Machine Learning (2022) 111:2323–2348
https://doi.org/10.1007/s10994-022-06133-8

Online active classification via margin-based and feature-based label queries

Tingting Zhai[1] · **Frédéric Koriche**[2] · **Yang Gao**[3] · **Junwu Zhu**[1] · **Bin Li**[1]

Received: 26 January 2021 / Revised: 22 December 2021 / Accepted: 6 February 2022 /
Published online: 11 March 2022

Abstract
In the paradigm of *online active classification*, the learner not only has to predict the label of each incoming instance, but also must decide whether the true label of that instance should be supplied, or not. The overall goal is to minimize the number of prediction mistakes with few label queries. In this paper, we focus on a novel framework for online active learning, with the aim of handling high dimensional classification problems. The key component of our framework is to exploit both the *margin-based predictive uncertainty* and the *feature-based discriminative information* of the current instance, in order to determine whether it should be labeled. Based on this labeling strategy, we propose several online active learning algorithms, for both binary classification tasks and multiclass ones. For these algorithms, which use adaptive subgradient methods for updating their linear model, expected mistake bounds are provided. Experiments on high-dimensional (binary and multiclass) classification datasets reveal the benefit of our label query strategy, and show the superiority of our algorithms over the existing ones.

Keywords Online active learning · High dimensional data · Multiclass active learning · Adaptive subgradient methods

1 Introduction

Online learning is well-studied framework in Machine Learning, with both theoretical and practical appeals (Shalev-Shwartz, 2012). For large-scale and possibly streaming applications, online learning has received widespread attention, owing to its efficiency and scalability by handling instances one-by-one. Conceptually, online learning for classification can be viewed as a sequential process, involving a learner and its environment. At each

Editors: Annalisa Appice, Sergio Escalera, Jose A. Gamez, Heike Trautmann.

✉ Tingting Zhai
zhtt@yzu.edu.cn

[1] College of Information Engineering, Yangzhou University, Yangzhou, China

[2] Center of Research in Information in Lens, Université d'Artois, Lens, France

[3] State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China

round t, the learner first receives an instance $\boldsymbol{x}_t$, from which it is required to predict a class or label according to its current predictor $\boldsymbol{w}_t$. Once the learner has committed to its prediction, say $\hat{y}_t$, the environment reveals the true label y_t, and the learner incurs a loss which assesses the discrepancy between the prediction $\hat{y}_t$ and the response y_t. Before proceeding to the next round, the learner is allowed to choose a new predictor $\boldsymbol{w}_{t+1}$ in the hope of improving its predictive performance for the subsequent rounds. In the past decades, various online learning algorithms have been proposed, including online first-order (Crammer et al. 2006; Shalev-Shwartz et al., 2011; Zhai et al., 2019; Zinkevich, 2003) and second-order methods (Crammer et al., 2012; Hazan et al., 2007; Luo et al., 2016), online kernel (Lu et al., 2016a; Song et al., 2017) and multiple kernel methods (Hoi et al., 2013), online ensemble learning methods (Sun et al., 2016; Zhai et al., 2017), and so on.

According to the above protocol, online learning is a *fully supervised* learning process, in which the label of each incoming instance is provided by the environment. Although this protocol has been successful for handling large, fully labeled data streams, it is ill-suited for dealing with applications where labels are scarce or expensive to obtain. Consider for example the task of classifying web pages according to a set of predefined topics. Collecting and encoding web pages as vectorized instances is a fairly automated process, but assigning them a topic often requires time-consuming and costly human expertise. Similarly, in personalized anti-spam filtering, various techniques are known to encode incoming messages as feature vectors, but it is unreasonable to assume that a user will label every message as a "spam" or a "ham". For such applications, a natural question arises: can we achieve strong online classification performance while using only few labeled instances?

Online *active* learning has recently come up as a promising approach for handling this issue. As usual, the learner starts each round t by making a prediction $\hat{y}_t$ for an incoming instance $\boldsymbol{x}_t$ using its model $\boldsymbol{w}_t$. But the key difference with passive online learning lies at the end of the round: in the active setting, the learner has to decide whether the true label y_t of the instance $\boldsymbol{x}_t$ should be supplied, or not. If y_t is queried, then the complete example $(\boldsymbol{x}_t, y_t)$ is obtained and the learner uses the example to derive a new predictor $\boldsymbol{w}_{t+1}$. Otherwise, the current predictor $\boldsymbol{w}_t$ is left unchanged.

In the literature, there exists another line of active learning, namely, offline (or pool-based) active learning (Lughofer, 2017; Settles, 2009), which assumes that a pool of unlabeled instances is available before learning, and query decisions are made by evaluating the whole pool of unlabeled instances. Various labeling strategies have been developed in this offline scenario. Margin-based methods (Awasthi et al., 2015; Balcan and Long, 2013; Zhang, 2018) query instances which are close to the estimated decision boundary. Disagreement-based methods (Golovin et al., 2010; Hanneke, 2014; Tosh and Dasgupta, 2017) maintain a set of hypotheses that are consistent with the currently labeled instances, and query the unlabeled instances about which those hypotheses most disagree. Multi-criterion methods (Demir and Bruzzone, 2014; Du et al., 2017; Huang et al., 2014; Wang and Ye, 2015) combine multiple criteria for assessing the value of an unlabeled instance and querying the most valuable instances.

Contrastingly, online active learning is more suited for large-scale and streaming data, by handling instances one-by-one. In this protocol, the query strategy is applied on each incoming, unclassified instance. Based on this online active learning framework, several perceptron-based active learning algorithms that rely on a *margin-based* query strategy have been proposed (Cesa-Bianchi et al., 2006). At each round t, the learner draws a random variable $Z_t \in \{0, 1\}$ from a Bernoulli distribution with parameter $b/(b + p_t)$, where $p_t = |\boldsymbol{w}_t^\top \boldsymbol{x}_t|$ is the prediction margin of the instance $\boldsymbol{x}_t$, and $b > 0$ is a predefined hyperparameter used to control the probability of asking the label y_t of $\boldsymbol{x}_t$. This label is revealed

only when $Z_t = 1$ and, in that case, the predictor $\boldsymbol{w}_t$ is updated according to a first-order or second-order perceptron rule. The margin-based label query was also advocated for the active versions of the Winnow algorithm(Cesa-Bianchi et al., 2006) and the Passive-Aggressive algorithm (Lu et al., 2016b). More recently, Hao et al.(Hao et al., 2018) have proposed a new algorithm, called Second-order Online Active Learning (SOAL), that exploits both the prediction margin and the margin variance for asking label queries, and which updates the predictor using a variant of the Adaptive Regularization Of Weights method (Crammer et al., 2013).

In practice, some second-order online active learning methods have shown better performance than the first-order methods (Cesa-Bianchi et al., 2006; Hao et al., 2018). In doing so, these second-order methods maintain a correlation matrix and use the matrix to update the online predictor. In presence of high-dimensional data, maintaining and using a *full* correlation matrix is prohibitive in time and space. Although Hao et al.(Hao et al., 2018) has realized this problem and has extended SOAL to use the *diagonal* correlation matrix, the empirical and theoretical analyses in the paper are only for the full matrix version of SOAL, and not for the diagonal matrix version of SOAL. On the other hand, first-order methods are more efficient in time and space than second-order methods in handling high-dimensional data, but may suffer from two critical limitations. First, their updating rules treat all dimensions of features equally and update each dimension in the same learning rate, which is deficient given that one feature may be seen hundreds of times, while another feature may be seen only once. Second, their margin-based label query strategy ignores the *feature-based discriminative information* of instances. At this point, it is well-known that infrequently occurring features are highly informative and discriminative (Crammer et al., 2012; Duchi et al., 2011) and should be taken more notice when they occur. Therefore, in the label query, when instances including such infrequent features appear, they should be given more chances to be queried. In summary, existing research on effective and efficient online active learning for high-dimensional data is still insufficient.

Furthermore, most of the aforementioned methods are designed only for binary classification tasks and how to generalize them to the multiclass scenario is left unknown. Indeed, to the best of our knowledge, there is only one research paper handling the multiclass problem in the online active learning setting. In (Lu et al., 2016b), Lu et al. extend their Passive-Aggressive Active learning algorithms (PAA) for binary classification to the multiclass setting and propose the Multiclass PAA (MPAA). MPAA uses the Multi-prototype method (Crammer et al., 2006) together with the Multiclass Passive-Aggressive algorithms for constructing and updating the multiclass classifier online, and also relies on a *multiclass margin-based* query strategy to query labels. In the query strategy, a decision variable $Z_t \in \{0, 1\}$ is drawn according to the Bernoulli distribution with parameter $b/(b + p_t)$, where p_t is a quantity used to approximate the true multiclass predictive margin. MPAA also suffer from two limitations. First, all dimensions of features are updated in the same learning rate. Second, the query strategy also ignores the feature-based discriminative information of instances.

In this study, we focus on novel online active classification methods, which can handle high dimensional data effectively and efficiently and present good extensions to the multiclass classification tasks. Our contributions are threefold:

1. Two novel online active learning algorithms for binary classification are proposed, which use the adaptive subgradient methods (Duchi et al. 2011) to update the online learner when the labels of instances are revealed and which exploit not only the margin-based

predictive uncertainty of instances, but also the feature-based discriminative information of instances to identify critical instances to query. Our updating rules can endow different dimensions of features with different learning rates by using a diagonal correlation matrix. Our label query strategy can discover instances that significantly improve the online predictive performance. In light of the above algorithmic design, the proposed methods can handle high dimensional data effectively and efficiently. Both algorithms have been extended to the multiclass scenario.

2. Expected mistake bounds for our proposed algorithms are provided and analyzed. The bounds reveal that when the label query ratio is larger than a certain value, our active learning algorithms are asymptotically comparable to the best fixed fully supervised classifier chosen in hindsight.
3. An ablation study on six high dimensional binary classification datasets show the superiority of our label query strategy. Comparative experiments also indicate that, at extensive label query ratios, our algorithms outperform (in terms of online F1-measure) existing online active learning methods. Furthermore, experiments on six multiclass classification datasets also show the advantage of our multiclass active learning algorithms.

The paper is organized as follows. Section 2 provides the notation used throughout this paper. Our proposed active learning algorithms for binary classification are presented and analyzed in Sect. 3. Further, both algorithms are extended to the multiclass classification tasks in Sect. 4. Experimental comparisons and analyses are provided in Sect. 5. Finally, Sect. 6 concludes this study with some perspectives of further research.

2 Notation

For a positive integer T, let $[T]$ denote the set $\{1, 2, \cdots T\}$. For an event E, we denote by $\mathbb{1}[E]$ the indicator function in $\{0, 1\}$ of E, namely, $\mathbb{1}[E] = 1$ if E happens and $\mathbb{1}[E] = 0$, otherwise. For a scalar a, we use $\text{sgn}(a)$ to denote the sign in $\{-1, +1\}$ of a. The ith element of a vector $\boldsymbol{x}_t$ is denoted by $x_{t,i}$. We use $\boldsymbol{a}_{1:t} = [a_1, \cdots, a_t]$ to denote the (row) vector representation of a scalar sequence $\{a_i\}_{i=1}^t$. By extension, we use $\boldsymbol{G}_{1:t} = [\boldsymbol{g}_1, \cdots, \boldsymbol{g}_t]$ to denote the $d \times t$ matrix representation of a sequence $\{\boldsymbol{g}_i\}_{i=1}^t$ of (column) vectors in $\mathbb{R}^d$, and here we use $\boldsymbol{G}_{1:t,i}$ to denote the ith row of $\boldsymbol{G}_{1:t}$. The inner product of two vectors $\boldsymbol{w}$ and $\boldsymbol{v}$ is denoted by $\boldsymbol{w}^\top \boldsymbol{v}$, and for any $p \in [1, \infty]$, we use $\|\boldsymbol{w}\|_p$ to denote the ℓ_p norm of $\boldsymbol{w}$. For a vector $\boldsymbol{v}$, we denote by $\text{diag}(\boldsymbol{v})$ the diagonal matrix with elements of $\boldsymbol{v}$ on the diagonal line, and we use $\|\boldsymbol{v}\|_{\boldsymbol{A}}$ to denote the Mahalanobis norm of $\boldsymbol{v}$ with respect to a positive definite matrix $\boldsymbol{A}$, which is given by $\sqrt{\boldsymbol{v}^\top \boldsymbol{A} \boldsymbol{v}}$. Let $\boldsymbol{I}$ denote an identity matrix. The trace of a matrix $\boldsymbol{M}$ is denoted by $\text{tr}(\boldsymbol{M})$. Given a closed convex set $\mathcal{W} \subseteq \mathbb{R}^d$, and a convex function $f : \mathbb{R}^d \to \mathbb{R}$, the sub-differential set of f at the point $\boldsymbol{w} \in \mathcal{W}$ is denoted by $\partial f(\boldsymbol{w})$. When f is differentiable, we use $\nabla f(\boldsymbol{w})$ to denote its unique subgradient (called gradient) at $\boldsymbol{w}$. We shall also exploit the next property.

