ and } y_t = +1\}$ the set of indexes for false negatives, and $\mathcal{M}_n = \{t | t \in \mathcal{M} \text{ and } y_t = -1\}$ the set of indexes for false positives. Further introduce notation $M = |\mathcal{M}|$ to denote the number of mistakes, $M_p = |\mathcal{M}_p|$ to denote the number of false negative and $M_n = |\mathcal{M}_n|$ to denote the number of false positives. Let T_p denote the number of positive instances and T_n denote the number of negative instances, we have the following performance metrics: *sensitivity* is defined as the ratio between the number of true positives $T_p - M_p$ and the number of positive examples; and *specificity* is defined as the ratio between $T_n - M_n$ and the number of negative examples. These can be summarized as:

$$sensitivity = \frac{T_p - M_p}{T_p}, \quad specificity = \frac{T_n - M_n}{T_n}.$$

Without loss of generality, we assume "positive" is the rare class, i.e., $T_p \ll T_n$.

For traditional online learning, the performance is measured by the prediction accuracy (or mistake rate equivalently) over the sequence of examples. This is inappropriate for imbalanced data because a trivial learner that simply classifies any example as negative could achieve a quite high accuracy for a highly imbalanced dataset. Thus, we propose to study new online learning algorithms, which can optimize a more appropriate performance metric, such as the *sum* of weighted *sensitivity* and *specificity*, i.e.,

$$sum = \eta_p \times sensitivity + \eta_n \times specificity, \tag{6}$$

where $0 \le \eta_p, \eta_n \le 1$ and $\eta_p + \eta_n = 1$. When $\eta_p = \eta_n = 1/2$, sum is the well-known balanced accuracy, which is adopted as a metric in the existing studies for anomaly detection (Li and Tsang 2011). In general, the higher the *sum* value, the better the performance. Besides, another suitable metric is the total cost suffered by the algorithm, which is defined as:

$$cost = c_p \times M_p + c_n \times M_n, \tag{7}$$

where M_p and M_n are the number of false negatives and false positives respectively, $0 \le c_p, c_n \le 1$ are the cost parameters for positive and negative classes, respectively, and we assume $c_p + c_n = 1$. The lower the *cost* value, the better the classification performance.

5.2 Cost-sensitive Passive-Aggressive Active learning algorithm (CSPAA)

We now propose the CSPAA framework for cost-sensitive online binary classification task by optimizing the previous two cost-sensitive measures. Before presenting our algorithms, we prove an important proposition below to motivate our solution. For simplicity, we assume $\|\mathbf{x}_t\| = 1$ for the rest.

Proposition 1 *Consider a cost-sensitive classification problem, the goal of maximizing the weighted sum in* (6) *or minimizing the weighted cost in* (7) *is equivalent to minimizing the following objective:*

$$\sum_{y_t=+1} \rho \mathbb{I}_{(y_t \mathbf{w}\cdot\mathbf{x}_t < 0)} + \sum_{y_t=-1} \mathbb{I}_{(y_t \mathbf{w}\cdot\mathbf{x}_t < 0)}, \tag{8}$$

where $\rho = \frac{\eta_p T_n}{\eta_n T_p}$ *for the maximization of the weighted sum,* T_p *and* T_n *are the number of positive examples and negative examples respectively,* $\rho = \frac{c_p}{c_n}$ *for the minimization of the weighted misclassification cost, and* $\mathbb{I}_\pi$ *is the indicator function that outputs 1 if the statement* π *holds and 0 otherwise.*

The proof can be found in the "Appendix 4".

Proposition 1 gives the explicit objective function for optimization, but the indicator function is non-convex. To tackle this issue, we replace the indicator function by its convex surrogate, i.e., a modified hinge loss function:

$$\ell(\mathbf{w}; (\mathbf{x}, y)) = \max(0, \rho * \mathbb{I}_{(y=1)} + \mathbb{I}_{(y=-1)} - y(\mathbf{w} \cdot \mathbf{x})). \tag{9}$$

As a result, we can formulate the primal objective function as follows:

$$\mathcal{F}_p^b(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{t=1}^{T}\ell_t(\mathbf{w}), \tag{10}$$

where the regularization parameter $C > 0$, the loss function $\ell_t(\mathbf{w}) = \max(0, \rho_t - y_t(\mathbf{w}\cdot\mathbf{x}_t))$ and $\rho_t = \rho * \mathbb{I}_{(y_t=1)} + \mathbb{I}_{(y_t=-1)}$. The idea of this formulation is somewhat similar to the biased formulation of batch SVM for learning with imbalanced datasets (Akbani et al. 2004).

To online optimize the above objective (10), following the passive-aggressive learning method (Crammer et al. 2006), we have a similar online optimization objective:

$$\mathbf{w}_{t+1} = \arg\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{2}\|\mathbf{w} - \mathbf{w}_t\|^2 + C\ell_t(\mathbf{w}),$$

which enjoys the following closed-form solution:

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + \tau_t y_t \mathbf{x}_t, \quad \text{where } \tau_t = \min(C, \ell_t(\mathbf{w}_t)). \tag{11}$$

At the t th round, the CSPAA algorithm decides if the class label should be queried according to the same margin based Bernoulli random variable $Z_t \in \{0, 1\}$ as that used in the PAA algorithm. Finally, Algorithm 4 summarizes the details of the proposed CSPAA algorithms.

Algorithm 4 Cost-Sensitive Passive-Aggressive Active learning algorithm (**CSPAA**).

```
INPUT: penalty parameter $C$, bias parameter $\rho$ and smooth parameter $\delta$.
INITIALIZATION : $\mathbf{w}_1 = \mathbf{0}$.
for $t = 1, \ldots, T$ do
  receive an incoming instance $\mathbf{x}_t \in \mathbb{R}^d$;
  predict label $\hat{y}_t = \text{sign}(p_t)$, where $p_t = \mathbf{w}_t \cdot \mathbf{x}_t$;
  draw a Bernoulli random variable $Z_t \in \{0, 1\}$ of parameter $\delta/(\delta + |p_t|)$;
  if $Z_t = 1$ then
    query label $y_t \in \{-1, +1\}$;
    suffer loss $\ell_t(\mathbf{w}_t) = \ell(\mathbf{w}_t; (\mathbf{x}_t, y_t))$;
    $\mathbf{w}_{t+1} = \mathbf{w}_t + \tau_t y_t \mathbf{x}_t$, where $\tau_t = \min\{C, \ell_t(\mathbf{w}_t)\}$;
  else
    $\mathbf{w}_{t+1} = \mathbf{w}_t + \tau_t y_t \mathbf{x}_t$, where $\tau_t = 0$;
  end if
end for
```

Remark It is interesting to analyze the impact of the sampling factor parameter δ. In general, the larger the value of δ, the larger the resulting number of queries issued by the online active learner. In particular, when setting $\delta \to \infty$, it is reduced to the extreme case of querying class label of every instance in the online learning process. In general, one can simply fix δ to some constant to trade off a proper ratio of queries. Besides, an even better approach is to adaptively change the value of δ during the online learning process. In particular, we expect to query more examples at the beginning of the online learning task in order to build a good classifier, and gradually reduce the ratio of queries when the classifier becomes more and more accurate during the online learning process. To this purpose, we suggest a simple yet effective scheme to adaptively update the parameter δ at the t th learning step as: $\delta_t \leftarrow \delta_{t-1} * \frac{t}{t+1}$. We will examine the impact of the sampling factor δ in our experiments.

5.3 Theoretical bound analysis for the CSPAA algorithms

Below gives theoretical analysis of its performance on online binary cost-sensitive active learning tasks in terms of two types of performance metrics. The proofs can be found in the “Appendices 5, 6 and 7”.

Theorem 6 *Let $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_T, y_T)$ be a sequence of examples where $\mathbf{x}_t \in \mathbb{R}^d$ and $y_t \in \{-1, +1\}$ and $\|\mathbf{x}_t\| = 1$ for all t. Then, for any vector $\mathbf{w} \in \mathbb{R}^d$, the expected weighted number of prediction mistakes made by CSPAA on this sequence of examples is bounded as:*

$$\mathbb{E}\left[\sum_{t=1}^{T} \rho_t M_t\right] \leq \frac{1}{\delta}\left\{\left(\frac{1+\delta}{2}\right)^2 \|\mathbf{w}\|^2 + \sum_{t=1}^{T}(1+\delta)C\ell_t(\mathbf{w})\right\},$$

where $C \geq \rho$ is the aggressiveness parameter for CSPAA.

Now our goal is to analyze the performance of the proposed algorithm in terms of the two metrics, sum and $cost$. We first consider the weighted sum of sensitivity and specificity, i.e.,

$$sum = \eta_p \times sensitivity + \eta_n \times specificity,$$

where $\eta_p + \eta_n = 1$ and $\eta_p \geq \eta_n > 0$. The following theorem gives the bound on the sum by the proposed CSPAA algorithm.

Theorem 7 *Let $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_T, y_T)$ be a sequence of examples, where $\mathbf{x}_t \in \mathbb{R}^d$, $y_t \in \{-1, +1\}$ and $\|\mathbf{x}_t\| = 1$ for all t. By setting $\rho = \frac{\eta_p T_n}{\eta_n T_p}$, and assuming $C \geq \rho$, for any $\mathbf{w} \in \mathbb{R}^d$, we have the following bound for the proposed CSPAA algorithm:*

$$\mathbb{E}[sum] \geq 1 - \frac{\eta_n}{T_n}\frac{1}{\delta}\left\{\left(\frac{1+\delta}{2}\right)^2 \|\mathbf{w}\|^2 + \sum_{t=1}^{T}(1+\delta)C\ell_t(\mathbf{w})\right\}.$$

Furthermore, when $\eta_p = \eta_n = 1/2$, the balanced accuracy (BA) is bounded from below by

$$\mathbb{E}[BA] \geq 1 - \frac{1}{2T_n}\frac{1}{\delta}\left\{\left(\frac{1+\delta}{2}\right)^2 \|\mathbf{w}\|^2 + \sum_{t=1}^{T}(1+\delta)C\ell_t(\mathbf{w})\right\}.$$

Remark In the above, setting $\delta = 1$ leads to the following bound

$$\mathbb{E}[sum] \geq 1 - \frac{\eta_n}{T_n}\left\{\|\mathbf{w}\|^2 + 2C\sum_{t=1}^{T}\ell_t(\mathbf{w})\right\}.$$

Setting $\delta = \sqrt{1 + \frac{4C\sum_{t=1}^{T}\ell_t(\mathbf{w}_t)}{\|\mathbf{w}\|^2}}$ leads to the following bound

$$\mathbb{E}[sum] \geq 1 - \frac{\eta_n}{T_n}\left\{\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{t=1}^{T}\ell_t(\mathbf{w}) + \frac{1}{2}\|\mathbf{w}\|\sqrt{\|\mathbf{w}\|^2 + 4C\sum_{t=1}^{T}\ell_t(\mathbf{w})}\right\}.$$

In the above approach, the bias parameter ρ is set to $\frac{\eta_p T_n}{\eta_n T_p}$, in which the ratio $\frac{T_n}{T_p}$ may not be available in advance. To alleviate this issue, we consider another approach using the cost based performance metric. Specifically, we propose to set $\rho = \frac{c_p}{c_n}$, where c_p and c_n are the predefined cost parameters of false negative and false positive, respectively. We assume

$c_p + c_n = 1$ and $0 \leq c_n \leq c_p$ since we would prefer to improve the accuracy of predicting the rare positive examples. By this setting, the following theorem gives us the cumulative cost bound of the proposed CSPAA algorithm.

