Online Asymmetric Active Learning with Imbalanced Data

Xiaoxuan Zhang, Tianbao Yang, Padmini Srinivasan
Department of Computer Science
The University of Iowa, IA 52242
{xiaoxuan-zhang, tianbao-yang, padmini-srinivasan}@uiowa.edu

ABSTRACT
This paper considers online learning with imbalanced streaming data under a query budget, where the act of querying for labels is constrained to a budget limit. We study different active querying strategies for classification. In particular, we propose an asymmetric active querying strategy that assigns different probabilities for query to examples predicted as positive and negative. To corroborate the proposed asymmetric query model, we provide a theoretical analysis on a weighted mistake bound. We conduct extensive evaluations of the proposed asymmetric active querying strategy in comparison with several baseline querying strategies and with previous online learning algorithms for imbalanced data. In particular, we perform two types of evaluations according to which examples appear as "positive"/"negative". In push evaluation only the positive predictions given to the user are taken into account; in push and query evaluation the decision to query is also considered for evaluation. The push and query evaluation strategy is particularly suited for a recommendation setting because the items selected for querying for labels may go to the end-user to enable customization and personalization. These would not be shown any differently to the end-user compared to recommended content (i.e., the examples predicated as positive). Additionally, given our interest in imbalanced data we measure $F$-score instead of accuracy that is traditionally considered by online classification algorithms. We also compare the querying strategies on five classification tasks from different domains, and show that the probabilistic query strategy achieves higher $F$-scores on both types of evaluation than deterministic strategy, especially when the budget is small, and the asymmetric query model further improves performance.

When compared to the state-of-the-art cost-sensitive online learning algorithm under a budget, our online classification algorithm with asymmetric querying achieves a higher $F$-score on four of the five tasks, especially on the push evaluation.

1. INTRODUCTION
Traditional batch classification algorithms that have been broadly applied in various data mining domains, such as document filtering [29], news classification [23], spam detection [2], and opinion mining [25], are challenged by applications characterized by large-scale, streaming data as for instance in web mining applications. Large amounts of data are continually generated by the web such as through social media and news. These datasets are often also highly imbalanced with respect to classes of interest. A key consequence of scale is that getting adequate training data becomes non-trivial in effort and costly. To address these problems, we study online classification algorithms where the act of querying for labels is constrained to a budget limit.

Lowering cost is always a goal in algorithms processing large-scale and real-time data. In classification, there are two major sources of cost. The first is the obvious cost incurred due to errors in performance (false positive and false negative decisions). The second is the cost of labeling the data used to initially build or retrain the model over time. Focusing on the second cost for the moment, labelled data may be collected in two general ways with cost differences that are both subtle and explicit. We may directly ask the client (end-user) to provide labels. This approach is particularly useful for personalized recommendation [8, 1]. As a consequence, in addition to giving the client the high confidence positive predictions made by the system we also show instances of low confidence to label. While risky these low confidence instances are likely to be the most useful for improving the performance of the classifier. However, over time the client may become disappointed if the system takes big risks with too many false positives shown. Therefore, it makes sense to limit the amount of queries to label sent to the end-user. In this paper, we tackle this issue in an online setting for streaming data. We aim to maximize the performance subject to a budget limit for querying the labels.

There are several fold of entangled difficulties: (i) how to decide which examples to query for the true labels given a budget limit; (ii) which performance measure should we target? (iii) how to evaluate the performance of the system? These difficulties become severe in the presence of imbalanced data. For example, if there are many more negative examples than positive examples, treating them equally for making the query decisions can be sub-optimal. A 'bad' query may even harm performance. For example, Figure 2
shows the results of two algorithms (the red and blue lines) on the OHSUMED dataset as discussed in the experiment section. The blue line queries the first 3,000 instances for labels then it stops querying. In contrast the red line algorithm keeps querying for labels of the examples that are predicted as positive. It shows that although the red line algorithm queries more labels it performs worse. The reason will be discussed in the experiment section. Moreover, the traditional performance measure by error rate is not appropriate for imbalanced data. Furthermore, the examples for label querying may also be pushed to the end-user in the same way as the recommended items and thus should be also taken into account in evaluation.