Claim (Duchi et al., 2011) *Let* $\{a_t\}_{t=1}^T$ *be an arbitrary sequence of scalars, and assume that* $\frac{0}{\sqrt{0}} = 0$. *Then,* $\sum_{t=1}^T \frac{a_t^2}{\|\boldsymbol{a}_{1:t}\|_2} \leq 2\|\boldsymbol{a}_{1:T}\|_2$.

3 Online active learning for binary classification

3.1 Problem definition

We first focus on online active learning for *binary* classification. Let $(\boldsymbol{x}_1, y_1), \cdots, (\boldsymbol{x}_T, y_T)$ be a sequence of input examples, where $\boldsymbol{x}_t \in \mathbb{R}^d$ and $y_t \in \{-1, +1\}$ for any $t \in [T]$. It should be noted that the entire sequence of examples can be arbitrary, but is chosen beforehand. At each round t, the learner first observes an instance $\boldsymbol{x}_t \in \mathbb{R}^d$, and next predicts the label $\hat{y}_t = \text{sgn}(\boldsymbol{w}_t^\top \boldsymbol{x}_t)$ using its current model $\boldsymbol{w}_t \in \mathbb{R}^d$. Then, the learner is given the choice of querying the true label y_t, or not. A variable $Z_t \in \{0, 1\}$ is associated with the query decision. If $Z_t = 1$, then y_t is queried and a loss $f(\boldsymbol{w}_t; (\boldsymbol{x}_t, y_t))$ that measures the discrepancy between $\hat{y}_t$ and y_t is revealed. In light of this information, the learner can compute a new predictor $\boldsymbol{w}_{t+1} \in \mathbb{R}^d$. On the other hand, if $Z_t = 0$, then y_t remains unknown and the learner simply sets $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t$.

In what follows, $f_t(\boldsymbol{w})$ is used as an abbreviation of $f(\boldsymbol{w}; (\boldsymbol{x}_t, y_t))$. At each online round t, the hinge loss $f_t(\boldsymbol{w}_t) = \max\{0, 1 - y_t \boldsymbol{w}_t^\top \boldsymbol{x}_t\}$ is used to measure the inaccuracy of the prediction. In order to evaluate the number of online prediction mistakes made by our learner, we introduce two new symbols:

$$M_t = \mathbb{1}[y_t \boldsymbol{w}_t^\top \boldsymbol{x}_t < 0] = \mathbb{1}[\hat{y}_t \neq y_t], \; L_t = \mathbb{1}[0 \leq y_t \boldsymbol{w}_t^\top \boldsymbol{x}_t < 1],$$

where M_t indicates whether the learner has made a prediction mistake at round t, and L_t indicates whether the learner has made a correct prediction but without sufficient confidence.

The main goal of an online active learner is to achieve a predictive performance that is comparable to the corresponding fully supervised online learner, but using few label queries. Therefore, we compare the expected number of prediction mistakes made by our online learner, that is, $\mathbb{E}[\sum_{t=1}^T M_t]$, with the cumulative hinge loss of the best fully supervised classifier $\boldsymbol{w}^*$, taken with the benefit of hindsight. Specifically, $\boldsymbol{w}^* = \text{argmin}_{\boldsymbol{w} \in \mathbb{R}^d} \sum_{t=1}^T f_t(\boldsymbol{w})$ and its cumulative loss is given by $\sum_{t=1}^T f_t(\boldsymbol{w}^*)$. Importantly, the prediction mistakes of our learner is evaluated on *all* rounds, including those where true labels remain unknown.

3.2 Adaptive subgradient methods for binary classification

Adaptive subgradient methods (Duchi et al. 2011) are a family of online algorithms that can exploit the historically observed subgradients to perform more informative learning and achieve asymptotically sub-linear regret. In this section, we introduce two specific implementation methods of adaptive subgradient methods that are efficient in time and space for high-dimensional data and that will be used for updating our active learner. One method is based on *dual averaging* and the other one is founded on *mirror descent*. Both methods are fully supervised and require to query each instance's label. Both methods can endow each dimension of the predictor with an adaptive learning step-size. In order to achieve this point, a diagonal matrix $\boldsymbol{H}_t$ is computed at each round t as:

1 Query $y_t \in \{-1, +1\}$ and get $\boldsymbol{g}_t \in \partial f_t(\boldsymbol{w}_t)$
2 Let $\boldsymbol{G}_{1:t} = [\boldsymbol{g}_1, \cdots, \boldsymbol{g}_t]$
3 Let $\boldsymbol{H}_t = \delta \boldsymbol{I} + \text{diag}(\boldsymbol{s}_t)$ where $s_{t,i} = ||\boldsymbol{G}_{1:t,i}||_2$

where $\delta > 0$ is a hyperparameter and $\boldsymbol{H}_t$ can be rewritten as

$$\boldsymbol{H}_t = \delta \boldsymbol{I} + \text{diag}\left(\sum_{k=1}^{t} \boldsymbol{g}_k \boldsymbol{g}_k^\top\right)^{\frac{1}{2}} = \delta \boldsymbol{I} + \text{diag}\left(\sum_{k=1}^{t} \mathbb{1}[f_k(\boldsymbol{w}_k) > 0] \boldsymbol{x}_k \boldsymbol{x}_k^\top\right)^{\frac{1}{2}}.$$

Informally, $\boldsymbol{H}_t$ is used to approximate the Hessian of the functions $f_t(\boldsymbol{w})$ (Duchi et al. 2011). Relying on $\boldsymbol{H}_t$, the updating rules for both methods at the end of round t are defined as follows.

Dual Averaging (DA) update: the new predictor is given by

$$\boldsymbol{w}_{t+1} = \text{argmin}_{\boldsymbol{w} \in \mathbb{R}^d} \left\{ \eta \boldsymbol{w}^\top \left(\sum_{k=1}^{t} \boldsymbol{g}_k\right) + \frac{1}{2} \boldsymbol{w}^\top \boldsymbol{H}_t \boldsymbol{w} \right\} \tag{1}$$

Mirror Descent (MD) update: the new predictor is given by

$$\boldsymbol{w}_{t+1} = \text{argmin}_{\boldsymbol{w} \in \mathbb{R}^d} \left\{ \eta \boldsymbol{g}_t^\top \boldsymbol{w} + \frac{1}{2} (\boldsymbol{w} - \boldsymbol{w}_t)^\top \boldsymbol{H}_t (\boldsymbol{w} - \boldsymbol{w}_t) \right\} \tag{2}$$

Here η is the step-size hyperparameter. The updating rules (1) and (2) both admit a closed form solution. For (1), we can get $\boldsymbol{w}_{t+1} = -\eta \boldsymbol{H}_t^{-1} \sum_{k=1}^{t} \boldsymbol{g}_k$ and for (2), we have $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \boldsymbol{H}_t^{-1} \boldsymbol{g}_t$. Informally, the above two methods give frequently occurring features very low learning rates and infrequent features high learning rates (Duchi et al. 2011), which is achieved by using $\boldsymbol{H}_t$. So conceptually, the value of each diagonal element in $\boldsymbol{H}_t$ captures how frequently the feature on that dimension is seen during the online learning process.

3.3 Novel online active learning methods for binary classification

In this section, we aim to develop novel online active learning algorithms. The core challenges for designing an online active learner include (a) label query strategy: how to identify critical instances to label so that the predictive performance of the online learner can be significantly improved, and (b) updating rule: how to effectively update the online learner when the true label of an incoming instance is revealed. Our proposed algorithms adopt a novel discrimination-based label query and the DA or MD updating rule to handle the above challenges.

Our label query strategy is motivated by the following idea. In various applications characterized by high-dimensional, yet sparse, data instances, infrequently occurring features are known to be highly discriminative (Crammer et al. 2012; Duchi et al. 2011). The instances including such infrequent features are therefore important for improving the predictive performance of the online predictor, and hence, it is crucial to obtain their labels. To this point, recall that the usual *margin-based* query strategy is to draw a random variable $Z_t \in \{0, 1\}$ from a Bernoulli distribution with parameter $b/(b + p_t)$, where $p_t = |\boldsymbol{w}_t^\top \boldsymbol{x}_t|$ and $b > 0$ is a predefined hyperparameter. So, this strategy, advocated for example in (Cesa-Bianchi et al. 2006; Lu et al. 2016b; Zhao and Hoi 2013), does not take account of the *feature-based* discriminative information of instances, but only considers the predictive uncertainty of instances.

Our query strategy takes full advantage of both aspects. Here, Z_t is drawn from a Bernoulli distribution with parameter $b/(b + q_t \mathbb{1}[q_t > 0])$ where

$$q_t = |\hat{p}_t| - \frac{\eta}{2} a_t v_t, \text{ with } \hat{p}_t = \boldsymbol{w}_t^\top \boldsymbol{x}_t, \ a_t \in [0,1] \text{ and } v_t = \boldsymbol{x}_t^\top \boldsymbol{H}_{t-1}^{-1} \boldsymbol{x}_t.$$

The value of a_t will be clarified in Remark 3 in Sect. 3.4. The matrix $\boldsymbol{H}_{t-1}$ is the diagonal matrix maintained by the two adaptive subgradient methods in the previous section. We later prove that such definition of q_t helps to reduce the upper bound of the online prediction mistakes made by our proposed active learning algorithms. Intuitively, $|\hat{p}_t|$ is used to assess the uncertainty of classifying the instance $\boldsymbol{x}_t$, but this term is compensated by v_t, which quantifies the feature-based discrimination of $\boldsymbol{x}_t$. Recall that the smaller value of the i-th diagonal element of $\boldsymbol{H}_{t-1}$ implies, in some extent, the less frequently occurring for the i-th dimensional feature. Thus, the larger is the value of v_t, the more is the infrequent features that $\boldsymbol{x}_t$ contains and the more important is $\boldsymbol{x}_t$. According to this strategy, labels of instances with small $|\hat{p}_t|$ and large v_t are given high probability to be asked. Notably, when an instance exhibits a high value of v_t, i.e. $\frac{\eta}{2} a_t v_t \geq |\hat{p}_t|$, its label is queried with certainty. If y_t is queried then, in light of this information, the new predictor $\boldsymbol{w}_{t+1}$ is computed according to (1) or (2). Otherwise, keep the predictor unchanged.

We present the proposed *Discrimination-based Active Dual Averaging (D-ADA)* algorithm and the *Discrimination-based Active Mirror Descent (D-AMD)* algorithm in Algorithm 1, where *discrimination* refers to the *margin-based* uncertainty and *feature-based* discrimination. Both algorithms are defined on the same query strategy (Lines 5-6), and only differ in the choice of the updating rule (Line 13 for D-ADA and Line 14 for D-AMD).