Theorem 8 *Let* $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_T, y_T)$ *be a sequence of examples, where* $\mathbf{x}_t \in \mathbb{R}^d$, $y_t \in \{-1, +1\}$ *and* $\|\mathbf{x}_t\| = 1$ *for all* t. *By setting* $\rho = \frac{c_p}{c_n}$, *and assuming* $C \geq \rho$, *for any* $\mathbf{w} \in \mathbb{R}^d$, *the overall cost made by the proposed CSPAA algorithm over this sequence of examples is bounded as follows:*

$$\mathbb{E}[cost] \leq c_n \frac{1}{\delta} \left\{ \left(\frac{1+\delta}{2} \right)^2 \|\mathbf{w}\|^2 + \sum_{t=1}^{T} (1+\delta) C \ell_t(\mathbf{w}) \right\}.$$

Remark Setting $\delta = 1$ for the above theorem leads to the following bound:

$$\mathbb{E}[cost] \leq c_n \left\{ \|\mathbf{w}\|^2 + 2C \sum_{t=1}^{T} \ell_t(\mathbf{w}) \right\}.$$

Setting $\delta = \sqrt{1 + \frac{4C \sum_{t=1}^{T} \ell_t(\mathbf{w}_t)}{\|\mathbf{w}\|^2}}$ leads to the following bound:

$$\mathbb{E}[cost] \leq c_n \left\{ \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{t=1}^{T} \ell_t(\mathbf{w}) + \frac{1}{2} \|\mathbf{w}\| \sqrt{\|\mathbf{w}\|^2 + 4C \sum_{t=1}^{T} \ell_t(\mathbf{w})} \right\}.$$

6 Experimental results

In this section, we evaluate the empirical performance of the proposed family of Passive-Aggressive Active-learning algorithms for three types of online active learning tasks: binary classification, multi-class classification, and cost-sensitive classification tasks.

6.1 Evaluation of PAA algorithms for binary classification tasks

This section will evaluate the empirical performance of the proposed PAA algorithms on online binary classification tasks.

6.1.1 Compared algorithms and experimental testbed

We compare the proposed PAA algorithms with the Perceptron-based Active learning, and their random variants, which are listed as follows:

- "RPE": the Random Perceptron algorithm (Cesa-Bianchi and Lugosi 2006);
- "RPA": the Random Passive-Aggressive algorithms, including RPA, RPA-I, RPA-II. These algorithms adopt the same updating strategy as the proposed PAA algorithms, while the querying strategy is different. Instead of actively querying the class label, these algorithms utilize a uniform random sampling approach. The comparison between the RPA algorithms and the proposed PAA algorithms will validate the effectiveness of the active querying strategy.
- "PEA": the Perceptron-based Active learning algorithm (Cesa-Bianchi et al. 2006); This algorithm adopts the same active querying strategy as the proposed PAA algorithm, while

Table 1 Summary of datasets used in binary online classification experiments

Dataset	#Instances	#Features
a8a	32,561	123
codrna	271,617	8
magic04	19,020	10
mushrooms	8124	112
spambase	4601	57
splice	3175	60
svmguide1	7089	4
w8a	64,700	300

the updating rule follows the Perceptron algorithm. The comparison between the proposed algorithms and the Perceptron-based algorithms will support our main motivation, to fully exploit the potential of every queried instance.

- "SEL-ada": the Selective Sampling Perceptron with Adaptive Parameter (Cesa-Bianchi et al. 2006); Unlike the PEA algorithm where the smoothing parameter δ is a constant for all iterations, the δ in the SEL-ada algorithm is set in an online fashion, i.e. $\delta_t \propto \sqrt{1 + ErrCount_{t-1}}$.
- "SEL-2nd": the Selective Sampling Second-Order Perceptron algorithm (Cesa-Bianchi et al. 2006). This algorithm adopts the same active querying strategy as the proposed PAA algorithm, but the update strategy follows the Second-order Perceptron algorithm (Cesa-Bianchi et al. 2005). The SEL-2nd achieved the best performance among all compared algorithms in (Cesa-Bianchi et al. 2006).
- "PAA": the Passive-Aggressive Active learning algorithms, including PAA, PAA-I, PAA-II.

To examine the performance, we conduct extensive experiments on a variety of benchmark datasets from web machine learning repositories. Table 1 shows the details of twelve binary-class datasets used in our experiments. All of these datasets can be downloaded from LIBSVM website[1] and UCI machine learning repository.[2] These datasets are chosen fairly randomly in order to cover various sizes of datasets.

All the compared algorithms learn a linear classifier for the binary classification tasks. The penalty parameter C is searched from $2^{[-5:5]}$ through cross validation for all the algorithms and datasets. The smoothing parameter δ is set as $2^{[-10:10]}$ in order to examine varied sampling situations. All the experiments were conducted over 20 runs of different random permutations for each dataset. All the results were reported by averaging over these 20 runs. For performance metrics, we select F-measure, which is defined as $F\text{-}measure = 2\frac{Precision*Recall}{Precision+Recall}$.

6.1.2 Evaluation on fixed ratio of queries

In the first experiment, we evaluate the performance of our proposed PAA algorithms on the online binary classification task with fixed query rates. We adjust δ to make the percentage of queried instances near 10 and 20 % and compare all the algorithms on a fair platform. The results are shown in Tables 2 and 3. Several observations can be drawn from the results.

[1] http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/.

[2] http://www.ics.uci.edu/~mlearn/MLRepository.html.

Table 2 Evaluation of the PAA algorithms against the other baselines (time in seconds)

Algorithm	Request 10 % labels			Request 20 % labels		
	F-measure	Time	Query (%)	F-measure	Time	Query (%)
svmguide1						
PEA	0.837 ± 0.006	0.149	9.77 ± 0.82	0.832 ± 0.003	0.151	20.13 ± 0.76
SEL$_{ada}$	0.836 ± 0.005	0.154	9.83 ± 1.34	0.831 ± 0.005	0.153	19.83 ± 1.83
SEL$_{2nd}$	0.838 ± 0.009	0.232	9.91 ± 2.62	0.834 ± 0.007	0.230	20.99 ± 5.98
RPE	0.800 ± 0.008	0.030	9.72 ± 0.31	0.796 ± 0.006	0.031	20.21 ± 0.47
RPA	0.793 ± 0.009	0.031	9.82 ± 0.29	0.796 ± 0.008	0.033	19.82 ± 0.51
RPA-I	0.859 ± 0.003	0.031	9.67 ± 0.39	0.860 ± 0.002	0.033	19.95 ± 0.39
RPA-II	0.858 ± 0.004	0.032	9.69 ± 0.35	0.858 ± 0.002	0.034	19.85 ± 0.56
PAA	0.830 ± 0.006	0.152	9.90 ± 0.62	0.827 ± 0.005	0.152	19.86 ± 0.91
PAA-I	**0.864 ± 0.002** ($p < 0.0001$)	0.152	9.71 ± 0.29	**0.864 ± 0.002** ($p < 0.0001$)	0.155	20.00 ± 0.49
PAA-II	**0.863 ± 0.001** ($p < 0.0001$)	0.152	9.73 ± 0.37	**0.862 ± 0.002** ($p = 0.0031$)	0.155	19.93 ± 0.60
mushrooms						
PEA	0.991 ± 0.001	0.184	9.85 ± 0.46	0.994 ± 0.001	0.183	20.76 ± 0.93
SEL$_{ada}$	0.990 ± 0.002	0.188	9.870 ± 0.63	0.992 ± 0.001	0.188	19.87 ± 0.82
SEL$_{2nd}$	0.993 ± 0.001	1.423	10.36 ± 0.77	0.995 ± 0.001	1.384	19.82 ± 1.20
RPE	0.971 ± 0.006	0.040	9.82 ± 0.28	0.984 ± 0.003	0.041	20.95 ± 0.37
RPA	0.987 ± 0.002	0.041	10.21 ± 0.19	0.992 ± 0.002	0.042	20.26 ± 0.42
RPA-I	0.986 ± 0.003	0.041	10.15 ± 0.22	0.992 ± 0.001	0.042	20.29 ± 0.36
RPA-II	0.986 ± 0.002	0.041	9.86 ± 0.34	0.992 ± 0.002	0.042	20.52 ± 0.35
PAA	**0.996 ± 0.000**	0.183	10.09 ± 0.33	**0.997 ± 0.000**	0.185	20.24 ± 0.58
PAA-I	**0.996 ± 0.001** ($p < 0.0001$)	0.183	10.14 ± 0.35	**0.997 ± 0.000** ($p < 0.0001$)	0.186	20.33 ± 0.37
PAA-II	**0.996 ± 0.001** ($p < 0.0001$)	0.184	9.73 ± 0.44	**0.997 ± 0.000** ($p < 0.0001$)	0.185	20.43 ± 0.51
a8a						
PEA	0.572 ± 0.008	0.763	10.29 ± 0.28	0.567 ± 0.007	0.777	19.81 ± 0.70
SEL$_{ada}$	0.568 ± 0.006	0.785	10.36 ± 0.58	0.565 ± 0.007	0.791	19.83 ± 1.06
SEL$_{2nd}$	0.569 ± 0.007	6.960	9.81 ± 0.67	0.574 ± 0.006	6.312	20.36 ± 1.25
RPE	0.510 ± 0.005	0.189	10.29 ± 0.15	0.514 ± 0.005	0.195	19.87 ± 0.21
RPA	0.512 ± 0.006	0.200	9.77 ± 0.11	0.516 ± 0.006	0.211	20.22 ± 0.23
RPA-I	0.606 ± 0.003	0.200	9.97 ± 0.16	0.608 ± 0.003	0.210	19.91 ± 0.20
RPA-II	0.603 ± 0.003	0.204	9.71 ± 0.13	0.606 ± 0.002	0.216	19.93 ± 0.18
PAA	0.571 ± 0.006	0.776	9.72 ± 0.41	0.566 ± 0.005	0.790	20.20 ± 0.47
PAA-I	**0.621 ± 0.003** ($p < 0.0001$)	0.780	9.90 ± 0.83	**0.626 ± 0.003** ($p < 0.0001$)	0.796	19.90 ± 0.39
PAA-II	**0.623 ± 0.004** ($p < 0.0001$)	0.783	9.73 ± 0.39	**0.628 ± 0.004** ($p < 0.0001$)	0.798	19.90 ± 0.37

Obviously, the two proposed soft margin algorithms, PAA-I and PAA-II are always the two winners (as highlighted). To test the significance of our experiment, we compare the F-measures of PAA-I and PAA-II with that of the best one among all the 7 compared algorithms and report the p value of the t test

Table 3 Evaluation of the PAA algorithms against the other baselines (time in seconds)

Algorithm	Request 10 % labels			Request 20 % labels		
	F-measure	Time	Query (%)	F-measure	Time	Query (%)
spambase						
PEA	0.846 ± 0.006	0.099	10.06 ± 0.46	0.854 ± 0.005	0.100	20.51 ± 0.67
SEL_{ada}	0.835 ± 0.012	0.102	10.45 ± 0.79	0.849 ± 0.004	0.102	20.22 ± 1.08
SEL_{2nd}	0.843 ± 0.012	0.279	9.85 ± 0.57	0.859 ± 0.006	0.281	20.36 ± 0.77
RPE	0.800 ± 0.012	0.020	10.04 ± 0.44	0.819 ± 0.007	0.021	20.51 ± 0.89
RPA	0.827 ± 0.011	0.021	9.94 ± 0.47	0.838 ± 0.009	0.022	20.33 ± 0.60
RPA-I	0.858 ± 0.007	0.021	9.87 ± 0.50	0.875 ± 0.003	0.022	20.10 ± 0.63
RPA-II	0.860 ± 0.005	0.022	10.01 ± 0.47	0.875 ± 0.005	0.023	19.98 ± 0.64
PAA	0.865 ± 0.006	0.100	9.83 ± 0.56	0.867 ± 0.006	0.101	20.35 ± 0.84
PAA-I	**0.881 ± 0.004**	0.101	9.72 ± 0.41	**0.888 ± 0.002**	0.103	20.06 ± 0.44
	($p < 0.0001$)			($p < 0.0001$)		
PAA-II	**0.884 ± 0.003**	0.101	9.91 ± 0.40	**0.889 ± 0.003**	0.104	19.71 ± 0.50
	($p < 0.0001$)			($p < 0.0001$)		
splice						
PEA	0.747 ± 0.014	0.068	9.813 ± 0.48	0.770 ± 0.007	0.068	19.73 ± 0.74
SEL_{ada}	0.740 ± 0.012	0.070	9.95 ± 0.49	0.759 ± 0.007	0.071	20.01 ± 0.93
SEL_{2nd}	0.752 ± 0.012	0.206	9.71 ± 0.49	0.773 ± 0.008	0.209	20.54 ± 1.06
RPE	0.718 ± 0.015	0.014	9.80 ± 0.59	0.746 ± 0.012	0.015	19.99 ± 0.55
RPA	0.741 ± 0.015	0.015	10.12 ± 0.42	0.763 ± 0.008	0.016	20.47 ± 0.72
RPA-I	0.768 ± 0.011	0.015	9.62 ± 0.57	0.793 ± 0.006	0.016	19.94 ± 0.85
RPA-II	0.771 ± 0.010	0.015	9.96 ± 0.47	0.795 ± 0.008	0.016	20.02 ± 0.49
PAA	0.780 ± 0.008	0.070	9.96 ± 0.39	0.790 ± 0.006	0.071	20.23 ± 0.67
PAA-I	**0.794 ± 0.008**	0.070	9.70 ± 0.44	**0.813 ± 0.005**	0.071	19.87 ± 0.69
	($p < 0.0001$)			($p < 0.0001$)		
PAA-II	**0.790 ± 0.008**	0.070	9.72 ± 0.39	**0.812 ± 0.005**	0.072	20.17 ± 0.63
	($p < 0.0001$)			($p < 0.0001$)		
w8a						
PEA	0.263 ± 0.015	1.799	10.04 ± 0.32	0.342 ± 0.013	1.847	20.05 ± 0.26
SEL_{ada}	0.236 ± 0.018	1.887	9.98 ± 0.57	0.304 ± 0.016	1.819	20.06 ± 0.39
SEL_{2nd}	0.363 ± 0.019	42.26	9.72 ± 0.53	0.428 ± 0.016	42.45	20.48 ± 0.83
RPE	0.188 ± 0.007	0.705	10.01 ± 0.09	0.225 ± 0.007	0.590	20.07 ± 0.15
RPA	0.198 ± 0.013	0.730	9.78 ± 0.14	0.249 ± 0.010	0.636	20.03 ± 0.15
RPA-I	0.280 ± 0.009	0.728	9.83 ± 0.11	0.370 ± 0.008	0.628	19.88 ± 0.12
RPA-II	0.227 ± 0.011	0.736	10.06 ± 0.12	0.290 ± 0.011	0.661	19.93 ± 0.12
PAA	0.385 ± 0.015	1.873	9.78 ± 0.32	0.441 ± 0.021	1.907	20.03 ± 0.66
PAA-I	**0.391 ± 0.015**	1.961	9.81 ± 0.16	**0.461 ± 0.009**	1.943	19.92 ± 0.22
	($p < 0.0001$)			($p < 0.0001$)		
PAA-II	**0.386 ± 0.015**	1.897	9.98 ± 0.26	**0.458 ± 0.012**	1.857	19.89 ± 0.37
	($p < 0.0001$)			($p < 0.0001$)		