The contributions of the paper are three-fold. Firstly, we propose an asymmetric active querying strategy, which is randomized in nature and assigns different probabilities for querying to examples that are predicted as positive and negative. We also provide a rigorous theoretical analysis of the proposed asymmetric query strategy comparing with previous symmetric query strategy. Secondly, we evaluate the performance of different algorithms by two methods. In the first performance is assessed both in terms of instances predicted as positives (pushed to client) and instances shown for labeling (querying the client). i.e., both instances that are positive predictions by the classifier and instances selected for label querying are shown to the user as positive predictions. This evaluation is particularly suited when the client is the source of the labeling. In the second evaluation, performance is only based on the positive predictions made by the classifier. This traditional evaluation method is more suitable when the queried labels are from another source, e.g., internal workers (or crowd-workers). We refer to the first evaluation as 'push and query' and the second as 'push' evaluation. These two evaluations are in essence based upon what the user 'sees', and are illustrated in Figure 1. In our experiments, we find that the proposed algorithm performs well on both evaluations. Lastly, we evaluate the performance by F-1 score, which is more suited for imbalanced data. Most existing studies of online classification or learning mainly minimize error rate (or maximize accuracy). Lower error rate is always good. However, we are interested in working with highly imbalanced data where error rate is not meaningful. If 90% of the examples are negative, simply classifying all of the dataset as negative will give an error rate of 0.1, which looks excellent but is actually meaningless. Therefore, we choose to use the F score as the performance metrics. In the paper, we will also show that the proposed asymmetric querying method also favors the F score.

The remainder of the paper is organized as follows. We first discuss the related work of online learning and online active classification in Section 2. Then we describe and analyze the algorithms in Section 3, including four algorithms with distinct query strategies and the variant with asymmetric query on positive and negative examples. The experimental design and results are discussed in Section 4 and the last section will be the discussion and conclusions.

2. RELATED WORK

Many online algorithms have been developed, e.g., the Perceptron algorithm [27], the exponentially weighted average algorithm [3, 22] and the online gradient descent [35, 16]. In the last ten years, we observe substantial applications of these algorithms in machine learning and data analytics, e.g., online classification [21, 15, 10, 19, 9]. Traditional online classification aims to minimize the number of mistakes of different classes [32, 14, 24, 28]. Batch learning with cost-sensitivity has been studied a lot, while few studies are devoted to online learning with imbalanced data [9, 31, 34]. These few studies either modify conventional loss functions to incorporate the given/defined cost matrix or use a different loss function to optimize a measure that is suited for imbalanced data. However, they also assume full knowledge of the label information for all received examples. This renders them unattractive for mining massive streaming data where querying for the labels is subject to a cost.

Online learning under a query budget has received little attention. Several papers have studied a similar problem in a context also known as label-efficient online learning or online active learning. Cesa-Bianchi, Lugosi, and Stoltz [6] studied the problem of learning from expert advice under a query budget. They proposed a simple strategy that uses an independent and identically distributed (i.i.d.) sequence $Z_1, \ldots, Z_T$ of Bernoulli random variables such that $\Pr(Z_t = 1) = \varepsilon$ and asks the label to be revealed whenever
\(Z_t = 1\). The limitation with the pure random query strategy is that it does not differentiate between examples with high confidence score and low confidence score for classification. Later on, the same authors [7] designed a new strategy of query that makes the probability of querying dependent on the absolute value of the prediction score. This query algorithm has also been used in a recent work for cost-sensitive online learning [33] Active learning for querying the labels has also been considered in different works for different algorithms from the perspective of sample complexity [11, 12, 4]. [11] analyzed the perceptron-like algorithm under an active setting and provided a complexity on the number of queried labels for achieving a certain generalization performance. [4, 12] studied online ridge-regression type of algorithms for classification for different querying strategies and established the sample complexity bound. However, these works have not consider the asymmetry between positive examples and negative examples for imbalanced data.

One of our goals in this paper is to compare different query strategies for learning with imbalanced data. Moreover, we note that the symmetric query model where the probability of querying is independent of the positive/negative decision is not well suited for imbalanced data. To understand this, consider that the number of positive examples is much smaller than the number of negative examples. As a result, there would be more false positives than false negatives. If we assign equal probabilities to these false predicted examples for querying the label, the algorithm would favor the negative class more than the positive class, consequentially harm the performance. Therefore, we propose a novel asymmetric query model that is demonstrated to be sound in theory and effective in practice as well. Moreover, the comparison of two different strategies to handle imbalanced data under a query budget, namely (i) an asymmetric query model plus a symmetric updating rule of the proposed algorithm, and (ii) an asymmetric updating rule plus a symmetric query model of a state-of-the-art algorithm [33], also demonstrate the proposed algorithm is very useful.

3. ONLINE ACTIVE LEARNING UNDER A QUERY BUDGET

We first introduce some notations. We denote by \(x_t \in \mathbb{R}^d\) the feature vector of the example received at the \(t\)-th round, and by \(y_t \in \{1, -1\}\) the label of \(x_t\). Let \(f(x)\) denote a prediction function. In the sequel, we will focus on the presentation using the linear model \(f(x) = w^\top x\), while one can easily generalize it to a non-linear function in a reproducing kernel Hilbert space. Let \(B > 0\) denote a budget limit on the number of queries.