Algorithm 1: D-ADA and D-AMD

Input: Hyperparameters $\delta > 0$, $\eta > 0$ and $b > 0$

Initialization step

1 Set $\boldsymbol{w}_1 = \boldsymbol{0}$ and $\boldsymbol{H}_0 = \delta \boldsymbol{I}$

Trials

2 **for** $t = 1, 2, \cdots$ **do**

3 Observe $\boldsymbol{x}_t$ and set $\hat{p}_t = \boldsymbol{w}_t^\top \boldsymbol{x}_t$

4 Predict with $\hat{y}_t = \text{sgn}(\hat{p}_t)$

Discrimination-based query:

5 Set $q_t = |\hat{p}_t| - \frac{\eta}{2} a_t v_t$ where $v_t = \boldsymbol{x}_t^\top \boldsymbol{H}_{t-1}^{-1} \boldsymbol{x}_t$

6 Draw a Bernoulli random variable $Z_t \in \{1, 0\}$ of parameter $b/(b + q_t \mathbb{1}[q_t > 0])$

7 **if** $Z_t = 1$ **then**

8 Query $y_t \in \{-1, +1\}$ and get $\boldsymbol{g}_t \in \partial f_t(\boldsymbol{w}_t)$

9 **else**

10 Set $\boldsymbol{g}_t = \boldsymbol{0}$

11 Set $\boldsymbol{G}_{1:t} = [\boldsymbol{g}_1, \cdots, \boldsymbol{g}_t]$

12 Set $\boldsymbol{H}_t = \delta \boldsymbol{I} + \text{diag}(\boldsymbol{s}_t)$ where $s_{t,i} = ||\boldsymbol{G}_{1:t,i}||_2$

13 **DA**: $\boldsymbol{w}_{t+1} = -\eta \, \boldsymbol{H}_t^{-1} \sum_{k=1}^{t} \boldsymbol{g}_k$

14 **MD**: $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \, \boldsymbol{H}_t^{-1} \boldsymbol{g}_t$

Our algorithms can be implemented in an efficient way. Indeed, using the fact that $\boldsymbol{s}_0 = \boldsymbol{0}$, together with the fact that $s_{t,i} = \sqrt{s_{t-1,i}^2 + g_{t,i}^2}$ for $i \in [d]$, the matrix $\boldsymbol{H}_t$ can be computed at round t in time proportional to d', the number of non-zero elements in $\boldsymbol{x}_t$, by simply using the vector $\boldsymbol{s}_{t-1}$ derived at round $t-1$ and the subgradient $\boldsymbol{g}_t$ obtained at round t. Since $\boldsymbol{H}_t$ is diagonal, its inverse can also be found in $O(d')$. Therefore, it is easy

to observe that the per-round time complexity of our algorithms is $O(d')$, and the per-round space complexity is $O(d)$.

3.4 Theoretical analysis for D-ADA and D-AMD

The next theorem provides for D-ADA and D-AMD expected mistake bounds, which refer to upper bounds for $\mathbb{E}\left[\sum_{t=1}^{T} M_t\right]$. In all results described below, expectations are taken with respect to the randomized query strategy, and $\boldsymbol{w}^* = \operatorname{argmin}_{\boldsymbol{w}\in\mathbb{R}^d} \sum_{t=1}^{T} f_t(\boldsymbol{w})$.

Theorem 1 *If D-ADA and D-AMD are run with $b \geq 2$, then the expected number of online prediction mistakes made by D-ADA for T rounds satisfies the inequality*:

$$\begin{aligned}\mathbb{E}\left[\sum_{t=1}^{T} M_t\right] \leq \mathbb{E}\left[\sum_{t=1}^{T} Z_t f_t(\boldsymbol{w}^*)\right] + \frac{bA_1}{2\eta}\operatorname{tr}(\mathbb{E}[\boldsymbol{H}_T]) - \frac{1}{b}\mathbb{E}\left[\sum_{t:q_t\leq 0} L_t\right] \\ + \frac{\eta}{2b}\mathbb{E}\left[\sum_{t:q_t\leq 0} a_t\|\boldsymbol{g}_t\|^2_{\boldsymbol{H}_{t-1}^{-1}}\right] + \frac{\eta}{2b}\mathbb{E}\left[\sum_{t=1}^{T}(1-a_t)\|\boldsymbol{g}_t\|^2_{\boldsymbol{H}_{t-1}^{-1}}\right]\end{aligned} \tag{3}$$

where $A_1 = \|\boldsymbol{w}^*\|^2_\infty$. For D-AMD, the following inequality holds:

$$\begin{aligned}\mathbb{E}\left[\sum_{t=1}^{T} M_t\right] \leq \mathbb{E}\left[\sum_{t=1}^{T} Z_t f_t(\boldsymbol{w}^*)\right] + \frac{A_2 + (b-1)^2 A_1}{\eta b}\operatorname{tr}(\mathbb{E}[\boldsymbol{H}_T]) - \frac{1}{b}\mathbb{E}\left[\sum_{t:q_t\leq 0} L_t\right] \\ + \frac{\eta}{2b}\mathbb{E}\left[\sum_{t:q_t\leq 0} a_t\|\boldsymbol{g}_t\|^2_{\boldsymbol{H}_t^{-1}}\right] + \frac{\eta}{2b}\mathbb{E}\left[\sum_{t=1}^{T}(1-a_t)\|\boldsymbol{g}_t\|^2_{\boldsymbol{H}_t^{-1}}\right]\end{aligned} \tag{4}$$

where $A_2 = \max_{t\in[T]} \|\boldsymbol{w}^* - \boldsymbol{w}_t\|^2_\infty$ and A_1 is defined as above.

Remark 1 Except the term $\mathbb{E}\left[\sum_{t=1}^{T} Z_t f_t(\boldsymbol{w}^*)\right]$, the dominant components of our mistake bounds depend on the expected trace of the diagonal matrix $\boldsymbol{H}_T$. Indeed, for D-ADA, with the assumption that $\delta \geq \max_t \|\boldsymbol{g}_t\|_\infty$, we can get

$$\sum_{t=1}^{T}\|\boldsymbol{g}_t\|^2_{\boldsymbol{H}_{t-1}^{-1}} \leq \sum_{t=1}^{T}\boldsymbol{g}_t^\top \operatorname{diag}(\boldsymbol{s}_t)^{-1}\boldsymbol{g}_t = \sum_{t=1}^{T}\sum_{i=1}^{d}\frac{g_{t,i}^2}{\|\boldsymbol{G}_{1:t,i}\|_2} \leq 2\sum_{i=1}^{d}\|\boldsymbol{G}_{1:T,i}\|_2$$

where the last inequality is derived from Claim 1. By contrast, for D-AMD, without any assumptions about δ, we can also get $\sum_{t=1}^{T}\|\boldsymbol{g}_t\|^2_{\boldsymbol{H}_t^{-1}} \leq 2\sum_{i=1}^{d}\|\boldsymbol{G}_{1:T,i}\|_2$. Therefore, for D-ADA, we have

$$\mathbb{E}\left[\sum_{t:q_t\leq 0} a_t\|\boldsymbol{g}_t\|^2_{\boldsymbol{H}_{t-1}^{-1}}\right] + \mathbb{E}\left[\sum_{t=1}^{T}(1-a_t)\|\boldsymbol{g}_t\|^2_{\boldsymbol{H}_{t-1}^{-1}}\right] \leq 2\mathbb{E}\left[\sum_{i=1}^{d}\|\boldsymbol{G}_{1:T,i}\|_2\right].$$

Similarly, for D-AMD, the sum of the last two terms in (4) is also less than or equal to $2\mathbb{E}\left[\sum_{i=1}^{d}\|\boldsymbol{G}_{1:T,i}\|_2\right]$. The facts that $\sum_{i=1}^{d}\|\boldsymbol{G}_{1:T,i}\|_2 = \operatorname{tr}(\boldsymbol{H}_T) - \delta d$ and $\operatorname{tr}(\boldsymbol{H}_T)$ is sublinear (Duchi et al. 2011) imply that as T increases, our algorithms can converge to $\boldsymbol{w}^*$ when the query hyperparameter $b \geq 2$.

Remark 2 The expected mistake bounds can reveal the theoretical motivation of our query rule. Taking D-ADA for example, if the query rule exploits only the margin-based uncertainty of instances, that is, taking $a_t = 0, \forall t \in [T]$, the sum of the last two terms in (3) reaches its maximal value $A = \frac{\eta}{2b}\mathbb{E}[\sum_{t=1}^{T} \|\boldsymbol{g}_t\|^2_{\boldsymbol{H}_{t-1}^{-1}}]$. However, if the query rule also takes full advantage of the feature-based discrimination of instances, namely, taking $0 < a_t \leq 1, \forall t \in [T]$, then the sum of the last two terms in (3) is $B = \frac{\eta}{2b}\mathbb{E}\left[\sum_{t:q_t \leq 0} a_t \|\boldsymbol{g}_t\|^2_{\boldsymbol{H}_{t-1}^{-1}} + \sum_{t=1}^{T}(1-a_t)\|\boldsymbol{g}_t\|^2_{\boldsymbol{H}_{t-1}^{-1}}\right]$. In view of the fact that $A - B = C = \frac{\eta}{2b}\mathbb{E}[\sum_{t:q_t>0} a_t \|\boldsymbol{g}_t\|^2_{\boldsymbol{H}_{t-1}^{-1}}] \geq 0$, we can derive that using the feature-based discrimination of instances tends to produce a smaller expected mistake bound, since the non-negative term C is eliminated from the upper bound of the expected number of online prediction mistakes made by D-ADA.

Remark 3 Three cases of a_t are considered:

- *Case 1* If $a_t = 0, \forall t \in [T]$, our query strategy becomes the margin-based query strategy in which feature-based discrimination of instances is not utilized.
- *Case 2* If $a_t = 1, \forall t \in [T]$, the sums of the last two terms in (3) and (4) reach their minimal value $\frac{\eta}{2b}\mathbb{E}[\sum_{t:q_t \leq 0} \|\boldsymbol{g}_t\|^2_{\boldsymbol{H}_{t-1}^{-1}}]$ and $\frac{\eta}{2b}\mathbb{E}[\sum_{t:q_t \leq 0} \|\boldsymbol{g}_t\|^2_{\boldsymbol{H}_t^{-1}}]$, respectively, which would be ideal when the number of online rounds for which $q_t \leq 0$, namely, $\sum_{t:q_t \leq 0} 1$, is also less. But if $\sum_{t:q_t \leq 0} 1$ cannot be made less, the number of labels queried by our algorithms is at least $\sum_{t:q_t \leq 0} 1$.
- *Case 3* If $a_t = 1/\max\{1, \boldsymbol{x}_t^\top \boldsymbol{x}_t\} \in (0, 1], \forall t \in [T]$, taking D-ADA for example, the sum of the last two terms in (3) is between $\frac{\eta}{2b}\mathbb{E}[\sum_{t:q_t \leq 0} \|\boldsymbol{g}_t\|^2_{\boldsymbol{H}_{t-1}^{-1}}]$ and $\frac{\eta}{2b}\mathbb{E}[\sum_{t=1}^{T} \|\boldsymbol{g}_t\|^2_{\boldsymbol{H}_{t-1}^{-1}}]$, which tends to produce a larger bound than that in Case 2, but a smaller bound than that in Case 1. However, since a_t takes a smaller value than that in Case 2, the number of online rounds for which $q_t \leq 0$ can be reduced so that smaller label query ratios can be obtained than in Case 2.

Remark 4 The expected number of labels queried by our algorithms is $\mathbb{E}[\sum_{t:q_t \leq 0} 1 + \sum_{t:q_t>0} \frac{b}{b+q_t}]$, where the value of q_t relies on a_t. As can be seen, our algorithms query at least $\mathbb{E}[\sum_{t:q_t \leq 0} 1]$ labels. By increasing the value of b, more label queries are triggered. Since q_t is data-dependent, we have been unable to provide an upper bound for the query number.