Obviously, the two proposed soft margin algorithms, PAA-I and PAA-II are always the two winners (as highlighted). To test the significance of our experiment, we compare the F-measures of PAA-I and PAA-II with that of the best one among all the 7 compared algorithms and report the p value of the t test

First of all, we observe that all the active learning algorithms outperform their corresponding random versions in terms of F-measure results, which validates the efficacy and advantage of the active learning strategies.

Second, we find that the two soft-margin PAA algorithms (i.e., PAA-I and PAA-II) achieve similar F-measure performance on most of the datasets, while the hard-margin PAA usually performs slightly worse. This may possibly be caused by overfitting on noisy training data, since PAA conducts a more aggressive update and is thus more sensitive to noise.

Third, under the same fraction of queried labels, the two soft PAA algorithms always achieve significantly higher F-measure than all compared algorithms. This promising result indicates that PAA can effectively exploit those requested labeled data, especially for those that are correctly classified but with low confidence. And this observation again supports our main motivation that the PAA algorithms are designed to address the key limitation of PEA who wastes the efforts of querying labels but may never uses them for effective update. Furthermore, the running time cost of PAA and PEA algorithms are similar, as well as in the same order of magnitude with randomized query algorithms, which validates the efficiency of the proposed methods.

Finally, we would like to discuss the performance of the SEL-ada and SEL-2nd algorithms, which were proposed as improved variants of the PEA algorithm. Surprisingly, SEL-ada performs slightly worse than PEA in all the datasets. This observation consists with earlier results while no explanation was provided by Cesa-Bianchi et al. (2006). We think that the inferior performance of SEL-ada may be caused by the ineffective setting the smoothing parameter δ. When aiming at maximizing the accumulated accuracy along the whole learning process, querying labels earlier is more likely to result in better performance. However, the increasing δ with time t is actually encouraging later query.

When analyzing the performance of SEL-2nd, we find that this algorithm usually outperforms PEA in terms of F-measure, which can be explained by the utilization of second order information. However, it still suffers from the same drawback with PEA, i.e. wasting the queried labels when the prediction is correct but with low confidence. Thus the F-measure is still significantly lower than PAA-I and PAA-II. Besides, the $O(d^2)$ time and space complexity of SEL-2nd limits its application in high dimensional applications, which is indicated by its time cost on the “w8a” dataset with 300 feature dimension.

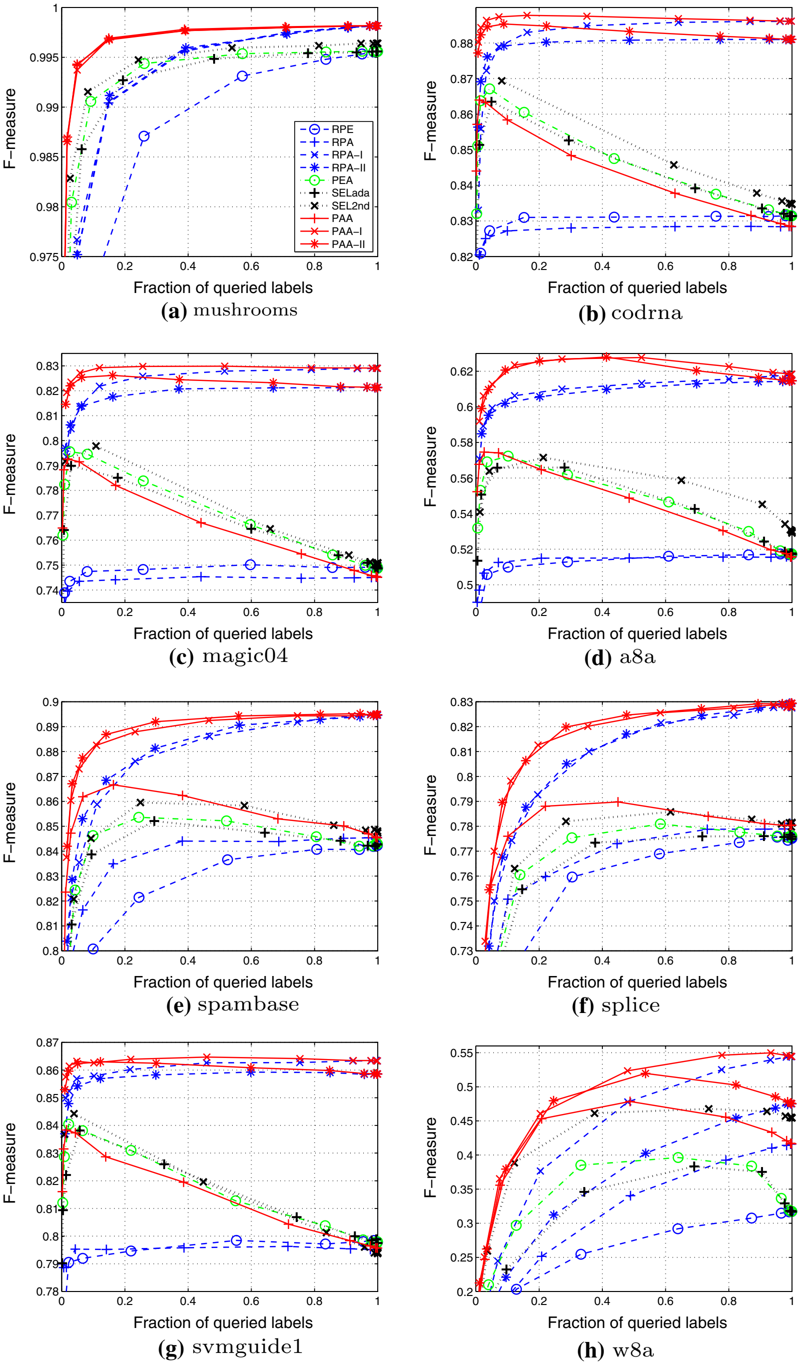
6.1.3 Evaluation on varied ratio of queries

This experiment is to evaluate the performance of the proposed algorithms by varying the query rate of different online learning algorithms, shown in Fig. 1. From the experimental results, several observations can be drawn.

Similar to the previous experiments under fixed query rate, we find that the two soft margin PAA algorithms always achieve higher F-measure than all the other active learning algorithms, which validates the successful update strategy of our proposed algorithm. And all the active learning algorithms outperform their corresponding random versions, which demonstrates the advantage of the active querying rule.

In addition, we observe that the F-measure usually increases as the fraction of queried labels increases at the beginning, but saturates quickly after the fraction of queried labels exceeds some value. This result indicates the proposed online active learning strategy can effectively explore those most informative instances for updating the classifiers in a rather effective and efficient way.

Finally, it is interesting to see that on some datasets (e.g., a8a, magic04, svmguide1, etc.), the F-measures achieved by PAA,PEA, SEL-ada and SEL-2nd could decrease when



(a) mushrooms **(b)** codrna

(c) magic04 **(d)** a8a

(e) spambase **(f)** splice

(g) svmguide1 **(h)** w8a

Fig. 1 Evaluation of F-measure against the fraction of queried labels on all binary datasets. The plotted curves are averaged over 20 random permutations

increasing the fraction of queried labels. This seems a little bit surprising as we usually expect the more the labeled data queried, the better the predictive performance. Note that this phenomenon only appears in the hard margin algorithms (PAA, PEA, SEL-ada and SEL-2nd), which are not designed for noisy data. While the other two soft-margin algorithms (PAA-I and PAA-II), which are robust to noisy, tend to be able to avoid such situations. Consequently, we suspect this was mainly caused due to the overfit issue on the noisy training data.

6.1.4 Application to online text classification

In this section, we apply our proposed Passive-Aggressive Active learning algorithms to online text classification. Our experimental testbed consists of: (i) a subset of the Reuters Corpus Volume 1 (RCV1)[3] which contains 4086 documents with 29,992 distinct words; (ii) 20 Newsgroups datasets,[4] we extract the "comp" versus "rec" and "rec" versus "sci" to form two binary classification tasks, which have a total of 8870 and 8928 documents, respectively. Each document is represented by a feature vector of 26,214 distinct words.

As discussed earlier, the SEL-ada algorithm mostly performs worse that PEA because of the ineffective setting of the δ value. In addition, the SEL-2nd is not applicable to high dimensional applications due to its $O(d^2)$ space and time complexity while its performance is always worse than PAA-I and PAA-II. We remove the two compared algorithms for the conciseness of our later experiments.

The text classification results are shown in Fig. 2. We could see that Passive-Aggressive based algorithms usually outperform the Perceptron based algorithms, which empirically shows the advantages of large margin approaches for active learning. Among all methods, PAA algorithms consistently perform better than random querying methods and perceptron based active learning methods, which further validates the efficacy of our proposed approaches.

6.1.5 Application to web data classification

To further evaluate the PAA algorithms, we apply them to web data classification tasks, which are (i) URL classification (Ma et al. 2009b) which contains 1,782,206 URLs with 3,231,961 features; (ii) webspam classification (Wang et al. 2012), which have a total of 350,000 instance with 254 features, respectively. These two datasets can be downloaded from the LIBSVM website.[5] Similar phenomenon could be observed from the results, as shown in Fig. 3.

6.2 Evaluation of the MPAA algorithm in multi-class classification tasks

This section will evaluate the empirical performance of the proposed MPAA algorithms on online multi-class classification tasks.