The framework of online classification under a query budget is presented in Algorithm 1. We let \(w_t\) and \(B_t\) denote the model available before the \((t+1)\)-th round and the budget used before the \((t+1)\)-th round. Initially, the model is \(w_0 = 0\) and \(B_0 = 0\). In line 5, the algorithm computes \(p_t = w_\top_t x_t\) and in line 6 it makes a decision about the binary label of \(x_t\) by a function \(\hat{y}_t = \text{Predict}(p_t)\) based on \(p_t\). The simplest \texttt{Predict} function is \(\hat{y}_t = \text{sign}(p_t)\), i.e., using the sign of \(p_t\) to determine the label. More generally, we can use a threshold \(\gamma\) and and let \(\hat{y}_t = \text{sign}(p_t - \gamma)\). For imbalanced data, we observe that using a threshold always yields better performance. We discuss how to set the value of \(\gamma\) in the experiment section.

After the binary decision is made, the algorithm enters the query stage, where it decides whether to query the label and update the model. When the algorithm reaches the budget limit, the model will not be updated because query is not allowed and there are two options for making future predictions. The Option I is to use the last updated model and the Option II is to use the averaged model before the iteration \(T_B\) when the budget is used up. Many previous studies have found that the averaged model might give more robust predictions than the last model [30].

If the remaining budget is not zero, i.e., \(B_{t-1} < B\), we use the output of a query function \(Z_t = \text{Query}(p_t)\), which might depend on \(p_t\) and some other parameters, to determine query \((Z_t = 1)\) or not \((Z_t = 0)\). If \(Z_t = 1\), then the algorithm queries the label of current example denoted by \(y_t \in \{1, -1\}\) from a source. Then the algorithm proceeds to update the model using \(w_t = \text{Update}(w_{t-1}, y_t, x_t)\) in line 13. Function \texttt{Update} depends on what optimization method is employed and what surrogate loss function is assumed. Indeed, many different updating schemes can be used, including the margin-based updating rules (e.g., Perceptron [27], online passive-aggressive algorithm [9], confidence-weighted learning algorithm [13], etc) and online gradient descent updates for different types of loss functions [35]. Take the Perceptron for example, \(w_t\) is updated by

\[
w_t = w_{t-1} + y_t x_t \mathbb{I}(y_t f_t \leq 0)
\]

where \(\mathbb{I}(v)\) is an identity function that outputs 1 if \(v\) is true and otherwise outputs zero.

The query model is the key concern in this paper. Below we first discuss some baseline models that either arise straightforwardly or appear in previous work. Then we present the proposed query model for imbalanced data.

3.1 Baseline Query Models

We discuss several baseline query models below and comment on their deficiencies.

- **First Come First Query (referred to as QF).** This strategy is simply to query the true labels of the first \(B\) examples and then uses the model learned from the first \(B\) examples to make predictions for all the following examples. This strategy is similar to the \(\epsilon\)-greedy strategy used in bandit learning [20], where the first stage with \(B\) examples is devoted to exploration and the second stage is exploitation.

- **Random Query (referred to as QR).** This strategy is to use a Bernoulli random variable \(Z_t = \text{Bern}(\epsilon)\) that equals 1 with a probability \(\epsilon\) to determine whether to query or not. The random strategy has been used in predict with expert advices under budget feedback setting [5].

- **Deterministic prediction dependent Query (referred to as QP).** Different from the first two query models where the decision to query does not depend on \(p_t\), we can make the query function dependent on \(p_t\). The motivation is that if \(p_t\) is closer to zero, meaning that the prediction is more uncertain, then we should query its label for updating the model. This strategy is very similar to active learning where uncertain examples should be queried first [18]. The deterministic predic-
Algorithm 1 Online Binary Classification Under A Query Budget

1: Input: budget $B$
2: Initialize $w_0 = 0$, $B_0 = 0$
3: for $t = 1, \ldots, T$ do
4:  Receive an example $x_t$
5:  Compute $p_t = w_{t-1}^\top x_t$
6:  Make a push decision $\hat{y}_t = \text{Predict}(p_t) \in \{1, -1\}$
7:  if $B_{t-1} \geq B$ then
8:      Set $w_t$ as
9:      Option I: $w_t = w_{t-1}$
10:     Option II: $w_t = \frac{1}{T_B} \sum_{t=0}^{T_B-1} w_t$
11:  end if
12: end for

where $T_B$ is the smallest number that the budget is used up, i.e., $B_{T_B-1} = B$.

else
13:  Compute a variable $Z_t = \text{Query}(p_t) \in \{1, 0\}$
14:  if $Z_t = 1$ then
15:     Query the true label $y_t$ from a source $S$
16:  end if
17:  Update the model $w_t = \text{Update}(w_{t-1}, y_t, x_t)$
18:  Update $B_t = B_{t-1} + 1$
19: end if
20: end for

### 3.2 Asymmetric Query Model

The issue of the randomized symmetric query model is to treat the positive examples and the negative examples equally. For imbalanced data, this will be vulnerable to the majority class (e.g., the negative class). If the negative class is the majority class, positive examples will be more likely to be predicted as negative. Therefore, intuitively for the sake of learning the model, it is better to query more false negative examples than false positive examples, i.e., making the query asymmetric. To quantify this, we propose the following asymmetric query model, referred to as $Q_A$:

$$\text{Pr}(Z_t = 1) = \begin{cases} \frac{c}{|p_t| + c} & \text{if } p_t \geq 0 \\ \frac{c}{|p_t| + c} & \text{otherwise} \end{cases}$$

We establish below the weighted mistake bound of Algorithm 1 using the asymmetric query model. The proof is deferred to the supplement due to the limits of space.