4 Extension to online multiclass classification

4.1 Problem setting

In this section, we extend D-ADA and D-AMD to *multiclass* classification tasks. To achieve the goal, both updating rules and query strategy need to be generalized to the multiclass setting. In generalizing the updating rules, we choose to use the multi-prototype method in (Crammer et al., 2006) since the method makes the extension feasible and more importantly, it contributes to good theoretical properties of our extended multiclass active learning methods. At each online round t, the method maintains a multiclass classifier $\boldsymbol{W}_t$

that consists of C class-specific predictors $\boldsymbol{w}_t^{(i)} \in \mathbb{R}^d, \forall i \in [C]$. For an incoming instance $\boldsymbol{x}_t$, the method predicts the label of $\boldsymbol{x}_t$ as $\hat{y}_t = \text{argmax}_{i\in[C]}\{(\boldsymbol{w}_t^{(i)})^\top \boldsymbol{x}_t\}$. Similarly to binary classification, a label query strategy is used to decide whether to query the true label $y_t \in [C]$ of $\boldsymbol{x}_t$. Once y_t is queried, a loss $f(\boldsymbol{W}_t;(\boldsymbol{x}_t, y_t))$ that measures the predictive inaccuracy of $\boldsymbol{W}_t$ on the example $(\boldsymbol{x}_t, y_t)$ is incurred. Relying on this loss, $\boldsymbol{W}_t$ is updated to $\boldsymbol{W}_{t+1}$, which can be converted into updating each class-specific classifier $\boldsymbol{w}_t^{(i)}$. If y_t is not queried, the current classifier $\boldsymbol{W}_t$ is kept unchanged.

In what follows, $f(\boldsymbol{W};(\boldsymbol{x}_t, y_t))$ is abbreviated as $f_t(\boldsymbol{W})$. The loss that we use at round t is the multiclass hinge loss $f_t(\boldsymbol{W}_t) = \max\left\{0, 1 + (\boldsymbol{w}_t^{(r_t)})^\top \boldsymbol{x}_t - (\boldsymbol{w}_t^{(y_t)})^\top \boldsymbol{x}_t\right\}$ where $r_t = \text{argmax}_{i\in[C], i\neq y_t} (\boldsymbol{w}_t^{(i)})^\top \boldsymbol{x}_t$. In order to evaluate the number of online prediction mistakes made by our multiclass classifier, M_t and L_t are re-defined as

$$M_t = \mathbb{1}[\boldsymbol{x}_t^\top \boldsymbol{w}_t^{(y_t)} < \boldsymbol{x}_t^\top \boldsymbol{w}_t^{(r_t)}] = \mathbb{1}[\hat{y}_t \neq y_t],\ L_t = \mathbb{1}[0 \leq \boldsymbol{x}_t^\top \boldsymbol{w}_t^{(y_t)} - \boldsymbol{x}_t^\top \boldsymbol{w}_t^{(r_t)} < 1].$$

Let $\boldsymbol{W}^* = [\boldsymbol{w}_*^{(1)}, \cdots, \boldsymbol{w}_*^{(C)}]$ be the best fully supervised multiclass classifier chosen in hindsight, that is, $\boldsymbol{W}^* = \text{argmin}_{\boldsymbol{W}\in\mathbb{R}^{d\times C}} \sum_{t=1}^T f_t(\boldsymbol{W})$. As usual, we compare the expected number of prediction mistakes made by our learner, that is, $\mathbb{E}[\sum_{t=1}^T M_t]$, with the cumulative multiclass hinge loss of $\boldsymbol{W}^*$, given by $\sum_{t=1}^T f_t(\boldsymbol{W}^*)$.

4.2 Novel online active learning algorithms for multiclass classification

4.2.1 Multiclass updating rules

We use the dual averaging and mirror descent methods to update each class-specific predictor $\boldsymbol{w}_t^{(i)}$. At each online round t, both updating rules need to maintain C class-specific diagonal matrix $\boldsymbol{H}_t^{(i)}$ computed in the following way:

1 Query $y_t \in [C]$ and get $\boldsymbol{g}_t^{(1)}, \cdots, \boldsymbol{g}_t^{(C)}$
2 $\forall i \in [C]$, let $\boldsymbol{G}_{1:t}^{(i)} = [\boldsymbol{g}_1^{(i)}, \cdots, \boldsymbol{g}_t^{(i)}]$
3 $\forall i \in [C]$, let $\boldsymbol{H}_t^{(i)} = \delta \boldsymbol{I} + \text{diag}(\boldsymbol{s}_t^{(i)})$ where $\forall j \in [d], s_{t,j}^{(i)} = ||\boldsymbol{G}_{1:t,j}^{(i)}||_2$

where $\boldsymbol{g}_t^{(i)}$ is the partial derivative of $f_t(\boldsymbol{W})$ with respect to $\boldsymbol{w}^{(i)}$ at the point $\boldsymbol{W}_t$. If $f_t(\boldsymbol{W}_t) > 0$, we can get

$$\boldsymbol{g}_t^{(i)} = \begin{cases} \boldsymbol{x}_t, & \text{if } i = r_t; \\ -\boldsymbol{x}_t, & \text{if } i = y_t; \\ \boldsymbol{0}, & \text{otherwise.} \end{cases}$$

Otherwise, it follows that $\boldsymbol{g}_t^{(i)} = \boldsymbol{0}$, $\forall i \in [C]$. Based on the matrix $\boldsymbol{H}_t^{(i)}$, each new predictor $\boldsymbol{w}_{t+1}^{(i)}$ at the end of round t is defined as follows.

Multiclass Dual Averaging (M-DA) update:

$$\boldsymbol{w}_{t+1}^{(i)} = \operatorname{argmin}_{\boldsymbol{w}\in\mathbb{R}^d} \left\{ \eta\, \boldsymbol{w}^\top \left(\sum_{k=1}^{t} \boldsymbol{g}_k^{(i)} \right) + \frac{1}{2} \boldsymbol{w}^\top \boldsymbol{H}_t^{(i)} \boldsymbol{w} \right\}, \forall i \in [C] \tag{5}$$

Multiclass Mirror Descent (M-MD) update:

$$\boldsymbol{w}_{t+1}^{(i)} = \operatorname{argmin}_{\boldsymbol{w}\in\mathbb{R}^d} \left\{ \eta \boldsymbol{w}^\top \boldsymbol{g}_t^{(i)} + \frac{1}{2} (\boldsymbol{w} - \boldsymbol{w}_t^{(i)})^\top \boldsymbol{H}_t^{(i)} (\boldsymbol{w} - \boldsymbol{w}_t^{(i)}) \right\}, \forall i \in [C] \tag{6}$$

Here η is again a step-size hyperparameter. According to (5) and (6), if $f_t(\boldsymbol{W}_t) > 0$, for (5), we obtain $\boldsymbol{w}_{t+1}^{(i)} = -\eta \left(\boldsymbol{H}_t^{(i)}\right)^{-1} \sum_{k=1}^{t} \boldsymbol{g}_k^{(i)}$ for $i \in \{y_t, r_t\}$ and $\boldsymbol{w}_{t+1}^{(i)} = \boldsymbol{w}_t^{(i)}$ for $\forall i \notin \{y_t, r_t\}$, and for (6), we have $\boldsymbol{w}_{t+1}^{(y_t)} = \boldsymbol{w}_t^{(y_t)} + \eta \left(\boldsymbol{H}_t^{(y_t)}\right)^{-1} \boldsymbol{x}_t$, $\boldsymbol{w}_{t+1}^{(r_t)} = \boldsymbol{w}_t^{(r_t)} - \eta \left(\boldsymbol{H}_t^{(r_t)}\right)^{-1} \boldsymbol{x}_t$ and $\boldsymbol{w}_{t+1}^{(i)} = \boldsymbol{w}_t^{(i)}$ for $\forall i \notin \{y_t, r_t\}$. If $f_t(\boldsymbol{W}_t) = 0$, then for both updating rules, it holds that $\boldsymbol{w}_{t+1}^{(i)} = \boldsymbol{w}_t^{(i)}$ for $\forall i \in [C]$.

From (5) and (6), it seems that each class-specific classifier $\boldsymbol{w}_{t+1}^{(i)}$ is updated independently of the others, but the fact is that all class-specific classifiers are simultaneously updated for achieving one global objective. One clue is that for any $i \in [C]$, $\boldsymbol{g}_t^{(i)}$ is connected with the common loss $f_t(\boldsymbol{W}_t)$. Indeed, by following the similar derivation process to that in (Duchi et al. 2011), one can easily prove that the above two fully-supervised multiclass classification methods can achieve a sublinear regret, which implies that they both asymptotically converge to the best hindsight $\boldsymbol{W}^*$.

4.2.2 Multiclass query strategy

The multiclass margin-based query strategy in (Lu et al. 2016b) uses an *approximated* margin to replace the genuine margin for measuring the predictive uncertainty. Specifically, for an instance $\boldsymbol{x}_t$ with the true label y_t, the multiclass predictive margin for $\boldsymbol{x}_t$ is originally defined as $m_t = (\boldsymbol{w}_t^{(y_t)})^\top \boldsymbol{x}_t - \max_{i\in[C], i\neq y_t} (\boldsymbol{w}_t^{(i)})^\top \boldsymbol{x}_t$. Since y_t is unknown before label querying, m_t cannot be computed. Therefore, an approximated margin p_t is used to replace m_t:

$$p_t = (\boldsymbol{w}_t^{(\hat{y}_t)})^\top \boldsymbol{x}_t - \max_{i\in[C], i\neq \hat{y}_t} (\boldsymbol{w}_t^{(i)})^\top \boldsymbol{x}_t. \tag{7}$$

It satisfies $p_t \leq |m_t|$ for any $t \in [T]$. The query strategy in (Lu et al. 2016b) then draws a random variable $Z_t \in \{0, 1\}$ from a Bernoulli distribution with parameter $b/(b + p_t)$, where $b > 0$ is still a scaling factor on p_t. This strategy does not take into account the feature-based discrimination of $\boldsymbol{x}_t$.

Our multiclass query strategy exploits both margin-based uncertainty and feature-based discrimination of instances. According to the corresponding closed form solution of (5) and (6), we can find that even if y_t is queried at round t, for both updating rules, it always holds that for $\forall i \notin \{y_t, r_t\}$, $\boldsymbol{w}_{t+1}^{(i)} = \boldsymbol{w}_t^{(i)}$. This suggests that the example $(\boldsymbol{x}_t, y_t)$ cannot improve all the other class-specific classifiers except $\boldsymbol{w}_t^{(y_t)}$ and $\boldsymbol{w}_t^{(r_t)}$. Therefore, it is pointless to evaluate the feature-based discrimination of $\boldsymbol{x}_t$ for all the other classes except the classes y_t and r_t. In view of the fact, we focus on evaluating the feature-based discrimination of $\boldsymbol{x}_t$ only for the two classes y_t and r_t, which is defined as

$$\rho_t = \boldsymbol{x}_t^\top (\boldsymbol{H}_{t-1}^{(y_t)})^{-1} \boldsymbol{x}_t + \boldsymbol{x}_t^\top (\boldsymbol{H}_{t-1}^{(r_t)})^{-1} \boldsymbol{x}_t.$$

The larger is ρ_t, the more is the infrequent features that $\boldsymbol{x}_t$ contains for the classes y_t and r_t. Similarly, y_t and r_t are unknown before label querying and thus ρ_t cannot be computed. We use an approximated quantity v_t to replace ρ_t:

$$v_t = \boldsymbol{x}_t^\top (\boldsymbol{H}_{t-1}^{(\hat{y}_t)})^{-1} \boldsymbol{x}_t + \max_{i \in [C], i \neq \hat{y}_t} \boldsymbol{x}_t^\top (\boldsymbol{H}_{t-1}^{(i)})^{-1} \boldsymbol{x}_t \tag{8}$$

It is easy to observe that if $y_t = \hat{y}_t$, then $v_t \geq \rho_t$; if $y_t \neq \hat{y}_t$, then $r_t = \hat{y}_t$ and it still follows that $v_t \geq \rho_t$.