6.2.1 Compared algorithms and experimental testbed

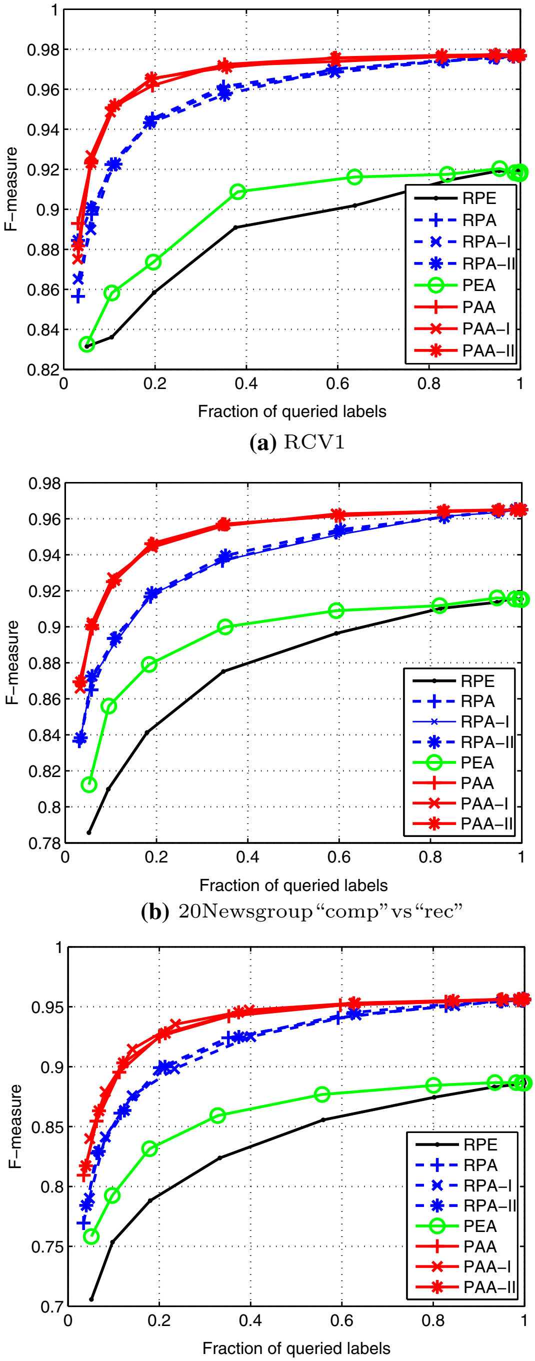
We compare the proposed MPAA algorithms with the Multi-class Perceptron Active learning algorithm, which adopts the same querying strategy as MPAA algorithms do but updates in

[3] http://thedatahub.org/dataset/rcv1.

[4] http://qwone.com/~jason/20Newsgroups/.

[5] http://www.csie.ntu.edu.tw/~cjlin/libsvmtools.

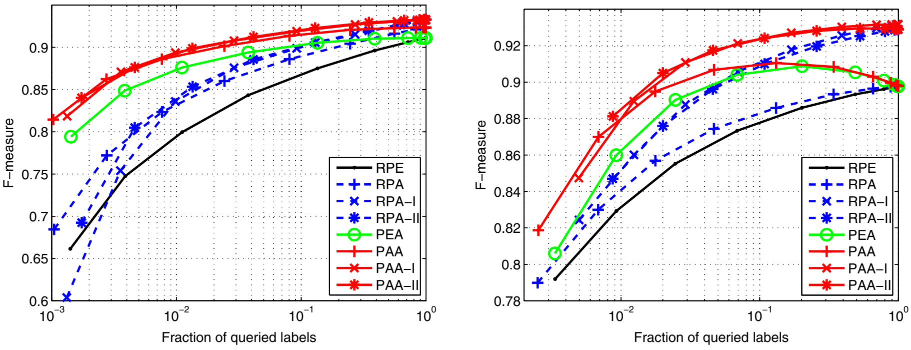
Fig. 2 Evaluation of F-measure against the fraction of queried labels for text classification applications



(a) RCV1

(b) 20Newsgroup “comp” vs “rec”

(c) 20Newsgroup “rec” vs “sci”



(a) URL **(b)** Webspam

Fig. 3 Evaluation of F-measure against the fraction of queried labels for web applications

Table 4 Details of multi-class classification datasets

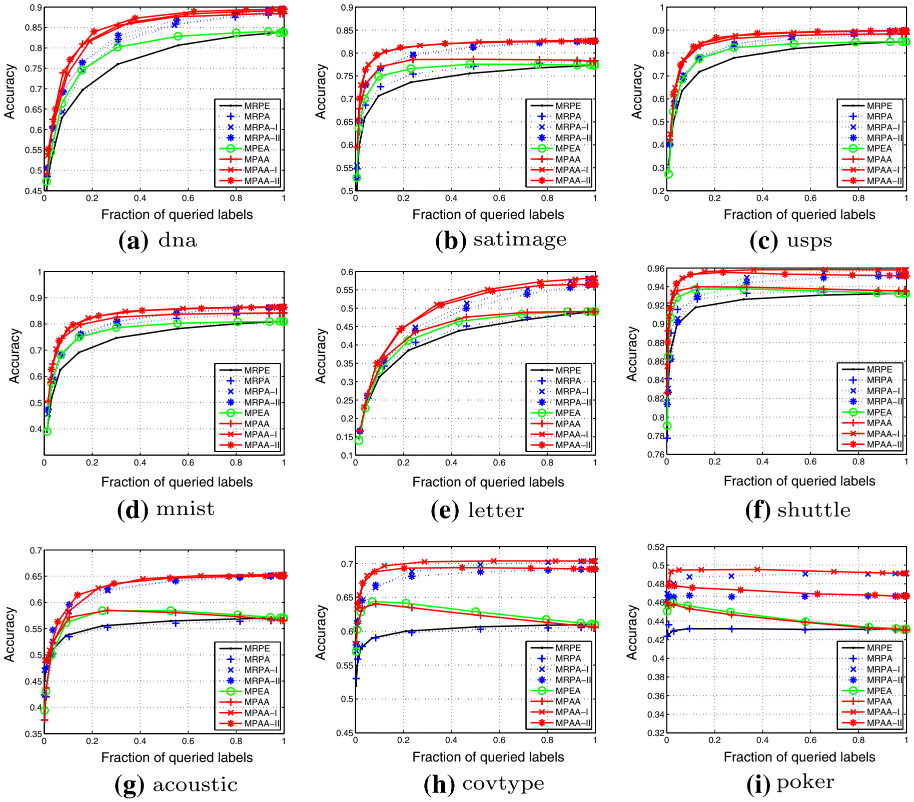
Dataset	#Instances	#Features	#Classes
dna	2000	180	3
satimage	4435	36	6
usps	7291	256	10
mnist	10,000	780	10
letter	15,000	16	26
shuttle	43,500	9	7
acoustic	78,823	50	3
covtype	581,012	54	7
poker	1,000,000	10	10

the Perceptron rule. To demonstrate the advantage of our querying strategy, we also compare the active learning algorithms with their random variants. Note that for all Perceptron based algorithms, we choose the max-score variant since it is similar to the PA based algorithms in only updating two weight vectors during each iteration, which is a fair comparison. The compared algorithms are listed as follows:

- "MRPE": the Multi-class Random Perceptron algorithm, an extension of RPE (Cesa-Bianchi and Lugosi 2006) to multi-class setting;
- "MRPA": the Multi-class Random Passive-Aggressive algorithms, including MRPA, MRPA-I, MRPA-II, which will uniformly randomly query labels;
- "MPEA": the Multi-class Perceptron-based Active learning algorithm, an extension of PEA (Cesa-Bianchi et al. 2006) in multi-class setting;
- "MPAA": the Multi-class Passive-Aggressive Active learning algorithms, including MPAA, MPAA-I, MPAA-II.

To examine the performance, we conduct extensive experiments on a variety of benchmark datasets from web machine learning repositories. Table 4 shows the details of 9 multi-class datasets used in our experiments. All of these datasets can be downloaded from LIBSVM website.[6] These datasets are chosen fairly randomly in order to cover various sizes of datasets.

[6] http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/.



(a) dna **(b)** satimage **(c)** usps

(d) mnist **(e)** letter **(f)** shuttle

(g) acoustic **(h)** covtype **(i)** poker

Fig. 4 Evaluation of accuracy against the fraction of queried labels on all multi-class datasets. The plotted curves are averaged over 20 random permutations

The parameter settings mostly follow those in the previous binary classification experiments excepts that we select online accuracy for performance metrics.

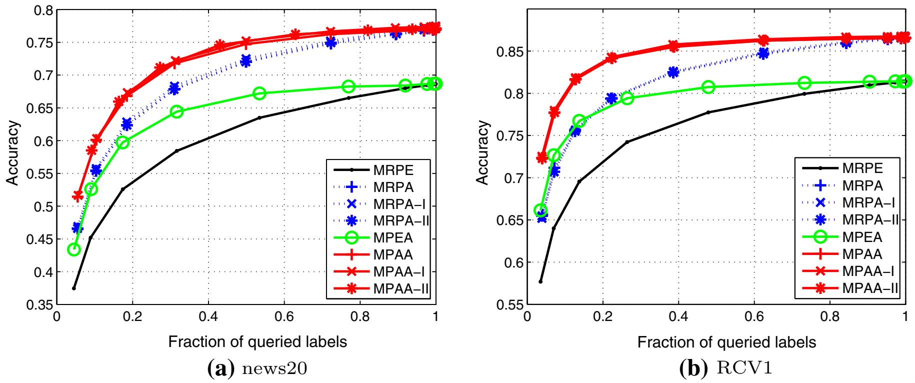
6.2.2 Performance evaluation

Next we evaluate the performance of all the algorithms on online multi-class active learning tasks. Figure 4 summarizes the average performance of the eight different algorithms for online active learning on the 9 datasets. Similar phenomenon could be observed from the results in Fig. 4 as that in binary setting, which further demonstrates that our proposed algorithms are effective in dealing with online multi-class active learning tasks.

6.2.3 Application to online text classification

In this section, we apply our proposed Multi-class Passive-Aggressive Active learning algorithms to online text classification. Our experimental testbed consists of: (i) 20 Newsgroups datasets,[7] we use the 20-class dataset, which have a total of 15,935 documents. Each document is represented by a feature vector of 62,061 distinct words. (ii) Reuters Corpus Volume

[7] http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multiclass.html#news20.



(a) news20 **(b)** RCV1

Fig. 5 Evaluation of accuracy against the fraction of queried labels for multi-class online text classification applications

1 (RCV1)[8] which is a 53-class datasets and contains 15,564 documents with 47,236 distinct words; The text classification results are shown in Fig. 5. Similar phenomenon could be observed from the results: Passive-Aggressive based algorithms usually outperform the Perceptron based algorithms and PAA algorithms consistently perform better than random querying methods and perceptron based active learning methods, which further validates the efficacy of our proposed approaches.

6.3 Evaluation of CSPAA algorithms for cost-sensitive classification tasks

This section will evaluate the empirical performance of the proposed CSPAA algorithm. Specifically, we will evaluate all the algorithms on online malicious URL detection (Ma et al. 2009a), which is a large-scale online learning task. Our experiments are designed to answer several open questions: (i) how does the class imbalance issue affect the predictive performance of online active learning? (ii) if the proposed online active learning approach is effective to reducing the amount of labeled data significantly in order to maintain comparable performance? (iii) how is the efficiency and scalability of the proposed learning algorithms for a web-scale application?

6.3.1 Experimental testbed

To examine the performance of the proposed CSPAA algorithms, we test them on a large-scale benchmark dataset for malicious URL detection tasks (Ma et al. 2009b), which can be downloaded from http://sysnet.ucsd.edu/projects/url/. The original data set was created in purpose to make it somehow class-balanced. In our experiment, we create two subsets by sampling from the original data set to make it close to a more realistic distribution scenario where the number of normal URLs is significantly larger than the number of malicious URLs. Table 5 shows the data sets used in our experiment for online malicious detection, where T_p/T_n denotes the ratio between the number of positive (malicious) instances and the number of negative (normal) instances. A variety of features were extracted to represent the content of a URL, including both lexical features (such as hostnames, primary domain, path

[8] http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multiclass.html#rcv1.multiclass.

Table 5 The data set of malicious URL detection

Dataset	#Training examples	#Features	T_p/T_n
URL1	1,000,000	3,231,961	1:9
URL2	1,000,000	3,231,961	1:99

tokens, etc) and host-based features (such as WHOIS info, IP prefix, AS number, Geographic, etc.). More details can be found in (Ma et al. 2009b).

6.3.2 Compared algorithms and setup

We compare the proposed CSPAA algorithms against a variety of state-of-the-art algorithms as follows:

- "PE": the classical PErceptron algorithm (Rosenblatt 1958), which queries label of every instance; this is impractical as it requires huge amount of labeled data, which is used as a yardstick to evaluate the efficacy of our algorithm;
- "PA": the regular Passive-Aggressive algorithm (Crammer et al. 2006), which also queries class label of every instance; similarly, this is another yardstick for comparison;
- "CW-diag": the Confidence Weighted (CW) algorithm (Crammer et al. 2008), which also queries label of every instance, and exploits the second-order info. We adopt the CW-diag version to make it feasible for high-dimensional data.
- "PAUM": this is the cost-sensitive Perceptron Algorithm with Uneven Margin (Li et al. 2002), which also queries label of every instance;
- "CPA": the Cost-sensitive Passive-Aggressive algorithm based on prediction (Crammer et al. 2006) which also queries all labels;
- "LEPE": the Label Efficient Perceptron algorithm (Cesa-Bianchi et al. 2006), which actively queries label for informative instances;
- "CSRND": a variant of the proposed CSPAA algorithm, but *randomly* queries label of incoming instances;
- "CSPAA": the proposed Cost-Sensitive Passive-Aggressive Active learning algorithm as shown in Algorithm 4.