**Theorem 2.** Let $T_B$ be the smallest number that $B_{T_B-1} = B$. If we run Algorithm 1 using Eqn. (4) as the query model, then for all $u \in \mathbb{R}^d$ and for all $\xi > 0$, the expected weighted number of mistakes up to $T_B$ satisfies

$$\mathbb{E} \left[ \sum_{t=1}^{T_B} \mathbb{I}_{\text{sign}(p_t) \neq y_t} \right] \leq \alpha \left\{ \sum_{y_t = 1} c_- \mathbb{I}_{\text{sign}(p_t) \neq y_t} + \sum_{y_t = -1} c_+ \mathbb{I}_{\text{sign}(p_t) \neq y_t} \right\} + \frac{\alpha^2}{2} \|u\|^2$$

where $\alpha = \max\left\{ \frac{c_+ + R^2/2}{\xi_+}, \frac{c_- + R^2/2}{\xi_-} \right\}$ with $R$ being the upper bound of data norm, i.e., $\max\|x_t\|_2 \leq R$.

**Remark:** Compared to Theorem 1, there are two key differences: (i) Theorem 2 is bounding the weighted number of mistakes, where the false negative is weighted by $c_-$ and false positive is weighted by $c_+$; (ii) the mistake bound in Theorem 2 is compared to the optimal loss that is defined using different margin $\xi_+$ and $\xi_-$ for positive and negative examples, respectively. It is these differences that render the flexibility of Algorithm 1 in balancing between false negative and false positive. Accordingly, we refer to Algorithm 1 using Eqn. (4) as the asymmetric query model 2. In the next section, we present the mistake bound of Algorithm 1 using the symmetric query model $Q_S$ in the following theorem.

**Theorem 1.** Let $T_B$ be the smallest number that $B_{T_B-1} = B$. If we run Algorithm 1 using Eqn. (3) as the query model, then for all $u \in \mathbb{R}^d$ and for all $\xi > 0$, the expected number of mistakes up to $T_B$ satisfies

$$\mathbb{E} \left[ \sum_{t=1}^{T_B} \mathbb{I}_{\text{sign}(p_t) \neq y_t} \right] \leq \frac{\alpha}{c} \sum_{t=1}^{T_B} \xi \left( y_t u^\top x_t \right) + \frac{\alpha^2}{2c} \|u\|^2$$

where $\xi(z) = \max\{0, \xi - z\}$ is a hinge loss parameterized by a margin parameter $\xi > 0$, and $\alpha = \frac{c+R^2/2}{\xi}$ with $R$ being the upper bound of data norm, i.e., $\max\|x_t\|_2 \leq R$. **Remark:** Since the upper bound holds for any $u$, we can minimize the upper bound by choosing the best $u$. Note that we only establish the number of mistake bound up to $T_B$ since the model will keep the same after that and its performance is determined by examples received before iteration $T_B$. The proof can be found in [5]. For completeness, we include a proof in the appendix.
and false positive for imbalanced data. Hence, it achieves the similar affect as using different costs for false negative and false positive, which has been widely adopted in previous studies on learning from imbalanced data. In particular, if the negative class is the dominant class, then it is expected that $c_-$ should be set to a larger value than $c_+$. This phenomenon has been observed in our experiments, which validates the result in Theorem 2.

**Proof.** We denote by $\hat{y}_t = \text{sign}(p_t)$ and introduce the Bernoulli random variable $M_t = I_{\hat{y}_t \neq y_t}$. Consider now a round $t$ where the algorithms queries a label and makes a mistake, i.e., $M_t Z_t = 1$. We consider two scenarios. First if $p_t \geq 0$, then we have for any $u \in \mathbb{R}^d$ and $\xi_+ > 0$,

$$
\xi_+ - \mathbb{E}[\xi_+ (y_t u^T x_t)] = \xi_+ - \max(0, \xi_+ - y_t u^T x_t) \\
\leq y_t u^T x_t = y_t (u - w_{t-1} + w_{t-1})^T x_t \\
= y_t w_{t-1} x_t + \frac{1}{2} \|u - w_{t-1}\|^2 - \frac{1}{2} \|u - w_{t-1}\|^2 + \frac{1}{2} \|w_{t-1} - w_t\|^2
$$

where we use the fact $w_t = w_{t-1} + y_t x_t$ for $M_t Z_t = 1$. Since $y_t \neq \hat{y}_t$, then $y_t w_{t-1} x_t \leq 0$. Replacing $u$ by $\alpha u$ with $\alpha > 0$ and reorganize the inequality we have

$$(\alpha \xi_+ + |p_t|)M_t Z_t \leq \frac{1}{2} \|\alpha u - w_{t-1}\|^2 + \frac{1}{2} \|w_{t-1} - w_t\|^2$$

We note that the above inequality holds for all rounds such that $\hat{y}_t = 1$. If $M_t Z_t = 0$, then $w_t = w_{t-1}$, and the above inequality holds because $\alpha \xi_+ (y_t u^T x_t) \geq 0$.