In our multiclass query strategy, Z_t is drawn from a Bernoulli distribution with parameter $b/(b + q_t \mathbb{1}[q_t > 0])$ where

$$q_t = p_t - \frac{\eta}{2} a_t v_t \text{ with } a_t \in [0, 1] \tag{9}$$

and p_t and v_t are defined in (7) and (8), respectively. Here a_t has the same definition and effect as that in the binary classification setting. It is easy to check that $q_t \leq |m_t| - \frac{\eta}{2} a_t \rho_t$ for any $t \in [T]$. Once $Z_t = 1$, M-DA update or M-MD update can be used to improve the multiclass classifier $\boldsymbol{W}_t$ to $\boldsymbol{W}_{t+1}$. Otherwise, set $\boldsymbol{g}_t^{(i)} = \boldsymbol{0}$ for $\forall i \in [C]$ and keep the classifier unchanged.

Based on the above discussion, we present the *Multiclass D-ADA (MD-ADA)* algorithm and the *Multiclass D-AMD (MD-AMD)* algorithm in Algorithm 2.

Algorithm 2: MD-ADA and MD-AMD

Input: hyperparameters $\delta > 0$, $\eta > 0$, and $b > 0$

Initialization step

$\forall i \in [C]$, set $\boldsymbol{w}_1^{(i)} = \boldsymbol{0}$ and $\boldsymbol{H}_0^{(i)} = \delta \boldsymbol{I}$

Trials

for $t = 1, 2, \cdots$ **do**

  Observe $\boldsymbol{x}_t$

  Predict with $\hat{y}_t = \text{argmax}_{i \in [C]} \{(\boldsymbol{w}_t^{(i)})^\top \boldsymbol{x}_t\}$

  Multiclass discrimination-based query:

  Compute q_t according to (9)

  Draw a Bernoulli random variable $Z_t \in \{1, 0\}$ of parameter $b/(b + q_t \mathbb{1}[q_t > 0])$

  if $Z_t = 1$ **then**

    Query $y_t \in [C]$ and get $\boldsymbol{g}_t^{(1)}, \cdots, \boldsymbol{g}_t^{(C)}$

  else

    $\forall i \in [C]$, set $\boldsymbol{g}_t^{(i)} = \boldsymbol{0}$

  $\forall i \in [C]$, let $\boldsymbol{G}_{1:t}^{(i)} = [\boldsymbol{g}_1^{(i)}, \cdots, \boldsymbol{g}_t^{(i)}]$

  $\forall i \in [C]$, let $\boldsymbol{H}_t^{(i)} = \delta \boldsymbol{I} + \text{diag}(\boldsymbol{s}_t^{(i)})$ where $\forall j \in [d], s_{t,j}^{(i)} = ||\boldsymbol{G}_{1:t,j}^{(i)}||_2$

  M-DA: $\forall i \in [C]$, get $\boldsymbol{w}_{t+1}^{(i)}$ by (5)

  M-MD: $\forall i \in [C]$, get $\boldsymbol{w}_{t+1}^{(i)}$ by (6)

4.3 Theoretical analysis for MD-ADA and MD-AMD

Theorem 2 *If MD-ADA and MD-AMD are run with $b \geq 2$, then the expected number of online prediction mistakes made by MD-ADA for T rounds satisfies the following inequality*:

$$\mathbb{E}\left[\sum_{t=1}^{T} M_t\right] \leq \mathbb{E}\left[\sum_{t=1}^{T} Z_t f_t(\boldsymbol{W}^*)\right] + \frac{bA_1}{2\eta}\sum_{i=1}^{C}\mathrm{tr}(\mathbb{E}[\boldsymbol{H}_T^{(i)}]) - \frac{1}{b}\mathbb{E}\left[\sum_{t:q_t\leq 0} L_t\right]$$
$$+\frac{\eta}{2b}\mathbb{E}\left[\sum_{t:q_t\leq 0}\sum_{i=1}^{C} a_t\|\boldsymbol{g}_t^{(i)}\|^2_{(\boldsymbol{H}_{t-1}^{(i)})^{-1}}\right] + \frac{\eta}{2b}\mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{C}(1-a_t)\|\boldsymbol{g}_t^{(i)}\|^2_{(\boldsymbol{H}_{t-1}^{(i)})^{-1}}\right]$$

where $A_1 = \max_{i\in[C]} \|\boldsymbol{w}_*^{(i)}\|_\infty^2$. *For MD-AMD, the following inequality holds*:

$$\mathbb{E}\left[\sum_{t=1}^{T} M_t\right] \leq \mathbb{E}\left[\sum_{t=1}^{T} Z_t f_t(\boldsymbol{W}^*)\right] + \frac{A_2 + (b-1)^2 A_1}{\eta b}\sum_{i=1}^{C}\mathrm{tr}(\mathbb{E}[\boldsymbol{H}_T^{(i)}]) - \frac{1}{b}\mathbb{E}\left[\sum_{t:q_t\leq 0} L_t\right]$$
$$+\frac{\eta}{2b}\mathbb{E}\left[\sum_{t:q_t\leq 0}\sum_{i=1}^{C} a_t\|\boldsymbol{g}_t^{(i)}\|^2_{(\boldsymbol{H}_t^{(i)})^{-1}}\right] + \frac{\eta}{2b}\mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{C}(1-a_t)\|\boldsymbol{g}_t^{(i)}\|^2_{(\boldsymbol{H}_t^{(i)})^{-1}}\right]$$

where $A_2 = \max_{i\in[C],t\in[T]} \|\boldsymbol{w}_*^{(i)} - \boldsymbol{w}_t^{(i)}\|_\infty^2$ *and* A_1 *is defined as above*.

The analytical process is similar to that for Theorem 1, so we just skip it here. The theorem reveals that our multiclass active learning algorithms can converge to the best fixed fully-supervised classifier $\boldsymbol{W}^*$ as the query hyperparameter $b \geq 2$.

5 Experiments

Two series of experiments have been conducted for evaluating the empirical performance of our proposed algorithms. The first series evaluates D-ADA and D-AMD for online binary classification tasks. The second series evaluates MD-ADA and MD-AMD for online multiclass classification tasks.

5.1 Evaluation of D-ADA and D-AMD for binary classification tasks

5.1.1 Binary classification datasets

We have randomly chosen six high-dimensional datasets to perform experiments. On these datasets, maintaining a full correlation matrix for updating the classifier is infeasible, so one has to use a diagonal matrix. The datasets are described in Table 1. *Basehock* and *Pcmac* are subsets extracted from *20Newsgroups*[1]. *Farm_ads* was collected from text ads found on twelve websites dealing with farm animal related topics, and the goal is to identify whether the content owner approves of the ad, or not. *Gisette* is a handwritten digit recognition problem, for which the task is to separate the digits '4' and '9'. Both *farm_ads* and *gisette* can be downloaded from UCI repository. *Spam_corpus* (Katakis et al., 2009), collected from the anti-spam platform SpamAssasin, contains 9,324 emails, each encoded as a boolean bag-of-words vector, and around 20% of these emails are spams. *Url_day0*, a subset of the *URL* dataset (Ma et al., 2009), contains all Day 0's URLs, each represented

[1] http://www.cad.zju.edu.cn/home/dengcai/Data/TextData.html.

Table 1 A summary of binary classification datasets in the experiments

Dataset	# inst.	# fea.	Dataset	# inst.	# fea.	Dataset	# inst.	# fea.
Basehock	1993	11148	Pcmac	1945	9877	Farm_ads	4143	54877
Gisette	7000	5000	Spam_corpus	9324	39916	Url_day0	16000	74113

[1] "# inst." = the number of instances, "# fea." = the number of features

by its lexical and host-based features. The task is to separate malicious URLs from benign ones, and around 33% of these URLs are malicious.

5.1.2 Evaluation of our label query strategy

We perform an ablation study to demonstrate the benefit of our label query strategy. We compare the following two groups of algorithms:

1. R-ADA, M-ADA, D-ADA, D-ADA-I: these algorithms use the same dual averaging updating rule, but different label query strategy.
2. R-AMD, M-AMD, D-AMD, D-AMD-I: these algorithms use the same mirror descent updating rule, but different label query strategy.

Different query strategies are as follows:

1. R-ADA and R-AMD use the random query strategy.
2. M-ADA and M-AMD use the margin-based query strategy which is equivalent to our query strategy that adopts $a_t = 0, \forall t \in [T]$ in Algorithm 1
3. D-ADA and D-AMD use our query strategy that adopts $a_t = 1/\max\{1, \boldsymbol{x}_t^\top \boldsymbol{x}_t\}, \forall t \in [T]$ in Algorithm 1
4. D-ADA-I and D-AMD-I also use our proposed query strategy, but adopt $a_t = 1, \forall t \in [T]$ in Algorithm 1

We evaluate the online F1-measure achieved by these algorithms at the label query ratio in $\{10^{-1}, 10^{-0.9}, \cdots, 10^{-0.1}\}$, where $F1 - measure = \frac{2 * precision * recall}{precision + recall}$. For each algorithm, hyperparameter optimization is carried out using grid search with cross validation. In performing cross validation, only one pass over the training splits is allowed. Once hyperparameters at each certain query ratio are determined, each algorithm is run 20 times, each time with a different random permutation of examples in the dataset. The online F1-measure achieved by these active learners at different query ratios is averaged over the 20 runs, and reported in Figs. 1 and 2.

From Fig. 1, we observe that R-ADA performs the worst, M-ADA the second worst, and D-ADA and D-ADA-I perform the best. This fact shows that using margin-based uncertainty is better than using nothing in the query strategy, while exploiting both margin-based predictive uncertainty and feature-based discrimination is also more beneficial than using only margin-based uncertainty. D-ADA-I sometimes cannot achieve low query ratios, for example, on the first three datasets. By using smaller a_t, D-ADA can achieve lower query ratios, but mostly at the price of performance degradation. Thus, the performance

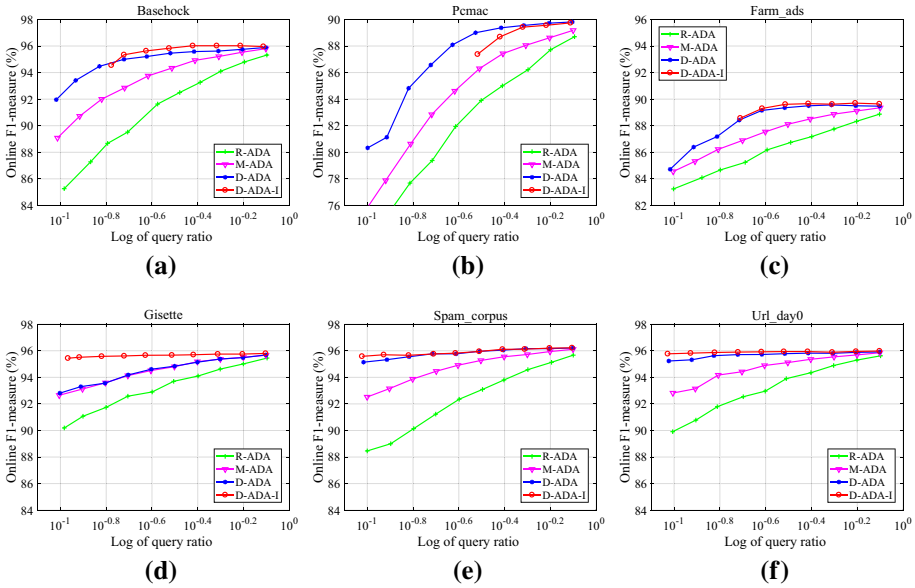

Fig. 1 Comparison of these algorithms based on dual averaging update at various query ratios

of D-ADA is generally better than that of M-ADA but worse than that of D-ADA-I. On *Gisette*, D-ADA behaves similarly to M-ADA since this dataset has very large feature values which leads to a small value of a_t. According to Algorithm 1, if a_t tends to zero, D-ADA will degrade to M-ADA. Similar phenomenon can also be observed from Fig. 2. These results corroborate the fact that exploiting feature-based discrimination of instances helps to identify the critical instances in the label queries and enhance predictive performance.