To make a fair comparison, all the algorithms adopt the same setup. All the compared algorithms learn a linear classifier for the malicious URL detection task. In particular, for all the compared algorithms, we set the penalty parameter $C = \rho = T_n/T_p$. For the proposed CSPAA$_{sum}$ algorithm, we set $\eta_p = \eta_n = 1/2$ for all cases, while for the CSPAA$_{cos}$, we set $c_p = T_n/T$ and $c_n = T_p/T$. The smoothing parameter δ for LEPE and CSPAA is set as $2^{[-10:2:10]}$ in order to examine varied query ratios.

All the experiments were conducted over 5 random permutations of the dataset. The results were reported by averaging over these 5 runs. We evaluate the online classification performance by two key metrics: the weighted **sum** of sensitivity and specificity, and the weighted **cost**. We denote by CSPAA$_{sum}$ the algorithm aiming to improve the weighted sum of sensitivity and specificity, and CSPAA$_{cos}$ the algorithm aiming to improve the overall cost. All experiments were run on a machine of 2.3GHz CPU.

6.3.3 Evaluation on fixed ratio of queries

The first experiment is to evaluate the performance by fixing the ratio of queries issued by the (active learning) algorithms. Table 6 shows the results of the *sum* performance under a fixed

Table 6 "Sum" evaluation of cost-sensitive online classification for malicious URL detection

Algorithm	Sum (%)	Sensi (%)	Speci (%)	Accur (%)	Time (s)	Query (%)
URL1[a]						
PE	94.582	90.247	98.917	98.050	8.218	100.000
	(±0.023)	(±0.041)	(±0.005)	(±0.008)		
PA	95.119	91.094	99.144	98.339	11.321	100.000
	(±0.025)	(±0.045)	(±0.006)	(±0.009)		
CW-diag	96.255	93.099	99.411	98.779	15.398	100.000
	(±0.034)	(±0.064)	(±0.004)	(±0.010)		
PAUM	96.734	95.533	97.935	97.695	11.782	100.000
	(±0.011)	(±0.029)	(±0.014)	(±0.011)		
CPA	96.567	94.905	98.229	97.896	18.187	100.000
	(±0.027)	(±0.052)	(±0.004)	(±0.007)		
LEPE	93.642	88.334	98.949	97.887	8.398	**10.087**
	(±0.040)	(±0.083)	(±0.008)	(±0.008)		(±0.055)
CSRND	95.676	95.214	96.138	96.045	9.005	**10.035**
	(±0.052)	(±0.068)	(±0.081)	(±0.073)		(±0.120)
CSPAA	96.712	96.382	97.042	96.976	8.772	**10.059**
	(±0.015)	(±0.029)	(±0.029)	(±0.025)		(±0.066)
CSPAA(a)	**96.891**	96.546	97.236	97.167	8.831	**10.077**
	(±0.017)	(±0.029)	(±0.030)	(±0.027)		(±0.073)
URL2[b]						
PE	87.012	74.284	99.741	99.486	18.903	100.000
	(±0.100)	(±0.199)	(±0.002)	(±0.004)		
PA	87.203	74.544	99.862	99.609	27.458	100.000
	(±0.059)	(±0.115)	(±0.003)	(±0.004)		
CW-diag	88.550	77.160	99.940	99.712	48.616	100.000
	(±0.067)	(±0.133)	(±0.001)	(±0.002)		
PAUM	89.049	78.770	99.329	99.123	28.527	100.000
	(±0.083)	(±0.166)	(±0.002)	(±0.003)		
CPA	**92.748**	86.410	99.087	98.960	41.248	100.000
	(±0.078)	(±0.154)	(±0.005)	(±0.006)		
LEPE	79.162	58.492	99.833	99.419	19.414	**2.019**
	(±0.476)	(±0.957)	(±0.011)	(±0.010)		(±0.057)
CSRND	87.776	79.018	96.534	96.358	20.984	**2.018**
	(±0.410)	(±0.711)	(±0.286)	(±0.284)		(±0.025)
CSPAA	**92.697**	88.156	97.237	97.146	20.304	**2.029**
	(±0.245)	(±0.513)	(±0.045)	(±0.042)		(±0.018)

We compare the Sum of our proposed algorithms with the best one among all compared algorithms with the same query rate and report the p value of the t test

[a] CSPAA versus CSRND $p < 0.0001$; CSPAA(a) versus CSRND $p < 0.0001$

[b] CSPAA versus CSRND $p < 0.0001$

Table 7 “Cost” evaluation of cost-sensitive online classification for malicious URL detection

Algorithm	Cost	Sensi (%)	Speci (%)	Accur (%)	Time (s)	Query (%)
URL1[a]						
PE	9752.120 (±40.872)	90.247 (±0.041)	98.917 (±0.005)	98.050 (±0.008)	8.111	100.000
PA	8785.540 (±45.050)	91.094 (±0.045)	99.144 (±0.006)	98.339 (±0.009)	11.376	100.000
CW-diag	6741.420 (±60.673)	93.099 (±0.064)	99.411 (±0.004)	98.779 (±0.010)	15.578	100.000
PAUM	5878.340 (±20.251)	95.533 (±0.029)	97.935 (±0.014)	97.695 (±0.011)	11.645	100.000
CPA	6179.400 (±48.521)	94.905 (±0.052)	98.229 (±0.004)	97.896 (±0.007)	18.229	100.000
LEPE	11,471.580 (±124.696)	88.322 (±0.141)	98.932 (±0.008)	97.871 (±0.013)	8.441	**10.051** (±0.139)
CSRND	7838.800 (±55.313)	95.107 (± 0.096)	96.183 (±0.079)	96.076 (±0.064)	8.904	**10.029** (±0.049)
CSPAA	5889.140 (±43.974)	96.403 (±0.053)	97.053 (±0.014)	96.988 (±0.012)	8.780	**10.010** (±0.042)
CSPAA(a)	**5604.460** (±50.018)	96.546 (±0.059)	97.227 (±0.018)	97.159 (±0.015)	8.759	**10.052** (±0.057)
URL2[b]						
PE	2571.568 (±19.862)	74.284 (±0.199)	99.741 (±0.002)	99.486 (±0.004)	19.994	100.000
PA	2533.800 (±11.678)	74.544 (±0.115)	99.862 (±0.003)	99.609 (±0.004)	28.800	100.000
CW-diag	2267.124 (±13.216)	77.160 (±0.133)	99.940 (±0.001)	99.712 (±0.002)	47.747	100.000
PAUM	2057.452 (±20.843)	79.840 (±0.208)	99.378 (±0.005)	99.182 (±0.006)	28.939	100.000
CPA	**1435.806** (±15.494)	86.410 (±0.154)	99.087 (±0.005)	98.960 (±0.006)	42.687	100.000
LEPE	4214.998 (±125.053)	57.592 (±1.275)	99.832 (±0.013)	99.410 (±0.007)	21.655	**1.984** (±0.045)
CSRND	2314.544 (±126.476)	80.030 (±1.265)	96.591 (±0.130)	96.425 (±0.130)	20.371	**2.027** (±0.043)
CSPAA	1482.338 (±31.270)	87.742 (±0.324)	97.285 (±0.029)	97.189 (±0.027)	22.237	**2.027** (±0.030)

We compare the Cost of our proposed algorithms with the best one among all compared algorithms with the same query rate and report the p value of the t test

[a] CSPAA versus CSRND $p < 0.0001$; CSPAA(a) versus CSRND $p < 0.0001$

[b] CSPAA versus CSRND $p < 0.0001$

ratio of queries to about 2 % for URL1 and 10 % for URL2 dataset, and Table 7 summarizes the *cost* performance under the similar query ratio.

Several observations can be drawn from the results. First of all, according to the classification *accuracy* (a misleading metric for cost-sensitive classification), we found that both PE and PA algorithms significantly outperform the other algorithms, while, in terms of both *sum* and *cost* measures, they are considerably worse than their cost-sensitive variants (i.e., PAUM and CPA). This indicates the importance of taking the class imbalance issue into consideration for online malicious detection tasks. Second, when querying the same ratio of labeled data, in terms of both *sum* and *cost* performances, CSPAA significantly outperforms the LEPE algorithm, which validates the effectiveness of the proposed cost-sensitive online updating strategy. Third, when querying the same ratio of labels, CSPAA significantly outperforms CSRND, which implies the proposed querying strategy is able to actively select those fairly informative instances for querying labels, which are considerably better than just randomly querying. Moreover, among all the approaches, the proposed CSPAA algorithm and the PAUM algorithm achieve the highest *sum* performance. However, the proposed CSPAA only queried a extremely small subset of labels, while the PAUM algorithm requires to query the labels of all the incoming instances, which is very expensive to label 1-million training instances in a real-world application. We thus believe the proposed CSPAA algorithm is more practically attractive and suitable for a web-scale application.

Finally, we notice that the proposed CSPAA algorithm not only achieves the best *sensitivity* performance, but also achieves fairly good specificity performance which is generally quite comparable to the other algorithms. This implies that the proposed CSPAA algorithm not only significantly improves the prediction accuracy on the rare class, but also does not sacrifice much the prediction accuracy on the other majority classes. This promising observation again validates the effectiveness of the proposed CSPAA algorithm.

6.3.4 Evaluation on varied query ratios

This experiment is to evaluate the performance of the proposed algorithms by varying the ratios of queries for comparing different online active learning algorithms. Figure 6 shows the online average *sum* performance and the online average *cost* performance under varied query ratios, respectively. From the experimental results, several observations can be drawn as follows.

First of all, among all four fully supervised online learning algorithms (PE, PA, PAUM, and CPA), the cost-sensitive algorithms (PAUM and CPA) generally outperform the cost-insensitive versions. This result validates the importance of studying the proposed cost-sensitive online learning methodology class imbalanced tasks.

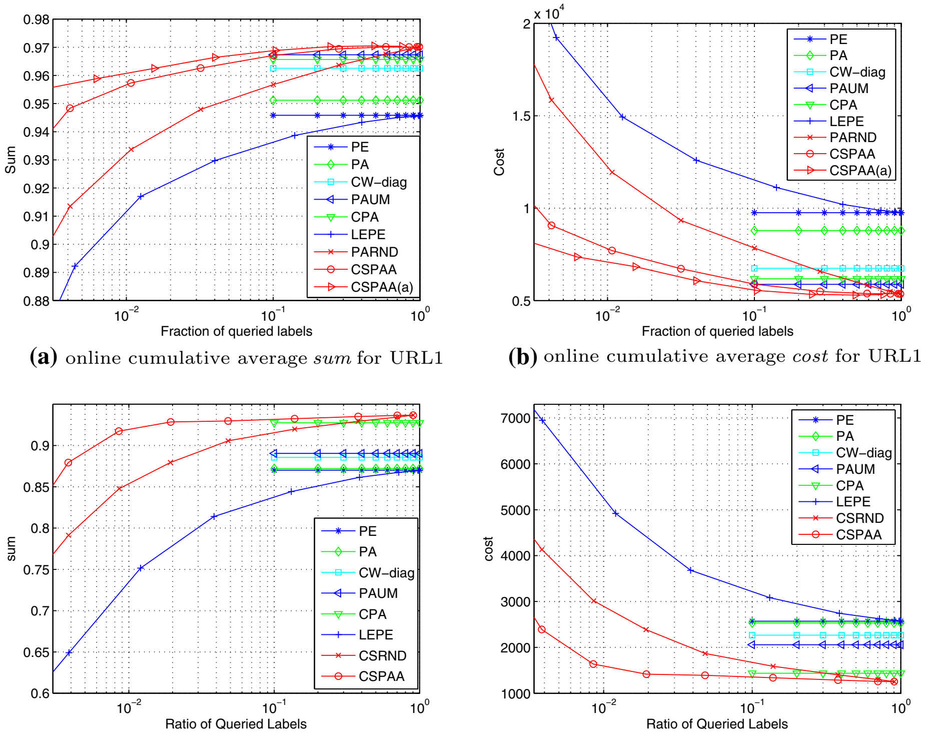
Second, compared with the CSRND algorithm that randomly queries the labels, CSPAA consistently achieves much higher *sum* and much lower *cost* performance over all the ratios of queried labels, especially when the query ratio is relatively small. This promising result indicates that the querying strategy of the proposed CSPAA technique is able to effectively query the informative labeled data from the sequentially arriving of unlabeled data instances.