Similarly, if $p_t < 0$, then for any $u \in \mathbb{R}^d$ and $\xi_- > 0$,

$$(\alpha \xi_- + |p_t|)M_t Z_t \leq \frac{1}{2} \|\alpha u - w_{t-1}\|^2 + \frac{1}{2} \|w_{t-1} - w_t\|^2 + \frac{1}{2} \|w_t - w_{t-1}\|^2$$

The above inequality holds for all rounds such that $\hat{y}_t = -1$. Summing the above inequality over $t = 1, \ldots, T_\theta$, we have

$$
\sum_{y_t = 1} (\alpha \xi_+ + |p_t|)M_t Z_t + \sum_{y_t = -1} (\alpha \xi_- + |p_t|)M_t Z_t \\
\leq \alpha \left[ \sum_{y_t = -1} \ell_{\xi_+} (y_t u^T x_t) + \sum_{y_t = 1} \ell_{\xi_-} (y_t u^T x_t) \right] \\
+ \frac{1}{2} \|\alpha u - w_0\|^2 + \frac{1}{2} \sum_{t = 1}^{T_\theta} \|w_t - w_{t-1}\|^2
$$

In the first inequality, on the right hand side, we use the summation over $y_t = 1$ and $y_t = -1$. The inequality holds because when $M_t Z_t = 1$, $\hat{y}_t = 1$ indicates $y_t = 1$, and $\hat{y}_t = -1$ indicates $y_t = -1$. Then,

$$
\sum_{y_t = 1} (\alpha \xi_+ + |p_t| - \frac{R^2}{2})M_t Z_t + \sum_{y_t = -1} (\alpha \xi_- + |p_t| - \frac{R^2}{2})M_t Z_t \\
\leq \alpha \left[ \sum_{y_t = -1} \ell_{\xi_+} (y_t u^T x_t) + \sum_{y_t = 1} \ell_{\xi_-} (y_t u^T x_t) \right] \\
+ \frac{1}{2} \|\alpha u - w_0\|^2 + \frac{1}{2} \sum_{t = 1}^{T_\theta} \|w_t - w_{t-1}\|^2 + \frac{1}{2} \sum_{t = 1}^{T_\theta} \|w_t - w_{t-1}\|^2 M_t Z_t
$$

Since $\alpha = \max \left\{ \frac{c_+ + R^2/2}{\xi_+}, \frac{c_- + R^2/2}{\xi_-} \right\}$, therefore $\alpha \xi_+ \geq c_+ + R^2/2$ and $\alpha \xi_- \geq c_- + R^2/2$, then have

$$
\sum_{y_t = 1} (c_+ + |p_t|)M_t Z_t + \sum_{y_t = -1} (c_- + |p_t|)M_t Z_t \\
\leq \alpha \left[ \sum_{y_t = -1} \ell_{\xi_+} (y_t u^T x_t) + \sum_{y_t = 1} \ell_{\xi_-} (y_t u^T x_t) \right] + \frac{\alpha^2}{2} \|u\|^2
$$

By taking expectation over randomness in $Z_t$ and noting that $E[Z_t] = \frac{c_+ + |p_t|}{c_+ + |p_t|}$ for $\hat{y}_t = 1$ and $E[Z_t] = \frac{c_- + |p_t|}{c_- + |p_t|}$ for $\hat{y}_t = -1$, we have

$$
E \left[ \sum_{y_t = 1} c_+ I_{\hat{y}_t \neq y_t} + \sum_{y_t = -1} c_- I_{\hat{y}_t \neq y_t} \right] \\
\leq \alpha \left[ \sum_{y_t = -1} \ell_{\xi_+} (y_t u^T x_t) + \sum_{y_t = 1} \ell_{\xi_-} (y_t u^T x_t) \right] + \frac{\alpha^2}{2} \|u\|^2
$$

i.e.,

$$
E \left[ \sum_{y_t = 1} c_+ I_{\hat{y}_t \neq y_t} + \sum_{y_t = -1} c_- I_{\hat{y}_t \neq y_t} \right] \\
\leq \alpha \left[ \sum_{y_t = -1} \ell_{\xi_+} (y_t u^T x_t) + \sum_{y_t = 1} \ell_{\xi_-} (y_t u^T x_t) \right] + \frac{\alpha^2}{2} \|u\|^2
$$