5.1.3 Comparison with existing algorithms

In this section, we have compared the following algorithms:

- PAA-II (Lu et al., 2016b): Passive Aggressive Active learning.
- SOP (Cesa-Bianchi et al., 2006): selective sampling Second-Order Perceptron.
- SOAL (Hao et al., 2018): Second-order Online Active Learning.
- D-ADA, D-AMD, D-ADA-I and D-AMD-I: as described in the previous section.
- DA and MD: the fully supervised version of D-ADA and D-AMD.

Notably, the diagonal matrix versions of SOAL and SOP that keep only diagonal elements of the full correlation matrix are used here. D-ADA and D-AMD are used on the first three datasets, while D-ADA-I and D-AMD-I are used on the remaining ones. Similarly to the previous experiments, grid search with cross validation is used to optimize hyperparameters. Each algorithm is run 20 times on each dataset and the online F1-measure achieved by these algorithms at different query ratios is averaged over the 20 runs, and reported in Fig. 3. Moreover, we also report in Table 2 the results at the query ratio near 10^{-1} and $10^{-0.7}$
.

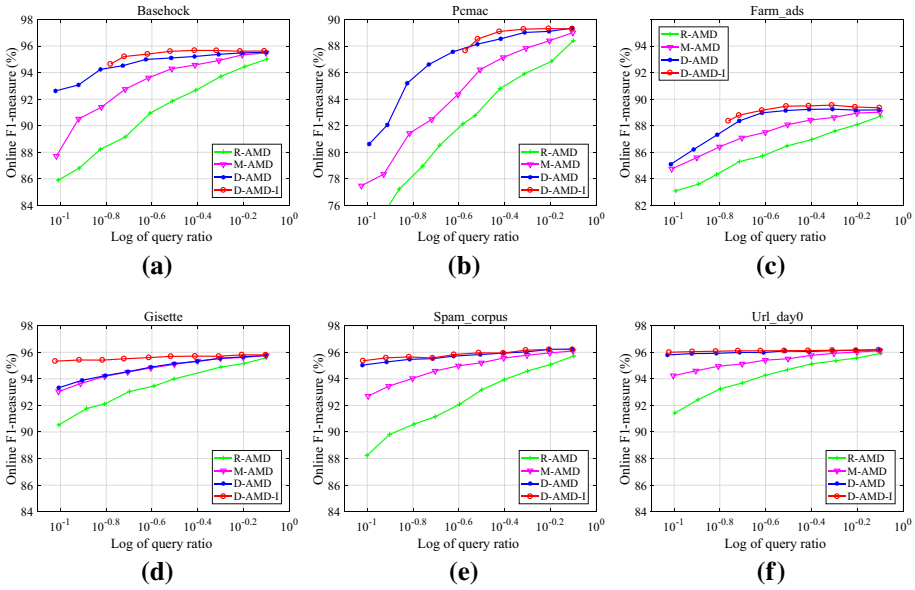

Fig. 2 Comparison of these algorithms based on mirror descent update at various query ratios

The most telling observation from Fig. 3 is that, our algorithms outperform all compared active learning algorithms at extensive label query ratios. Specifically, SOP performs the worst, PAA-II the second worst, then it comes to SOAL, which is inferior to our algorithms. Moreover, D-ADA (D-ADA-I) sometimes outperforms D-AMD (D-AMD-I), but sometimes not. We also notice that our algorithms can achieve comparable F1-measure to their fully supervised counterpart, but using fewer label queries on these datasets. From Table 2, according to paired t-tests at 95% confidence level, we observe that only on *Farm_ads* at query ratio near 10%, our algorithms perform comparably to SOAL, while in the rest of all cases, our algorithms are significantly better than the other competitors. These experimental results demonstrate the superiority of our algorithms over the existing ones.

5.1.4 Sensitivity analysis

In this section, we focus on analyzing the sensitivity of the proposed algorithms to the hyperparameters. Specifically, we observe that (a) when the hyperparameter b is fixed as $b = 1$, how online F1-measure and query ratio vary with different δ and η; (b) when the hyperparameter δ is fixed as $\delta = 0.001$, how online F1-measure and query ratio vary with different η and b; (c) when the hyperparameter η is fixed as $\eta = 0.01$, how online F1-measure and query ratio vary with different δ and b. Due to the space constraint, we only present the results for D-ADA on *Basehock* in Fig. 4, where different colors represent different F1-measure or query ratio.

From Fig. 4, we observe a common phenomenon that under many small query ratios, D-ADA can obtain F1-measures that are comparable to or even better than that under large query ratios, which implies the advantage of D-ADA. From Fig. 4a, d, c and f, we find that a large δ often leads to low F1-measures, but a small δ leads to high query ratios. This involves a tradeoff between F1-measure and query ratio. Once δ is

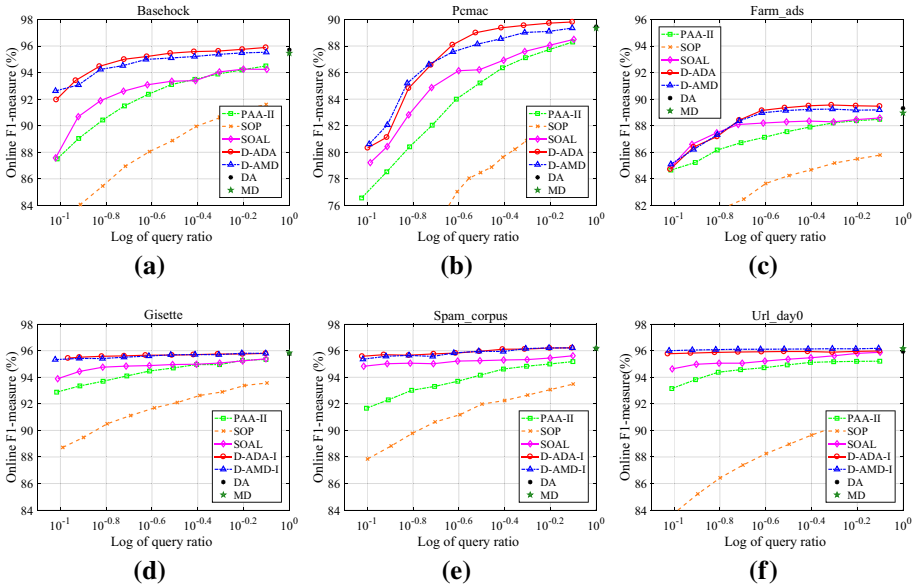

Fig. 3 Online F1-measure achieved by each active learning algorithm at different query ratios

fixed, η should be neither too large nor too small, according to Fig. 4b. This observation is consistent with Theorem 1 since too large or too small values of η both lead to large mistake bounds. So the optimal η should be searched around 1. From Fig. 4b, e, c and f, we observe that when δ and η are fixed, the minimal query ratio that D-ADA can attain is determined accordingly. Although the query ratio decreases with diminishing b, one can only obtain a query ratio above the minimal query ratio. In practice, we recommend to first find appropriate values of δ and η by a grid search, then tune b to get the desired query ratio.

5.1.5 Comparison with existing algorithms in a fixed parameter setting

In this section, we adopt a different parameter setting to perform the comparative experiments in Sect. 5.1.3. Specifically, we fix the other parameters except the query parameter b on each dataset, and adjust b to obtain different query ratios and observe how much F1-measure can be obtained. This setting is deemed to be more suitable for stream-based learning. The fixed parameter values are given in Table 3. Note that these parameter values are chosen coarsely in order to observe how these algorithms behave in a more practical setting, rather than an ideal one. SOP is excluded in this experiment since it performs the worst according to Sect. 5.1.3. The experimental results are displayed in Fig. 5.

From Fig. 5, we observe that our proposed algorithms beat the other algorithms at extensive query ratios. Although our algorithms sometimes cannot achieve very low query ratios, such as on *Pcmac*, this phenomenon also exists for SOAL. Basically,

Table 2 Online F1-measure obtained at the label query ratio near 10^{-1} and $10^{-0.7}$

Algorithm	F1measure (%)	query (%)	F1measure (%)	query (%)	F1measure (%)	query (%)
	Basehock		Pcmac		Farm_ads	
PAA-II	87.50±1.40	9.68±0.41	76.56±2.42	9.45±0.61	84.66±0.47	9.68±0.43
SOP	82.02±2.05	9.75±0.55	70.12±3.39	9.53±0.57	79.92±1.30	10.07±0.45
SOAL	87.58±4.14	9.53±0.54	79.22±1.94	10.31±0.55	**84.96±1.56**	9.78±0.47
D-ADA	91.96±1.12	9.59±0.45	**80.33±1.70**	10.04±0.68	**84.72±1.05**	9.61±0.50
D-AMD	**92.62±0.72**	9.52±0.34	**80.62±2.75**	10.20±0.82	**85.10±0.80**	9.68±0.33
	Gisette		Spam_corpus		Url_day0	
PAA-II	92.88±0.48	9.63±0.23	91.67±0.83	9.92±0.15	93.15±0.34	9.77±0.15
SOP	88.72±0.62	10.24±0.47	87.85±1.01	10.03±0.45	83.77±1.00	10.01±0.28
SOAL	93.89±0.76	9.73±0.23	94.83±0.39	9.65±0.25	94.63±0.28	9.81±0.17
D-ADA-I	**95.44±0.16**	10.79±0.18	**95.58±0.32**	9.49±0.33	95.77±0.10	9.38±0.17
D-AMD-I	**95.33±0.19**	9.49±0.28	95.36±0.40	9.58±0.30	**96.00±0.11**	9.51±0.29
	Basehock		Pcmac		Farm_ads	
PAA-II	91.49±0.86	19.04±0.58	82.04±1.64	19.22±0.51	86.72±0.39	19.61±0.46
SOP	86.95±0.97	19.22±0.77	75.26±1.45	19.61±0.52	82.47±0.70	20.14±0.69
SOAL	92.60±1.24	18.86±0.68	84.87±2.14	19.10±0.95	88.09±0.31	19.06±0.55
D-ADA	**95.00±0.42**	18.96±0.33	**86.57±0.74**	18.95±0.48	**88.42±0.41**	19.25±0.56
D-AMD	94.52±0.60	18.74±0.36	**86.61±0.65**	18.59±0.48	**88.36±0.38**	19.25±0.47
	Gisette		Spam_corpus		Url_day0	
PAA-II	94.10±0.21	19.50±0.27	93.31±0.45	19.69±0.16	94.57±0.11	19.52±0.14
SOP	91.12±0.40	20.27±0.62	90.64±0.62	19.80±0.41	87.39±0.37	19.99±0.41
SOAL	94.84±0.18	19.36±0.32	95.02±0.30	19.45±0.57	95.06±0.13	19.84±0.34
D-ADA-I	**95.61±0.15**	18.99±0.36	**95.76±0.33**	19.49±0.69	95.90±0.08	19.02±0.42
D-AMD-I	**95.51±0.20**	18.96±0.62	95.57±0.33	19.31±0.71	**96.11±0.10**	19.00±0.59

[1] The best result and its comparable ones (according to paired t-tests at 95% confidence level) on each dataset are displayed in bold and the p-values of all t-tests are below 0.05

in the fixed parameter setting, one can get consistent conclusions with that made in Sect. 5.1.3.

5.2 Evaluation of MD-ADA and MD-AMD for multiclass classification tasks

5.2.1 Multiclass classification datasets

Six multiclass datasets are chosen randomly to perform the experiments. These datasets are described in Table 4 and can be downloaded from LIBSVM website.[2]

[2] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multiclass.html.