Third, compared with LEPE, CSPAA achieves higher *sum* over all the ratios of queried labels, which implies that the proposed online updating strategy is able to effectively exploits the labeled data for improving the classifier. In addition, compared with PA, CSPAA with query ratio equals to 1 (equivalent to querying label of every instance) achieves a significantly higher *sum* performance, which shows the biased penalty function does effectively optimize the objective metric of the weighted sum of sensitivity and specificity.

Finally, we notice that when the query ratio increases, we generally observe an improvement of the cost-sensitive classification performance by the proposed CSPAA algorithm. However, While the query ratio reaches about 1 %, the improvement tends to become saturated, which is very close to the same algorithm that queries the label of every unlabeled data. This interesting observation indicates that the proposed learning strategy is able to attain potentially the best possible predictive performance using a small amount of label data (only 1 % or even less) over the entire training data set, which can thus save a significant amount of labeling cost in a practical real-world application.

6.3.5 Evaluation on adaptive sampling factor

In the above experiments, the sampling factor δ was simply fixed to a constant. This experiment aims to examine if it is possible to further improve the proposed CSPAA approach using the adaptive sampling factor, denoted as "CSPAA(a)" for short (as discussed in the "remark" before Sect. 5.3). In this experiment, the initial value of δ is set to an extremely large value, i.e., $\delta_0 = 2^{14}$, and is updated adaptively using the proposed strategy in Sect. 5.3. To enable a fair comparison, we set appropriate parameters of the other algorithms (LEPE, CSRND and CSPAA) to make them sample the similar ratio of labeled data. Table 8 shows the experimental results for URL2 dataset, where "CSPAA" adopts the constant sampling factor. In addition, we also show the results on URL1 dataset of varied ratio of queries in Fig. 6.



(a) online cumulative average *sum* for URL1 **(b)** online cumulative average *cost* for URL1

(c) online cumulative average *sum* for URL2 **(d)** online cumulative average *cost* for URL2

Fig. 6 Evaluation of the performance with respect to varied query ratios

Table 8 Evaluation of malicious URL detection performance on “URL2”

Algorithm	Measures[a]					
	Sum (%)	Sensi (%)	Speci (%)	Accur (%)	Time (s)	Query (%)
CPA	**92.748**	86.410	99.087	98.960	37.087	100.000
	(±0.078)	(±0.154)	(±0.005)	(±0.006)		
LEPE	69.556	39.274	99.838	99.232	16.777	**0.515**
	(±1.353)	(±2.712)	(±0.015)	(±0.025)		(±0.017)
CSRND	80.724	66.578	94.871	94.588	17.039	**0.526**
	(±1.852)	(±3.855)	(±0.206)	(±0.177)		(±0.011)
CSPAA	88.756	83.628	93.883	93.781	17.260	**0.513**
	(±0.746)	(±1.701)	(±0.373)	(±0.359)		(± 0.020)
CSPAA(a)	**92.401**	89.054	95.748	95.681	18.211	**0.510**
	(±0.703)	(±1.810)	(± 0.406)	(±0.384)		(±0.014)
Algorithm	Measures[b]					
	Cost	Sensi (%)	Speci (%)	Accur (%)	Time (s)	Query (%)
CPA	**1435.806**	86.410	99.087	98.960	36.640	100.000
	(±15.494)	(±0.154)	(±0.005)	(±0.006)		
LEPE	6170.384	37.818	99.855	99.235	17.137	**0.525**
	(±152.639)	(±1.549)	(±0.009)	(±0.009)		(±0.023)
CSRND	3618.938	68.634	94.811	94.549	16.843	**0.522**
	(±466.228)	(±5.240)	(±0.598)	(±0.546)		(±0.017)
CSPAA	2265.136	82.916	94.204	94.091	17.603	**0.525**
	(±299.126)	(±3.226)	(±0.275)	(±0.249)		(±0.017)
CSPAA(a)	**1484.396**	89.498	95.508	95.448	17.917	**0.525**
	(±117.269)	(±0.831)	(±0.490)	(±0.490)		(±0.015)

We compare the Cost and Sum of our proposed algorithms with the best one among all compared algorithms with the same query rate and report the p value of the t test

[a] CSPAA versus CSRND $p < 0.0001$; CSPAA(a) versus CSRND $p < 0.0001$

[b] CSPAA versus CSRND $p = 0.0006$; CSPAA(a) versus CSRND $p < 0.0001$

Some observations can be drawn from the results. First, the CSPAA(a) algorithm using the adaptive sampling factor significantly outperforms both CSRND using the random query strategy and CSPAA using a constant sampling factor under the same query ratio. Second, we found that by querying only 0.5 % out of the entire 1-million instances, the proposed CSPAA(a) algorithm is able to achieve the best performance, which is almost the same (statistically no difference according to student t test) to the state-of-the-art cost-sensitive algorithm CPA which has to query labels for all the 1-million instances. This promising result shows that the proposed CSPAA technique is able to save a significant amount of labeling cost while maintaining the state-of-the-art performance.

6.3.6 Evaluation on efficiency and scalability

Finally, we examine the time efficiency of the proposed algorithms. The “time” columns of Tables 6, 7 and 8 show the average time costs of the proposed CSPAA algorithms on the

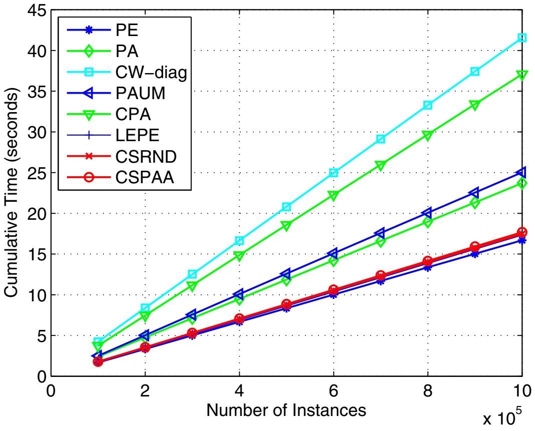


Fig. 7 Evaluation of online cumulative time cost on the "URL2" dataset

fixed query ratios. In addition to these tables, we also evaluate the scalability of the proposed algorithms, as shown in Fig. 7, which measures the online cumulative time cost of different algorithms over the number of received instances in the online malicious URL detection process.

From the results, we can see that all the proposed online learning algorithms are fairly efficient and scalable, which typically took about 20–30 s to run on the data set with 1-million instances on a single regular machine. Moreover, by examining the efficiency and scalability of the proposed CSPAA algorithms, we found that CSPAA is among the most efficient and scalable algorithms, which is at least as efficient as the other algorithms and even slightly better than some of the other algorithms. These encouraging results again validate the practical value of the proposed CSPAA algorithm for web-scale real-world applications.

7 Conclusions

This paper investigated online active learning techniques for resolving the open challenges of learning sequentially arriving data in varied settings. The proposed novel online active learning technique not only overcomes the drawback of conventional supervised passive online learning algorithms that have to query (or wait) class labels of every incoming instances, but also improves the limitation of the existing perceptron-based active learning algorithm that often wastes a lot of queried/received labeled instances that are barely classified correctly but with low prediction confidence. Specifically, we have proposed a family of passive aggressive active (PAA) learning algorithms to tackle three different kinds of online predictive tasks, including online binary classification, online multi-class classification, and cost-sensitive online classification tasks. We theoretically analyzed the mistake bounds for the proposed PAA algorithms in the three different settings, in which the bounds generally enjoy the similar bounds as those regular fully supervised Passive-Aggressive online learning algorithms when requesting class labels of every incoming instance. We conducted an extensive set of empirical studies, in which our encouraging results showed that the proposed PAA algorithms significantly outperform the baseline approaches. For future work, we plan to address more other challenges of online learning tasks, such as concept drifting issues (Minku and Yao 2012).

Appendix 1: Proof Lemma 1

Proof First of all, we need to prove the following inequality holds for every t

$$\begin{aligned}&(L_t Z_t 2\tau_t(\alpha - |p_t|) + M_t Z_t 2\tau_t(\alpha + |p_t|)\\&\quad\leq (\|\mathbf{w}_t - \alpha\mathbf{w}\|^2 - \|\mathbf{w}_{t+1} - \alpha\mathbf{w}\|^2) + \tau_t^2\|\mathbf{x}_t\|^2 + 2\alpha\tau_t\ell_t(\mathbf{w}).\end{aligned}$$

To prove that, we enumerate all the possible cases for discussions as follows:

Case 1: "$Z_t = 0$" It is clear that the inequality holds with equality since $\mathbf{w}_t = \mathbf{w}_{t+1}$ and $\tau_t = 0$.

Case 2: "$Z_t = 1$ and $M_t = 0$" The label is requested, but no mistake occurs. Sub-case 2.1: "$L_t = 0$" Since $\ell_t(\mathbf{w}_t) = 0$, $\tau_t = 0$ and $\mathbf{w}_{t+1} = \mathbf{w}_t$. Thus, the inequality holds.

Sub-case 2.2: "$L_t = 1$" Since $\ell_t(\mathbf{w}_t) > 0$, we have

$$\|\mathbf{w}_t - \alpha\mathbf{w}\|^2 - \|\mathbf{w}_{t+1} - \alpha\mathbf{w}\|^2 = -2\tau_t y_t\mathbf{w}_t \cdot \mathbf{x}_t + 2\tau_t\alpha y_t\mathbf{w} \cdot \mathbf{x}_t - \tau_t^2\|\mathbf{x}_t\|^2.$$

Since $\ell_t(\mathbf{w}) = \max(0, 1 - y_t\mathbf{w} \cdot \mathbf{x}_t) \geq 1 - y_t\mathbf{w} \cdot \mathbf{x}_t$, we have

$$\|\mathbf{w}_t - \alpha\mathbf{w}\|^2 - \|\mathbf{w}_{t+1} - \alpha\mathbf{w}\|^2 + \tau_t^2\|\mathbf{x}_t\|^2 + 2\alpha\tau_t\ell_t(\mathbf{w}) \geq 2\tau_t(\alpha - y_t\mathbf{w}_t \cdot \mathbf{x}_t).$$

Also $M_t = 0$ and $\ell_t(\mathbf{w}_t) > 0$ implies $0 \leq y_t\mathbf{w}_t \cdot \mathbf{x}_t < 1$. Thus, we have the inequality

$$||\mathbf{w}_t - \alpha\mathbf{w}\|^2 - \|\mathbf{w}_{t+1} - \alpha\mathbf{w}\|^2 + \tau_t^2\|\mathbf{x}_t\|^2 + 2\alpha\tau_t\ell_t(\mathbf{w}) \geq 2\tau_t(\alpha - |p_t|).$$

Case 3: "$Z_t = 1$ and $M_t = 1$" It means the label is requested and a mistake occurs, but $L_t = 0$. Similarly, we have

$$\|\mathbf{w}_t - \alpha\mathbf{w}\|^2 - \|\mathbf{w}_{t+1} - \alpha\mathbf{w}\|^2 + \tau_t^2\|\mathbf{x}_t\|^2 + 2\alpha\tau_t\ell_t(\mathbf{w}) \geq 2\tau_t(\alpha - y_t\mathbf{w}_t \cdot \mathbf{x}_t).$$

Since $M_t = 1$ implies $y_t\mathbf{w}_t \cdot \mathbf{x}_t \leq 0$ and $-y_t\mathbf{w}_t \cdot \mathbf{x}_t = |p_t|$, we have

$$\|\mathbf{w}_t - \alpha\mathbf{w}\|^2 - \|\mathbf{w}_{t+1} - \alpha\mathbf{w}\|^2 + \tau_t^2\|\mathbf{x}_t\|^2 + 2\alpha\tau_t\ell_t(\mathbf{w}) \geq 2\tau_t(\alpha + |p_t|).$$

Combining the above cases for all $t = 1, \ldots, T$, we have

$$\begin{aligned}&\sum_{t=1}^{T}(L_t Z_t 2\tau_t(\alpha - |p_t|) + M_t Z_t 2\tau_t(\alpha + |p_t|)\\&\quad\leq \sum_{t=1}^{T}(\|\mathbf{w}_t - \alpha\mathbf{w}\|^2 - \|\mathbf{w}_{t+1} - \alpha\mathbf{w}\|^2) + \tau_t^2\|\mathbf{x}_t\|^2 + 2\alpha\tau_t\ell_t(\mathbf{w})\\&\quad\leq \alpha^2\|\mathbf{w}\|^2 + \sum_{t=1}^{T}\tau_t^2\|\mathbf{x}_t\|^2 + \sum_{t=1}^{T}2\alpha\tau_t\ell_t(\mathbf{w})\ .\end{aligned}$$

□

Appendix 2: Lemma 2

Similarly to the binary section, before presenting the mistake bounds for multi-class classification, we begin by presenting a technical lemma which would facilitate the proof of Theorem 5.