From the result in Theorem 2, we can see the proposed asymmetric query strategy aims to minimize the cost-sensitive error. Next, we leverage the previous results to show that minimizing the cost-sensitive error with appropriate costs is equivalent to the F-measure maximization. To present the results, we first give some notations. Let $h(x) \in \mathcal{H} : \mathbb{R}^d \rightarrow \{1, -1\}$ denote a classifier and $e(h) = (e_1(h), e_2(h))^T$ denote the false negative (FN) error and false positive (FP) error of $h(x)$ on the population level, respectively, i.e.,

$$
e_1(h) = \Pr(y = 1, h(x) = -1)$$

$$e_2(h) = \Pr(y = -1, h(x) = 1)$$

where $\Pr(\cdot)$ denotes the probability over $(x, y)$. When it is clear from the context, we write $e = e(h)$ for short. Let $P_1$ denote the marginal probability of the positive class, i.e., $P_1 = \Pr(y = 1)$. Then the F-measure (i.e., F-1 score) of $h(\cdot)$ on the population level can be computed by [26]

$$F(h) \triangleq F(e) = \frac{2(P_1 - e_1)}{2P_1 - e_1 + e_2}$$

Let $c(t) = (1 - t, t)^T$. The following proposition exhibits that maximizing the F-measure is equivalent to minimizing a cost-sensitive error.

**Proposition 1.** (Proposition 4 [26]) Let $F_c = \max_{e} F(e)$. Then we have $e_* = \arg \min_{e} c(F_c)^T e \Leftrightarrow F(e_*) = F_c$.

The above proposition indicates that one can optimize the following cost-sensitive error

$$c(F_c)^T e = \left(1 - \frac{F_c}{2}\right) e_1 + \frac{F_c}{2} e_2$$
to obtain an optimal classifier $h^*(x)$, which will give the
optimal F-measure, i.e., $F(h^*) = F_\theta$. However, the cost-
sensitive error in (5) requires knowing the exact value of the
optimal F-measure. To address this issue, we discretize
(0, 1) to have a set of evenly distributed values $\left\{\theta_1, \ldots, \theta_K\right\}$
such that $\theta_{j+1} - \theta_j = \epsilon_0/2$. Then we can solve for a series of $K$
classifiers to minimize the cost-sensitive error
\[
h^*_j = \arg\min_{h \in H} \left(1 - \frac{\theta_j}{2}\right) e_1 + \frac{\theta_j}{2} e_2 = c(\theta_j)^T e, j = 1, \ldots, K
\]
(6)

The following proposition shows that there exists one classifier
among $\left\{h^*_1, \ldots, h^*_K\right\}$ that can achieve a close-to-optimal
F-measure as long as $\epsilon_0$ is small enough.

**Proposition 2.** Let $\left\{\theta_1, \ldots, \theta_K\right\}$ be a set of values evenly
distributed in $(0, 1)$ such that $\theta_{j+1} - \theta_j = \epsilon_0/2$. Then there
exists $h^*_j \in \left\{h^*_1, \ldots, h^*_K\right\}$ such that
\[
F(h^*_j) \geq F_\theta - \frac{2\epsilon_0 B}{P_1}
\]
where $B = \max_e \|e\|_2$.

**Remark:** The above proposition is a corollary of Proposition
5 in [26]. Note that the cost-sensitive error in (6) is just the
population level counterpart of the cost-sensitive error in
Theorem 2, which further justifies the proposed asymmetric
query model.

The above analysis implies that we can try different values
for the costs associated with the false negative error and false positive error, and use the cross-validation approach to
choose the best setting.

4. EXPERIMENT AND RESULTS

In order to investigate the algorithms on data of different
types, dimensionality, and proportion of positive examples, we
conduct the experiment on 5 binary classification tasks
from 3 datasets. All datasets are split as 2:1 for validation
and testing respectively, with more information listed in Table
1. The validation data is used to tune the parameters in the
compared algorithms. The testing data is used to evaluate the
performance of different algorithms. On each collection, we evaluate the performance mainly through $F_1$
score across the number of received examples. Also, we provide both “query evaluation” and “query and push evaluation”
according to the potential two types of label sources.

4.1 Data

One collection is cover type dataset from the UCI repository
of Machine Learning databases. It contains 581,012 examples and 7 classes of forest type, namely, Spruce-Fir, Lodgepole Pine, Ponderosa Pine, Cottonwood/Willow, Aspen, Douglas-fir, and Krummholz. Since the 7 classes have distinct rates of positive examples, we conduct the experiment on three of them with different level of imbalance to investigate the performance of algorithms. The three binary classification problems are Ponderosa Pine vs Non Ponderosa Pine, referred to as “cov1”, Spruce-Fir vs Non Spruce-Fir, referred to as “cov2”, and Lodgepole Pine vs None Lodgepole Pine, referred to as “cov3”. From cov1, cov2 to cov3, the level of imbalance decreases. Detailed information about the three tasks can be seen in Table 1.