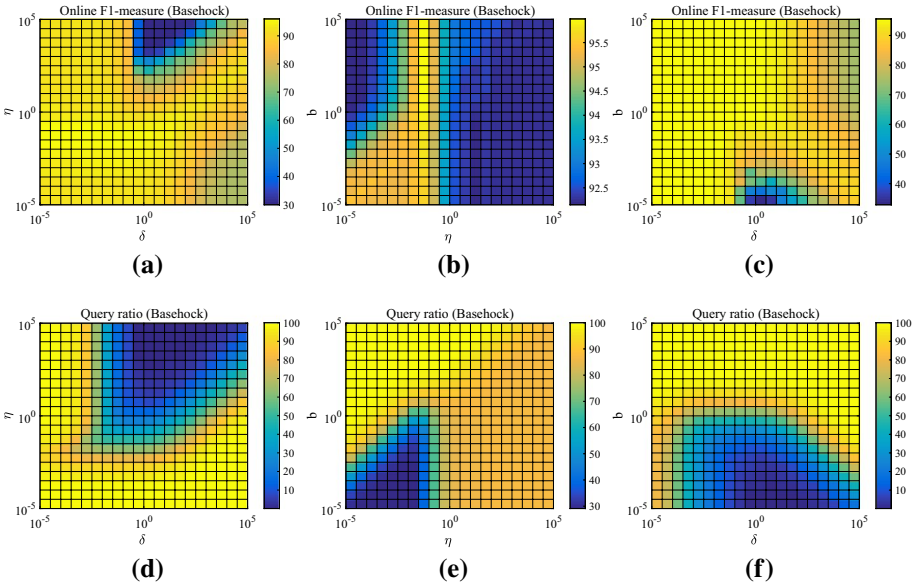

Fig. 4 Evaluation of the hyperparameter sensitivity for D-ADA on Basehock

Table 3 Fixed parameter values on each dataset

Algorithm	Parameter values (Dateset)
PAA-II	$C = 0.01$ (Basehock, Pcmac, Farm_ads, Spam_corpus, Url_day0)
	$C = 10^{-9}$ (Gisette)
SOAL	$\gamma = 100, \eta = 0.01$ (Basehock, Pcmac, Farm_ads, Url_day0)
	$\gamma = 1000, \eta = 0.01$ (Spam_corpus)
	$\gamma = 10^{9}, \eta = 10^{-8}$ (Gisette)
D-ADA / D-AMD	$\delta = 0.01, \eta = 0.01$ (Basehock, Pcmac, Farm_ads)
D-ADA-I / D-AMD-I	$\delta = 1, \eta = 0.1$ (Spam_corpus, Url_day0)
	$\delta = 100, \eta = 10^{-5}$ (Gisette)

5.2.2 Performance comparison

We have compared the following online multiclass active learning algorithms:

- MPAA-II (Lu et al., 2016b): Multiclass Passive Aggressive Active learning which uses the MPA-II updating rule and the multiclass margin-based label query strategy.
- MDA and MMD: the fully supervised versions of Algorithm 2, which query the labels of all incoming instances.
- MR-ADA and MR-AMD: use our multiclass updating rules, but the random label query strategy.

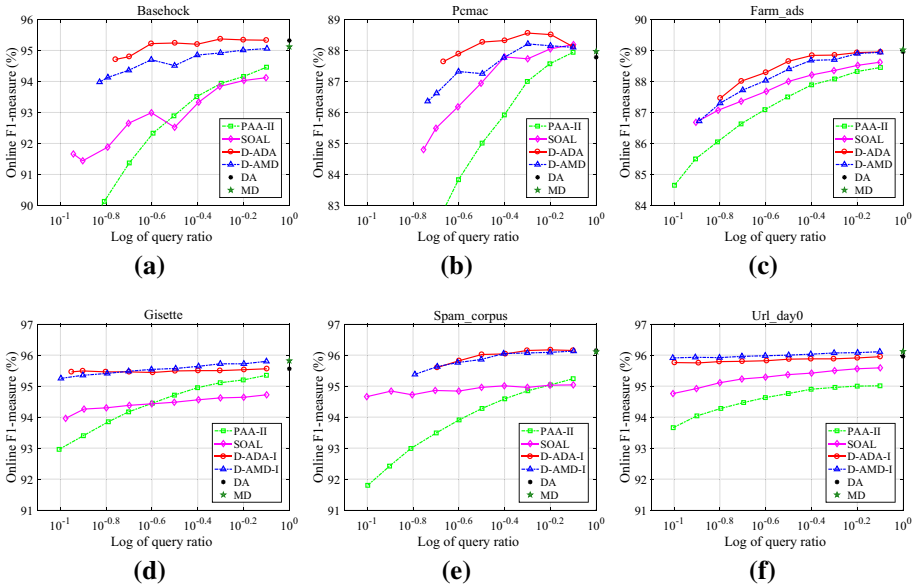

Fig. 5 Performance comparison in a fixed parameter setting

- MM-ADA and MM-AMD: use our multiclass updating rules, but the multiclass margin-based label query strategy. They are equivalent to Algorithm 2 that adopts $a_t = 0, \forall t \in [T]$.
- MD-ADA and MD-AMD: Algorithm 2 with $a_t = 1/\max\{1, \boldsymbol{x}_t^\top \boldsymbol{x}_t\}, \forall t \in [T]$.
- MD-ADA-I and MD-AMD-I: Algorithm 2 that adopts $a_t = 1, \forall t \in [T]$.

The experimental setting is similar to that for binary classification except that online accuracy is used for the performance metric. Figs. 6 and 7 present the online accuracy achieved by these algorithms at different query ratios. Note that one line (originally one point) is drawn for the fully supervised MDA and MMD. To clearly measure the performance difference, we also report in Table 5 the results at the fixed query ratio near 10^{-1} and $10^{-0.7}$.

From Figs. 6 and 7, we observe that MD-ADA-I and MD-AMD-I cannot achieve low query ratios on many datasets, but at those query ratios they can obtain, they mostly perform the best. Such a relationship of accuracy can be observed on all datasets: MD-ADA-I $\geq$ MD-ADA$\geq$ MM-ADA $>$ MR-ADA, and MD-AMD-I $\geq$ MD-AMD$\geq$ MM-AMD $>$ MR-AMD. The fact shows again the importance of exploiting the feature-based discrimination of instances in the query strategy. MM-ADA outperforms MPAA-II on four datasets and MM-AMD outperforms MPAA-II on all six datasets, which shows that our second-order updating rules generally lead to better performance than the first-order rule of MPAA-II. From Table 5, we further observe that MD-AMD-I or MD-AMD significantly outperform the other algorithms on all six datasets, according to paired t-tests at 95% confidence level. We also find that using the same label query strategy, M-MD updating tends to bring better performance than M-DA updating on these datasets. In conclusion, we discover that our updating rules, working together with our query strategy, can make very promising results on multiclass tasks.

Table 4 A summary of multiclass classification datasets

Dataset	# inst.	# fea.	# class	Dataset	# inst.	# fea.	# class	Dataset	# inst.	# fea.	# class
20newsgroups	18,846	26,214	20	Letter	15,000	16	26	Mnist	60,000	780	10
Connect4	67,557	126	3	Acoustic	78,823	50	3	Covtype	581,012	54	7

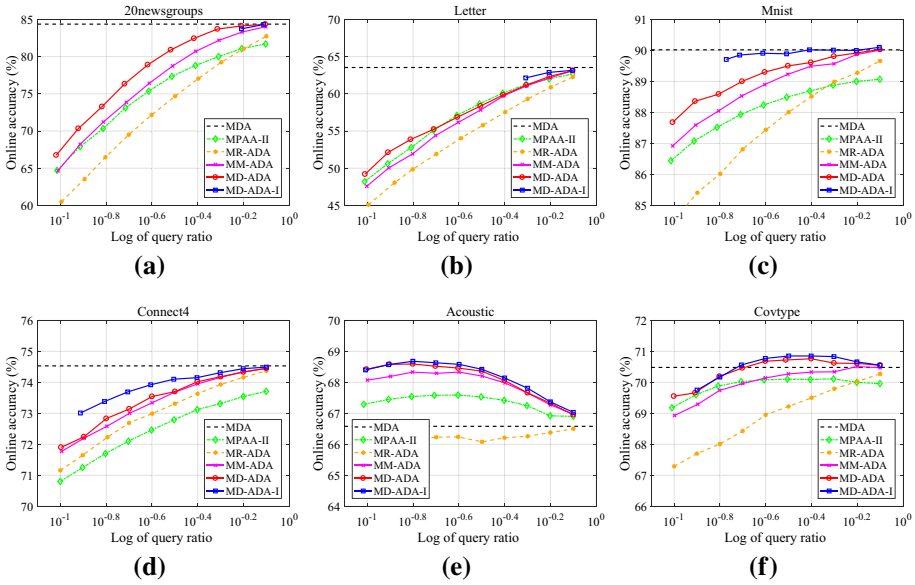

Fig. 6 Comparison of algorithms based on dual averaging update with existing methods

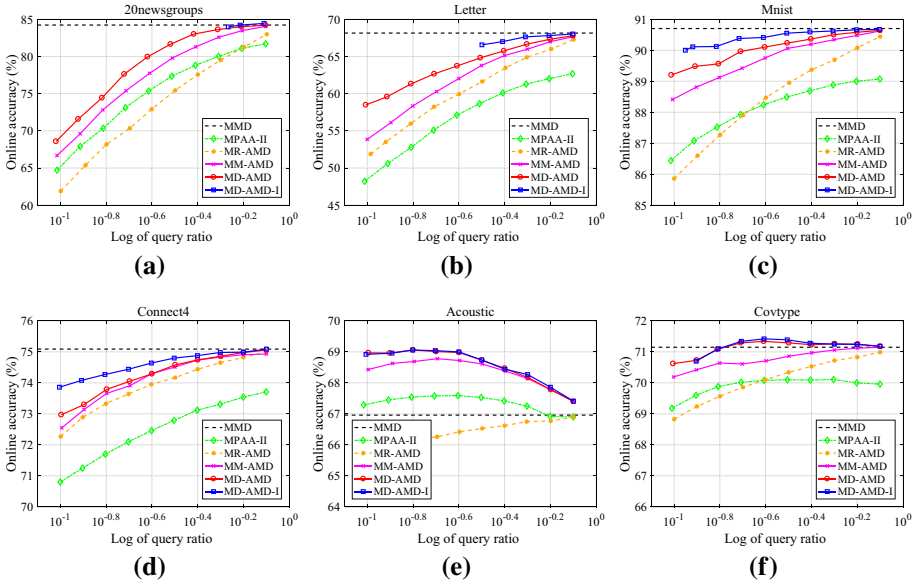

Fig. 7 Comparison of algorithms based on mirror descent update with existing methods

Table 5 Online accuracy obtained at the label query ratio near 10^{-1} and $10^{-0.7}$