Lemma 2 *Let* $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_T, y_T)$ *be a sequence of input instances, where* $\mathbf{x}_t \in \mathbb{R}^d$ *and* $y_t \in Y = \{1, \ldots, k\}$ *for all* t. *Let* τ_t *be the stepsize parameter for either of the three MPAA variants as given in Eq.* (4). *For any* $\alpha > 0$, *the following bound holds for any classifier* $\overline{\mathbf{w}}$ *made up of* k *vectors* $\mathbf{w}^r \in \mathbb{R}^d$, $r \in Y$.

$$\sum_{t=1}^{T} Z_t 2\tau_t \big[L_t(\alpha - p_t) + M_t(\alpha + p_t)\big]$$
$$\leq \alpha^2 \sum_{r=1}^{k} \|\mathbf{w}^r\|^2 + \sum_{t=1}^{T} 2\tau_t^2 \|\mathbf{x}_t\|^2 + \sum_{t=1}^{T} 2\alpha\tau_t \ell_t(\overline{\mathbf{w}}),$$

where $M_t = \mathbb{I}_{(t\in\mathcal{M})}$, $L_t = \mathbb{I}_{(t\in\mathcal{L})}$, $\mathbb{I}$ *is an indicator function.*

Proof First of all, we need to prove the following inequality holds for every t

$$\begin{aligned}&(L_t Z_t 2\tau_t(\alpha - p_t) + M_t Z_t 2\tau_t(\alpha + p_t)\\&\quad\leq \sum_{r=1}^{k} (\|\mathbf{w}_t^r - \alpha\mathbf{w}^r\|^2 - \|\mathbf{w}_{t+1}^r - \alpha\mathbf{w}^r\|^2) + 2\tau_t^2\|\mathbf{x}_t\|^2 + 2\alpha\tau_t\ell_t(\overline{\mathbf{w}}).\end{aligned} \tag{12}$$

To prove that, we should enumerate all the possible cases for discussions. For conciseness, we only prove the two cases: $M_t = 1$ and $L_t = 1$ and omit the others since they are similar to that in Lemma 1.

First, when $L_t = 1$, Since $\ell_t(\mathbf{w}_t) > 0$, we have

$$\begin{aligned}&\sum_{r=1}^{k} (\|\mathbf{w}_t^r - \alpha\mathbf{w}^r\|^2 - \|\mathbf{w}_{t+1}^r - \alpha\mathbf{w}^r\|^2)\\&=\|\mathbf{w}_t^{y_t} - \alpha\mathbf{w}^{y_t}\|^2 - \|\mathbf{w}_t^{y_t} - \alpha\mathbf{w}^{y_t} + \tau_t\mathbf{x}_t\|^2\\&\quad + \|\mathbf{w}_t^{s_t} - \alpha\mathbf{w}^{s_t}\|^2 - \|\mathbf{w}_t^{s_t} - \alpha\mathbf{w}^{s_t} - \tau_t\mathbf{x}_t\|^2\\&= -2\tau_t^2||\mathbf{x}_t||^2 + 2\tau_t(\mathbf{x}_t \cdot \mathbf{w}_t^{s_t} - \mathbf{x}_t \cdot \mathbf{w}_t^{y_t}) + 2\alpha\tau_t(\mathbf{x}_t \cdot \mathbf{w}^{y_t} - \mathbf{x}_t \cdot \mathbf{w}^{s_t}).\end{aligned} \tag{13}$$

Note that s_t is the highest ranked irrelevant label with regard to the classifier $\overline{\mathbf{w}}_t$, not $\overline{\mathbf{w}}$. In a correct prediction, $\mathbf{x}_t \cdot \mathbf{w}_t^{s_t} - \mathbf{x}_t \cdot \mathbf{w}_t^{y_t} = -p_t$ and $\mathbf{x}_t \cdot \mathbf{w}^{y_t} - \mathbf{x}_t \cdot \mathbf{w}^{s_t} \geq \gamma_{t,\overline{\mathbf{w}}}$, where $\gamma_{t,\overline{\mathbf{w}}}$ is used to denote the margin of instance $\mathbf{x}_t$ with regard to the classifier $\overline{\mathbf{w}}$ (since we used γ_t to denote the margin of $\mathbf{x}_t$ with regards to $\overline{\mathbf{w}}_t$).

According to the definition of hinge loss, we have $\ell_t(\overline{\mathbf{w}}) \geq 1 - \gamma_{t,\overline{\mathbf{w}}}$ and thus $\gamma_{t,\overline{\mathbf{w}}} \geq 1 - \ell_t(\overline{\mathbf{w}})$. Combining the above facts with (13) yields to

$$\begin{aligned}\sum_{r=1}^{k} (\|\mathbf{w}_t^r - \alpha\mathbf{w}^r\|^2 - \|\mathbf{w}_{t+1}^r - \alpha\mathbf{w}^r\|^2) &\geq -2\tau_t^2||\mathbf{x}_t||^2 - 2\tau_t p_t + 2\alpha\tau_t(1 - \ell_t(\overline{\mathbf{w}}))\\&= -2\tau_t^2||\mathbf{x}_t||^2 + 2\tau_t(\alpha - p_t) - 2\alpha\tau_t\ell(\overline{\mathbf{w}}).\end{aligned}$$

We write the above formula as:

$$2\tau_t(\alpha - p_t) \leq \sum_{r=1}^{k} (\|\mathbf{w}_t^r - \alpha\mathbf{w}^r\|^2 - \|\mathbf{w}_{t+1}^r - \alpha\mathbf{w}^r\|^2) + 2\tau_t^2||\mathbf{x}_t||^2 + 2\alpha\tau_t\ell_t(\overline{\mathbf{w}}).$$

Second, when $M_t = 1$, i.e. incorrect prediction, $s_t = \hat{y}_t$ and thus

$$\mathbf{x}_t \cdot \mathbf{w}_t^{s_t} - \mathbf{x}_t \cdot \mathbf{w}_t^{y_t} \geq p_t.$$

Combining this fact and (13), we get

$$2\tau_t(\alpha + p_t) \le \sum_{r=1}^{k}(\|\mathbf{w}_t^r - \alpha\mathbf{w}^r\|^2 - \|\mathbf{w}_{t+1}^r - \alpha\mathbf{w}^r\|^2) + 2\tau_t^2||\mathbf{x}_t||^2 + 2\alpha\tau_t\ell_t(\overline{\mathbf{w}}).$$

Considering all cases, we can finally prove (12) is correct. This proof is finished by summing (12) over all iterations $t = 1, \ldots, T$. □

Appendix 3: Proof of Theorem 5

Proof Since $\ell_t(\overline{\mathbf{w}}) = 0, \forall t \in [T]$, according to Lemma 2, we have

$$\begin{aligned}\alpha^2\sum_{r=1}^{k}\|\mathbf{w}^r\|^2 &\ge \sum_{t=1}^{T} Z_t 2\tau_t\big[L_t(\alpha - p_t) + M_t(\alpha + p_t)\big] - \sum_{t=1}^{T} 2\tau_t^2\|\mathbf{x}_t\|^2\\ &= \sum_{t=1}^{T} Z_t 2\tau_t\big[L_t(\alpha - p_t - \tau_t\|\mathbf{x}_t\|^2) + M_t(\alpha + p_t - \tau_t\|\mathbf{x}_t\|^2)\big]\\ &= \sum_{t=1}^{T} Z_t 2\tau_t\big[L_t(\alpha - p_t - \frac{\ell_t(\overline{\mathbf{w}}_t)}{2}) + M_t(\alpha + p_t - \frac{\ell_t(\overline{\mathbf{w}}_t)}{2})\big]\\ &= \sum_{t=1}^{T} L_t Z_t 2\tau_t\left(\alpha - \frac{1+p_t}{2}\right) + \sum_{t=1}^{T} M_t Z_t 2\tau_t\left(\alpha - \frac{1-p_t}{2}\right).\end{aligned}$$

Plugging $\alpha = \frac{\delta+1}{2}, \delta \ge 1$ into the above inequality results in

$$\left(\frac{1+\delta}{2}\right)^2\sum_{r=1}^{k}\|\mathbf{w}^r\|^2 \ge \sum_{t=1}^{T} M_t Z_t \tau_t(\delta + p_t),$$

In addition, combining the fact $\tau_t = \ell_t(\overline{\mathbf{w}}_t)/2\|\mathbf{x}_t\|^2 \ge \ell_t(\overline{\mathbf{w}}_t)/2R^2$ with the above inequality concludes:

$$\left(\frac{1+\delta}{2}\right)^2\sum_{r=1}^{k}\|\mathbf{w}^r\|^2 \ge \frac{1}{2R^2}\sum_{t=1}^{T} M_t Z_t \ell_t(\overline{\mathbf{w}}_t)(\delta + p_t).$$

Taking expectation with the above inequality results in

$$\mathbb{E}\left[\frac{1}{2R^2}\sum_{t=1}^{T} M_t\ell_t(\overline{\mathbf{w}}_t)Z_t(\delta + p_t)\right] = \frac{1}{2R^2}\mathbb{E}\left[\delta\sum_{t=1}^{T} M_t\ell_t(\overline{\mathbf{w}}_t)\right] \le \left(\frac{1+\delta}{2}\right)^2\sum_{r=1}^{k}\|\mathbf{w}^r\|^2.$$

□

Appendix 4: Proof of Proposition 1

Proof First of all, by analyzing the weighted sum in (6), we can derive:

$$\begin{aligned} sum &= \eta_p \frac{T_p - M_p}{T_p} + \eta_n \frac{T_n - M_n}{T_n} \\ &= 1 - \frac{\eta_n}{T_n}\left[\frac{\eta_p T_n}{\eta_n T_p} \sum_{y_t=+1} \mathbb{I}_{(y_t \mathbf{w}\cdot\mathbf{x}_t<0)} + \sum_{y_t=-1} \mathbb{I}_{(y_t \mathbf{w}\cdot\mathbf{x}_t<0)}\right]. \end{aligned}$$

Thus, maximizing sum is equivalent to minimizing

$$\frac{\eta_p T_n}{\eta_n T_p} \sum_{y_t=+1} \mathbb{I}_{(y_t \mathbf{w}\cdot\mathbf{x}_t<0)} + \sum_{y_t=-1} \mathbb{I}_{(y_t \mathbf{w}\cdot\mathbf{x}_t<0)}.$$

Second, by analyzing the weighted cost in (7), we can also derive:

$$cost = c_p M_p + c_n M_n = c_n\left[\frac{c_p}{c_n} \sum_{y_t=+1} \mathbb{I}_{(y_t \mathbf{w}\cdot\mathbf{x}_t<0)} + \sum_{y_t=-1} \mathbb{I}_{(y_t \mathbf{w}\cdot\mathbf{x}_t<0)}\right].$$

Thus, minimizing $cost$ is equivalent to minimizing

$$\frac{c_p}{c_n} \sum_{y_t=+1} \mathbb{I}_{(y_t \mathbf{w}\cdot\mathbf{x}_t<0)} + \sum_{y_t=-1} \mathbb{I}_{(y_t \mathbf{w}\cdot\mathbf{x}_t<0)}.$$

Thus, the proposition holds by setting $\rho = \frac{\eta_p T_n}{\eta_n T_p}$ for sum, and $\rho = \frac{c_p}{c_n}$ for cost. □

Appendix 5: Proof of Theorem 6

Proof As proven in Theorem 2,

$$\begin{aligned} &\alpha^2\|\mathbf{w}\|^2 + \sum_{t=1}^{T} 2\alpha\tau_t \ell_t(\mathbf{w}) \\ &\geq \sum_{t=1}^{T} L_t Z_t 2\tau_t\left(\alpha - \frac{1+|p_t|}{2}\right) + \sum_{t=1}^{T} M_t Z_t 2\tau_t\left(\alpha - \frac{1-|p_t|}{2}\right). \end{aligned}$$