<table>
<thead>
<tr>
<th># of examples</th>
<th>Percentage</th>
<th># of Positive features</th>
</tr>
</thead>
<tbody>
<tr>
<td>cov1 341</td>
<td>0.1 0.1 0</td>
<td>10 0.1</td>
</tr>
<tr>
<td>cov2 387,341</td>
<td>0.005 0.01 0.1 1 0.01</td>
<td></td>
</tr>
<tr>
<td>cov3 48.76</td>
<td>0.01 0.1 0.1 0.1 1</td>
<td></td>
</tr>
<tr>
<td>hd 155,630</td>
<td>0.1 0.1 0.1 0.1 0.1</td>
<td></td>
</tr>
<tr>
<td>2days 666,667</td>
<td>0.001 0.01 0.01 0.01 0.01</td>
<td></td>
</tr>
<tr>
<td>CSOAL 76 11 0 1 0 1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2: The best parameter values of different algorithms using PE

We also evaluate our algorithm on two more datasets from
different domains, OHSUMED - a dataset of biomedical
publications, and 2 days' tweets collected from Twitter.
OHSUMED [17] is a well-known dataset, collecting 348, 543
medical documents from MEDLINE from the year 1987 to
1991. Each document consists of all or some of following
fields: MEDLINE identifier, MeSH terms, title, publication
type, abstract, author, and source. Each document is associated
with one or more MeSH terms, the medical subject
heading assigned by human. Since the MeSH term is organized
in a tree structure, we pick a subtree rooted at the
MeSH term “Heart Disease” to be the positive class, and
leave all the terms not in the subtree as negative. Specifically, a document is a positive example if and only if it contains
at least one MeSH term in the “Heart Disease” subtree.
The task is referred to as hd.

The two days' tweets, referred as 2days in the following
discussion, is a collection related to life satisfaction of the
author of the tweets. It is collected by keywords such as “life”,
“my”, etc. And each tweet is manually labeled as satisfy,
dissatisfy, or irrelevant. We consider the binary classification
problem of “related to the topic of life satisfaction” (positive
event) or not relevant (negative event). Due to the nature of the
data (e.g., tweet is a short text). this is a difficult task.

4.2 Evaluation

As we mentioned in the introduction section, we mainly
evaluate the algorithms by $F_1$ score. Specifically, we plot
the accumulative $F_1$ score along with the increase of the
iteration. We also conduct two types of evaluation according
to two types of label source. A push evaluation (PE)
means that the labeling is independent from the use of the
application, while push and query evaluation (PQE) means
the labels are completely obtained from the user feedback.
Specifically, in PE, only the examples such that $y_i = 1$ are
counted as positive predication, and in PQE the queried ex-

Table 1: Statistics of five classification tasks

The above analysis implies that we can try different values
for the costs associated with the false negative error and false positive error, and use the cross-validation approach to
choose the best setting.
Figure 3: Comparison of different query models on covtype dataset with budget $B = 1, 250$.

4.3 Results

As mentioned above, each collection is split into a validation and a testing subset. To be fair, we tune the parameters on the validation set for all the algorithms. Due to limits of space, we only report the best parameter values using PE in Table 2. For the three tasks on the covtype data, we fix the budget to $B = 1250$ and mainly investigate how the performance changes as the ratio of the number of positive examples change. For hd and 2days, we try two different values for the budget, a lower budget and a higher budget.

We show the results on covtype data in Figure 3 that compares different query strategies and Figure 4 that compares $Q_A$ to CSOAL. We only report the results with a lower budget value $B = 1, 250$, and the results with a larger budget value $B = 5000$ are included in the supplementary material due to limits of space. The results on hd are shown in Figure 5 for two different values of budget. Figure 6 shows the results on 2days data for two different values of budget.

4.4 Discussion of the results

We first discuss the best parameter values of the proposed algorithm $Q_A$. It can be seen from Table 2 the value of $c_-$ is larger than $c_+$ on imbalanced data (cov1, cov2, hd, and 2days), and they are the same for the balanced data cov3. This is consistent with our theoretical findings.

The comparison between different query strategies on covtype data (Figure 3) clearly demonstrate the effectiveness of the proposed asymmetric query model. In particular, when the data is imbalanced (cov1, cov2), the asymmetric query model is better than other baseline query models. When the data is balanced, the asymmetric query model reduces to the symmetric query model. Therefore, the line of $Q_S$ overlaps with the line of $Q_A$ in Figure 3(e), (f) and the line of $Q_D$ in Figure 3(e) as well. Moreover in this case, the probabilistic query model does not have any advantage over deterministic query strategy. On 2days data, we have similar observations. On hd data, the comparison between different query strategies (Figure 5 (a)–(d)) shows that using a smaller query budget favors the asymmetric query model more than using a larger value of budget. From the results, we also observe that using PE favors $Q_A$ more than using PQE.