Algorithm	accuracy (%)	query (%)	accuracy (%)	query (%)	accuracy (%)	query (%)
	20newsgroups		Letter		Mnist	
MPAA-II	64.71±0.41(3)	9.67±0.10	48.21±0.74(5)	9.74±0.18	86.44±0.10(5)	9.68±0.08
MR-ADA	60.50±0.79(5)	10.11±0.10	45.09±0.65(7)	10.11±0.11	84.54±0.24(7)	10.04±0.10
MM-ADA	64.66±0.65(3)	9.75±0.15	47.58±0.71(6)	9.95±0.29	86.92±0.14(4)	9.86±0.07
MD-ADA	66.75±0.55(2)	9.59±0.14	49.19±0.55(4)	9.78±0.20	87.68±0.10(3)	9.86±0.24
MR-AMD	61.90±0.80(4)	10.03±0.35	51.88±0.51(3)	10.35±0.00	85.86±0.24(6)	10.00±0.07
MM-AMD	66.67±0.60(2)	9.66±0.16	53.85±0.67(2)	10.00±0.27	88.41±0.12(2)	9.84±0.13
MD-AMD	**68.57±0.54**(1)	9.57±0.13	**58.48±0.66**(1)	9.85±0.20	**89.20±0.18**(1)	9.73±0.40
	Connect4		Acoustic		Covtype	
MPAA-II	70.80±0.18(7)	9.98±0.25	67.29±0.09(4)	9.65±0.27	69.18±0.15(4)	9.79±0.16
MR-ADA	71.16±0.17(6)	9.99±0.06	65.80±0.26(5)	9.89±0.06	67.29±0.07(6)	10.01±0.05
MM-ADA	71.77±0.18(5)	10.12±0.29	68.07±0.17(3)	10.02±0.28	68.93±0.12(5)	10.04±0.11
MD-ADA	71.90±0.12(4)	10.07±0.20	68.42±0.12(2)	9.93±0.33	69.55±0.11(3)	9.98±0.17
MR-AMD	72.27±0.20(3)	10.03±0.15	65.79±0.15(5)	9.93±0.03	68.83±0.10(5)	10.01±0.04
MM-AMD	72.55±0.27(2)	10.14±0.20	68.43±0.17(2)	10.09±0.38	70.19±0.10(2)	9.95±0.06
MD-AMD	**72.97±0.19**(1)	10.09±0.22	**68.97±0.16**(1)	10.12±0.40	**70.62±0.13**(1)	9.90±0.36
	20newsgroups		Letter		Mnist	
MPAA-II	73.12±0.25(5)	19.25±0.19	55.09±0.58(4)	19.49±0.22	87.93±0.09(7)	19.51±0.11
MR-ADA	69.50±0.48(7)	19.88±0.08	51.92±0.58(6)	20.06±0.17	86.81±0.14(8)	19.94±0.08
MM-ADA	73.85±0.31(4)	19.42±0.26	54.40±0.64(5)	19.88±0.27	88.53±0.10(6)	19.72±0.14
MD-ADA	76.33±0.42(2)	19.08±0.23	55.24±0.34(4)	19.54±0.37	89.00±0.08(5)	19.82±0.34
MD-ADA-I	–	–	–	–	89.85±0.09(3)	19.33±0.41
MR-AMD	70.34±0.50(6)	20.10±0.30	58.23±0.53(3)	19.58±0.00	87.90±0.14(7)	19.99±0.16
MM-AMD	75.39±0.23(3)	19.35±0.16	60.28±0.35(2)	20.07±0.35	89.42±0.09(4)	19.81±0.15
MD-AMD	**77.62±0.28**(1)	19.01±0.20	**62.65±0.42**(1)	19.62±0.28	89.96±0.12(2)	19.54±0.62
MD-AMD-I	-	-	-	-	**90.38±0.07**(1)	19.22±0.70
	Connect4		Acoustic		Covtype	
MPAA-II	72.10±0.12(8)	19.81±0.21	67.58±0.08(6)	19.64±0.21	70.02±0.04(5)	19.62±0.13
MR-ADA	72.69±0.14(7)	19.94±0.11	66.23±0.17(7)	20.00±0.11	68.43±0.07(8)	19.94±0.05
MM-ADA	73.00±0.11(6)	20.04±0.18	68.29±0.11(5)	20.06±0.25	69.96±0.05(6)	19.95±0.11
MD-ADA	73.14±0.11(5)	20.00±0.19	68.52±0.10(4)	19.87±0.34	70.46±0.10(4)	19.87±0.47
MD-ADA-I	73.69±0.07(4)	19.68±0.19	68.63±0.09(3)	19.93±0.33	70.56±0.06(3)	19.67±0.39
MR-AMD	73.64±0.10(4)	19.84±0.04	66.26±0.16(7)	20.18±0.10	69.86±0.05(7)	19.99±0.05
MM-AMD	73.91±0.14(3)	20.13±0.26	68.78±0.13(2)	20.26±0.43	70.61±0.10(3)	19.83±0.12
MD-AMD	74.05±0.09(2)	20.06±0.19	**69.01±0.16**(1)	20.02±0.35	71.29±0.05(2)	19.52±0.87
MD-AMD-I	**74.44±0.09**(1)	19.85±0.31	**69.04±0.12**(1)	19.82±0.44	**71.34±0.05**(1)	19.68±0.25

[1] "–" represents that the algorithm cannot attain the query ratio.

[2] The best result and its comparable ones (paired t-tests at 95% confidence level) are displayed in bold and the p-values of all t-tests are below 0.05. The number in brackets shows the ranking of each algorithm

6 Conclusion

In this paper, two novel online active learning algorithms for binary classification, called D-ADA and D-AMD, have been proposed and analyzed. Both algorithms maintain a diagonal matrix for recording the updating information of all dimensions and exploit the matrix to endow different dimensions with adaptive learning rates. Especially, D-ADA uses the dual averaging idea to update its predictor, while D-AMD uses the mirror descent idea. In order to identify critical instances to label, different from the usual margin-based methods that only use the predictive uncertainty of instances, D-ADA and D-AMD also take full advantage of the feature-based discriminative information of instances. Further, D-ADA and D-AMD have been extended to the multiclass classification setting. The expected mistake bounds for our proposed algorithms are provided, which show that when the label query ratio exceeds a certain value, our active learning algorithms are asymptotically comparable to the best fixed fully supervised classifier chosen in hindsight. Experiments on six high-dimensional binary classification datasets corroborate the merits of our label query strategy and demonstrate that D-ADA and D-AMD outperform existing second-order and first-order active learning methods, at various label query ratios. Experiments on six multiclass classification datasets also show the superiority of our multiclass active learning algorithms. In the future, it is interesting to investigate how to extend our methods to the multi-label classification setting and the cost-sensitive setting.

Author contributions Conceptualization: T.Z., F.K.; Methodology: T.Z.; Formal analysis and investigation: T.Z., F.K.; Writing - original draft preparation: T.Z.; Writing - review and editing: F.K., T.Z.; Funding acquisition: T.Z., J.Z., B.L.; Resources: Y.G., J.Z, B.L.; Supervision: Y.G., J.Z., B.L.

Funding This work is supported by National Natural Science Foundation of China (Nos. 61906165, 61872313, 61972335) and Natural Science Foundation of the Jiangsu Higher Education Institutions of China (Nos. 19KJB520064).

Data availability The hyperlinks of the websites for downloading the open datasets that we used have been provided in the paper.

Declarations

Conflict of interest The authors declare that they have no conflicts of interest.

Ethics approval The authors declare that the submitted work is original research that has not been published previously, and not under consideration for publication elsewhere, in whole or in part. All the authors listed have approved the manuscript that is enclosed.

Code availability https://github.com/LUCKY-ting/online_active_learning.

References

Awasthi, P., Balcan, M., Haghtalab, N., & Urner, R. (2015). Efficient learning of linear separators under bounded noise. In *Proceedings of the 28th Conference on Learning Theory, Paris, France*, vol 40 (pp. 167–190).

Balcan, M., & Long, P. M. (2013). Active and passive learning of linear separators under log-concave distributions. In *Proceedings of the 26th Annual Conference on Learning Theory, Princeton University, NJ, USA*, vol 30 (pp. 288–316).

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

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

Crammer, K., Dredze, M., & Pereira, F. (2012). Confidence-weighted linear classification for text categorization. *Journal of Machine Learning Research, 13*, 1891–1926.

Crammer, K., Kulesza, A., & Dredze, M. (2013). Adaptive regularization of weight vectors. *Machine Learning, 91*(2), 155–187.

Demir, B., & Bruzzone, L. (2014). A multiple criteria active learning method for support vector regression. *Pattern Recognition, 47*(7), 2558–2567.

Du, B., Wang, Z., Zhang, L., Zhang, L., Liu, W., Shen, J., & Tao, D. (2017). Exploring representativeness and informativeness for active learning. *IEEE Transactions on Cybernetics, 47*(1), 14–26.

Duchi, J. C., Hazan, E., & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research, 12*, 2121–2159.

Golovin, D., Krause, A., & Ray, D. (2010). Near-optimal bayesian active learning with noisy observations. In *Advances in Neural Information Processing Systems* (pp. 766–774).

Hanneke, S. (2014). Theory of disagreement-based active learning. *Foundations and Trends in Machine Learning, 7*(2–3), 131–309.

Hao, S., Lu, J., Zhao, P., Zhang, C., Hoi, S. C. H., & Miao, C. (2018). Second-order online active learning and its applications. *IEEE Transactions on Knowledge and Data Engineering, 30*(7), 1338–1351.

Hazan, E., Agarwal, A., & Kale, S. (2007). Logarithmic regret algorithms for online convex optimization. *Machine Learning, 69*(2–3), 169–192.

Hoi, S. C. H., Jin, R., Zhao, P., & Yang, T. (2013). Online multiple kernel classification. *Machine Learning, 90*(2), 289–316.

Huang, S., Jin, R., & Zhou, Z. (2014). Active learning by querying informative and representative examples. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 36*(10), 1936–1949.

Katakis, I., Tsoumakas, G., Banos, E., Bassiliades, N., & Vlahavas, I. P. (2009). An adaptive personalized news dissemination system. *Journal of Intelligent Information Systems, 32*(2), 191–212.

Lu, J., Hoi, S. C. H., Wang, J., Zhao, P., & Liu, Z. (2016a). Large scale online kernel learning. *Journal of Machine Learning Research, 17*, 47:1-47:43.

Lu, J., Zhao, P., & Hoi, S. C. H. (2016). Online passive-aggressive active learning. *Machine Learning, 103*(2), 141–183.

Lughofer, E. (2017). On-line active learning: A new paradigm to improve practical useability of data stream modeling methods. *Information Sciences, 415*, 356–376.

Luo, H., Agarwal, A., Cesa-Bianchi, N., & Langford, J. (2016). Efficient second order online learning by sketching. In *Advances in Neural Information Processing Systems* (pp. 902–910).

Ma, J., Saul, L. K., Savage, S., & Voelker, G. M. (2009). Identifying suspicious urls: an application of large-scale online learning. In Proceedings of the 26th International Conference on Machine Learning, Montreal, Quebec, Canada (pp. 681–688).

Settles, B. (2009). Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin-Madison.

Shalev-Shwartz, S. (2012). Online learning and online convex optimization. *Foundations and Trends in Machine Learning, 4*(2), 107–194.

Shalev-Shwartz, S., Singer, Y., Srebro, N., & Cotter, A. (2011). Pegasos: Primal estimated sub-gradient solver for SVM. *Mathematical Programming, 127*(1), 3–30.

Song, Q., Xu, Z., Fan, H., & Wang, D. (2017). Robust recurrent kernel online learning. *IEEE Transactions on Neural Networks and Learning Systems, 28*(5), 1068–1081.

Sun, Y., Tang, K., Minku, L. L., Wang, S., & Yao, X. (2016). Online ensemble learning of data streams with gradually evolved classes. *IEEE Transactions on Knowledge and Data Engineering, 28*(6), 1532–1545.

Tosh, C., & Dasgupta, S. (2017). Diameter-based active learning. In Proceedings of the 34th International Conference on Machine Learning, vol 70 (pp. 3444–3452).

Wang, Z., & Ye, J. (2015). Querying discriminative and representative samples for batch mode active learning. *ACM Transactions on Knowledge Discovery from Data, 9*(3), 17:1-17:23.

Zhai, T., Gao, Y., Wang, H., & Cao, L. (2017). Classification of high-dimensional evolving data streams via a resource-efficient online ensemble. *Data Mining and Knowledge Discovery, 31*(5), 1242–1265.

Zhai, T., Koriche, F., Wang, H., & Gao, Y. (2019). Tracking sparse linear classifiers. *IEEE Transactions on Neural Networks and Learning Systems, 30*(7), 2079–2092.

Zhang, C. (2018). Efficient active learning of sparse halfspaces. In Proceeding of the 31st Conference on Learning Theory, Stockholm, Sweden, vol 75 (pp. 1856–1880).

Zhao, P., & Hoi, S. C. H. (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, Chicago, IL, USA (pp. 919–927).

Zinkevich, M. (2003). Online convex programming and generalized infinitesimal gradient ascent. In Proceedings of the 20th International Conference on Machine Learning, Washington, DC, USA (pp. 928–936).