Plugging $\alpha = \frac{1+\delta}{2}$, $\delta \geq 1$ into the above inequality results in

$$\left(\frac{1+\delta}{2}\right)^2 \|\mathbf{w}\|^2 + \sum_{t=1}^{T}(1+\delta)\tau_t\ell_t(\mathbf{w}) \geq \sum_{t=1}^{T} M_t Z_t \tau_t(\delta + |p_t|),$$

since when $L_t = 1$, $|p_t| \in [0, 1)$, $(\alpha - \frac{1+|p_t|}{2}) = \frac{\delta-|p_t|}{2} > 0$, and $(\alpha - \frac{1-|p_t|}{2}) = \frac{\delta+|p_t|}{2}$. Furthermore, because when $M_t = 1$, $\tau_t = \min\{C, \frac{\ell_t(\mathbf{w}_t)}{\|\mathbf{x}_t\|^2}\} \geq \min\{C, \rho_t\} = \rho_t$, and $\tau_t \leq C$, the above inequality implies:

$$\left(\frac{1+\delta}{2}\right)^2 \|\mathbf{w}\|^2 + \sum_{t=1}^{T}(1+\delta)C\ell_t(\mathbf{w}) \geq \sum_{t=1}^{T} \rho_t M_t Z_t(\delta + |p_t|),$$

Taking expectation with the above equality and re-arranging the result conclude the theorem, since

$$\mathbb{E}\sum_{t=1}^{T}\rho_t M_t Z_t(\delta+|p_t|)=\mathbb{E}\sum_{t=1}^{T}\rho_t M_t(\delta+|p_t|)\mathbb{E}_t Z_t=\delta\mathbb{E}\sum_{t=1}^{T}\rho_t M_t.$$

□

Appendix 6: Proof of Theorem 7

Proof Following the condition that $\rho=\frac{\eta_p T_n}{\eta_n T_p}\geq 1$ and the result of Theorem 6, we have

$$\begin{aligned}\frac{1}{\delta}\left\{\left(\frac{1+\delta}{2}\right)^2\|\mathbf{w}\|^2+\sum_{t=1}^{T}(1+\delta)C\ell_t(\mathbf{w})\right\}&\geq\big(\rho\mathbb{E}M_p+\mathbb{E}M_n\big)\\&=\left[\left(\frac{\eta_p T_n}{\eta_n T_p}\right)\mathbb{E}M_p+\mathbb{E}M_n\right]=\frac{T_n}{\eta_n}\left[\eta_p\left(\frac{\mathbb{E}M_p}{T_p}\right)+\eta_n\frac{\mathbb{E}M_n}{T_n}\right]\\&=\frac{T_n}{\eta_n}\big(\eta_p(1-\mathbb{E}sen)+\eta_n(1-\mathbb{E}spe)\big)=\frac{T_n}{\eta_n}(1-\mathbb{E}[sum]).\end{aligned}$$

Rearranging the above inequality leads to the conclusion:

$$\mathbb{E}[sum]\geq 1-\frac{\eta_n}{T_n}\frac{1}{\delta}\left\{\left(\frac{1+\delta}{2}\right)^2\|\mathbf{w}\|^2+\sum_{t=1}^{T}(1+\delta)C\ell_t(\mathbf{w})\right\}.$$

□

Appendix 7: Proof of Theorem 8

Proof Following the result of Theorem 6, we have

$$\begin{aligned}\frac{1}{\delta}\left\{\left(\frac{1+\delta}{2}\right)^2\|\mathbf{w}\|^2+\sum_{t=1}^{T}(1+\delta)C\ell_t(\mathbf{w})\right\}&\geq(\mathbb{E}M_p(\rho)+\mathbb{E}M_n)\\&=\left(\mathbb{E}M_p\left(\frac{c_p}{c_n}\right)+\mathbb{E}M_n\right)=\frac{1}{c_n}\mathbb{E}[cost].\end{aligned}$$

Rearranging the above inequality concludes the theorem. □

References

Akbani, R., Kwek, S., & Japkowicz, N. (2004). Applying support vector machines to imbalanced datasets. *Machine learning: ECML 2004* (pp. 39–50). Springer.

Balcan, M.-F., Beygelzimer, A., & Langford, J. (2006). Agnostic active learning. *ICML* (pp. 65–72).

Balcan, M.-F., Broder, A., & Zhang, T. (2007). Margin based active learning. *COLT* (pp. 35–50).

Brodersen, K. H., Ong, C. S., Stephan, K. E., & Buhmann, J. M. (2010). The balanced accuracy and its posterior distribution, *2010 20th international conference on pattern recognition (ICPR)*. IEEE (pp. 3121–3124).

Castro, R. M., & Nowak, R. D. (2007). Minimax bounds for active learning. *COLT* (pp. 151–156).

Cavallanti, G., Cesa-Bianchi, N., & Gentile, C. (2009). Linear classification and selective sampling under low noise conditions. *Advances in Neural Information Processing Systems* (pp. 249–256).

Cesa-Bianchi, N., Conconi, A., & Gentile, C. (2004). On the generalization ability of on-line learning algorithms. *IEEE Transactions on Information Theory*, *50*(9), 2050–2057.

Cesa-Bianchi, N., Conconi, A., & Gentile, C. (2005). A second-order perceptron algorithm. *SIAM Journal on Computing*, *34*(3), 640–668.

Cesa-Bianchi, N., Gentile, C., & Orabona, F. (2009). Robust bounds for classification via selective sampling. In *Proceedings of the 26th annual international conference on machine learning*. ACM (pp. 121–128).

Cesa-Bianchi, N., Gentile, C., & Zaniboni, L. (2006). Worst-case analysis of selective sampling for linear classification. *The Journal of Machine Learning Research*, *7*, 1205–1230.

Cesa-Bianchi, N., & Lugosi, G. (2006). *Prediction, learning, and games*. Cambridge: Cambridge University Press.

Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., & Singer, Y. (2006). Online passive-aggressive algorithms. *JMLR*, *7*, 551–585.

Crammer, K., Dredze, M., & Pereira, F. (2008). Exact convex confidence-weighted learning. *Advances in Neural Information Processing Systems (NIPS)* (pp. 345–352).

Crammer, K., Kulesza, A., & Dredze, M. (2009). Adaptive regularization of weight vectors. *Advances in Neural Information Processing Systems (NIPS)* (pp. 414–422).

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

Dasgupta, S., Kalai, A. T., & Monteleoni, C. (2009). Analysis of perceptron-based active learning. *JMLR*, *10*, 281–299.

Dekel, O., Gentile, C., & Sridharan, K. (2010). Robust selective sampling from single and multiple teachers. *COLT* (pp. 346–358).

Elkan, C. (2001). The foundations of cost-sensitive learning. *International joint conference on artificial intelligence* (Vol. 17, Citeseer, pp. 973–978).

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

Freund, Y., Seung, H. S., Shamir, E., & Tishby, N. (1997). Selective sampling using the query by committee algorithm. *Machine Learning*, *28*(2–3), 133–168.

Gentile, C. (2001). A new approximate maximal margin classification algorithm. *Journal of Machine Learning Research*, *2*, 213–242.

Hoi, S. C. H., Wang, J., & Zhao, P. (2014). LIBOL: A library for online learning algorithms. *Journal of Machine Learning Research*, *15*(1):495–499. http://LIBOL.stevenhoi.org/.

Jiawei, H., & Kamber, M. (2001). *Data mining: Concepts and techniques* (Vol. 5). San Francisco, CA, itd: Morgan Kaufmann.

Kivinen, J., Smola, A. J., & Williamson, R. C. (2001). Online learning with kernels. *NIPS* (pp. 785–792).

Krauth, W., & Mézard, M. (1987). Learning algorithms with optimal stability in neural networks. *Journal of Physics A: Mathematical and General*, *20*(11), L745.

Li, S., & Tsang, I. W. (2011). Maximum margin/volume outlier detection. *ICTAI* (pp. 385–392).

Li, Y., & Long, P. M. (1999). The relaxed online maximum margin algorithm. *Advances in Neural Information Processing Systems (NIPS)* (pp. 498–504).

Li, Y., Zaragoza, H., Herbrich, R., Shawe-Taylor, J., & Kandola, J. S. (2002). The perceptron algorithm with uneven margins. In *Proceedings of the nineteenth international conference on machine learning (ICML2002)* (pp. 379–386).

Liu, X.-Y., & Zhou, Z.-H. (2006). The influence of class imbalance on cost-sensitive learning: An empirical study. *2006. ICDM'06 sixth international conference on data mining*. IEEE (pp. 970–974).

Lu, J., Zhao, P., & Hoi, S. C. (2014). Online passive aggressive active learning and its applications. In *Proceedings of the Sixth Asian Conference on Machine Learning* (ACML) (pp. 266–282).

Ma, J., Saul, L. K., Savage, S., & Voelker, G. M. (2009a). Beyond blacklists: Learning to detect malicious web sites from suspicious urls. *KDD* (pp. 1245–1254).

Ma, J., Saul, L. K., Savage, S., & Voelker, G. M. (2009b). Identifying suspicious urls: An application of large-scale online learning. *ICML* (p. 86).

McCallum, A., & Nigam, K. (1998). Employing em and pool-based active learning for text classification. ICML, San Francisco, CA (pp. 350–358).

Minku, L. L., & Yao, X. (2012). DDD: A new ensemble approach for dealing with concept drift. *IEEE Transactions on Knowledge and Data Engineering*, *24*(4), 619–633.

Orabona, F., & Cesa-Bianchi, N. (2011). Better algorithms for selective sampling. *International conference on machine learning*, Omnipress (pp. 433–440).

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

Sculley, D. (2007). Online active learning methods for fast label-efficient spam filtering. *CEAS*.

Tong, S., & Koller, D. (2002). Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, *2*, 45–66.

Wang, D., Irani, D., & Pu, C. (2012). Evolutionary study of web spam: Webb spam corpus 2011 versus webb spam corpus 2006. *CollaborateCom* (pp. 40–49).

Wang, J., Zhao, P., & Hoi, S. C. (2014). Cost-sensitive online classification. *IEEE Transactions on Knowledge and Data Engineering*, *26*(10), 2425–2438.

Wang, J., Zhao, P., & Hoi, S. C. H. (2012). Exact soft confidence-weighted learning. In *Proceedings of the 29th international conference on machine learning, ICML 2012, Edinburgh, Scotland, UK, June 26-July 1, 2012*. http://icml.cc/discuss/2012/86.html.

Zadrozny, B., Langford, J., & Abe, N. (2003). Cost-sensitive learning by cost-proportionate example weighting. *Third IEEE international conference on data mining, 2003. ICDM 2003*. IEEE (pp. 435–442).

Zhao, P., & Hoi, S. C. H. (2010). OTL: A framework of online transfer learning. In *Proceedings of the 27th international conference on machine learning (ICML-10), June 21–24, 2010, Haifa, Israel*, pp. 1231–1238. http://www.icml2010.org/papers/219.pdf.

Zhao, P., & Hoi, S. C. (2013). Cost-sensitive online active learning with application to malicious URL detection. In *Proceedings of the 19th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 919–927). ACM.

Zhao, P., Hoi, S. C. H., & Jin, R. (2011). Double updating online learning. *Journal of Machine Learning Research*, *12*, 1587–1615.

Zhao, P., Jin, R., Yang, T., & Hoi, S. C. (2011). Online auc maximization. In *Proceedings of the 28th international conference on machine learning (ICML-11)* (pp. 233–240).

Zhu, X., & Wu, X. (2006). Class noise handling for effective cost-sensitive learning by cost-guided iterative classification filtering. *IEEE Transactions on Knowledge and Data Engineering*, *18*(10), 1435–1440.

Zhu, X., Zhang, P., Lin, X., & Shi, Y. (2007). Active learning from data streams. *Seventh IEEE international conference on data mining, 2007. ICDM 2007*. IEEE (pp. 757–762).

Zhu, X., Zhang, P., Lin, X., & Shi, Y. (2010). Active learning from stream data using optimal weight classifier ensemble. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, *40*(6), 1607–1621.

Žliobaitė, I., Bifet, A., Pfahringer, B., & Holmes, G. (2011). Active learning with evolving streaming data. *Machine learning and knowledge discovery in databases*. Springer.

Zliobaite, I., Bifet, A., Pfahringer, B., & Holmes, G. (2014). Active learning with drifting streaming data. *IEEE Transactions on Neural Networks and Learning Systems*, *25*(1), 27–39.