Next, we discuss the comparison to CSOAL. On cover type dataset, $Q_A$ performs better than CSOAL on all the classes and both evaluations. On the hd data, the CSOAL performs better when $B = 5, 000$ especially on PQE, however, it loses to $Q_A$ when the budget is low ($B = 100$). On 2days data, $Q_A$ performs consistently better than CSOAL.

Finally, it is worth mentioning that when only querying those examples predicted as positive the algorithm performs extremely bad as seen in Figure 2. We conjecture that the
reason is that many of the positive predictions are difficult, i.e. their features look like positive examples but they are actually negative. There can be many such examples as the dataset is imbalanced. And querying those examples will push the decision boundary to that side that includes fewer positive examples and consequentially harm performance. Therefore, using a probabilistic query model with a smaller value of $c_+$ will reduce querying for such examples, which is consistent with observation both in theory and practice.

5. CONCLUSIONS

In this paper, we have considered online classification with imbalanced data under a query budget. We compare and investigate different query strategies in an online classification algorithm. We also propose a novel asymmetric query model and provide a theoretical analysis of the weighted mistake bound. We conducted extensive experiments on five classification tasks from three real datasets. The experimental results demonstrate the usefulness of the proposed online classification algorithm with an asymmetric query model.

6. ACKNOWLEDGMENTS

This work is supported by the National Science Foundation under grant 1545995. T. Yang was also partially supported by NSF (1463988). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the National Science Foundation.

7. REFERENCES

Figure 5: Results on the hd data with different budget

APPENDIX

A. PROOF OF THEOREM 1

**Proof.** We denote by \( \hat{y}_t = \text{sign}(p_t) \) and introduce the Bernoulli random variable \( M_t = 1_{\hat{y}_t \neq y_t} \). Consider a round \( t \) where the algorithms queries a label and makes a mistake. Then \( Z_t = 1 \) and \( M_t = 1 \). Then we have for any \( u \in \mathbb{R}^d \)

\[
\xi - \ell_t(y_t u^\top x_t) = \xi - \max(0, \xi - y_t u^\top x_t) \leq y_t u^\top x_t = y_t(u - w_{t-1} + w_{t+1})^\top x_t = y_t w_{t-1}^\top x_t + \frac{1}{2} \|u - w_{t-1}\|^2 - \frac{1}{2} \|u - w_t\|^2 + \frac{1}{2} \|w_{t-1} - w_t\|^2
\]

where we use the fact \( w_t = w_{t-1} + y_t x_t \) for \( Z_t = 1 \) and \( M_t = 1 \). Since \( y_t \neq \hat{y}_t \), then \( y_t w_{t-1}^\top x_t \leq 0 \). Replacing \( u \) by \( \omega \) with \( \alpha > 0 \) and reorganize the inequality we have

\[
(\alpha \xi + |p_t|)M_t Z_t \leq \alpha \ell_t(y_t u^\top x_t) + \frac{1}{2} \|u - w_{t-1}\|^2 - \frac{1}{2} \|u - w_t\|^2 + \frac{1}{2} \|w_{t-1} - w_t\|^2
\]

We note that the above inequality holds for all \( t = 1, \ldots, T \). If \( M_t Z_t = 0 \), then \( w_t = w_{t-1} \), and the above inequality holds because \( \alpha \ell_t(y_t u^\top x_t) \geq 0 \). Summing the above inequality over \( t = 1, \ldots, T \), we have

\[
\sum_{t=1}^{T} (\alpha \xi + |p_t|)M_t Z_t \leq \alpha \sum_{t=1}^{T} \ell_t(y_t u^\top x_t) + \frac{1}{2} \|u - w_0\|^2 + \sum_{t=1}^{T} \|w_{t-1} - w_t\|^2 + \frac{1}{2} \|w_{T-1} - w_T\|^2 M_T Z_T
\]

Therefore

\[
\sum_{t=1}^{T} (\alpha \xi + |p_t|) \leq \alpha \sum_{t=1}^{T} \ell_t(y_t u^\top x_t) + \frac{1}{2} \|u\|^2 - \frac{R^2}{2} M_T Z_T
\]

Due to \( \alpha = \frac{c + R^2/2}{t} \), we then have

\[
\sum_{t=1}^{T} (c + |p_t|)M_t Z_t \leq \alpha \sum_{t=1}^{T} \ell_t(y_t u^\top x_t) + \frac{c^2}{2} \|u\|^2
\]

By taking expectation over randomness in \( Z_t \) and noting that \( E[Z_t] = \frac{c + |p_t|}{c + |p_t|} \), we have

\[
E \left[ \sum_{t=1}^{T} c M_t \right] \leq \alpha \sum_{t=1}^{T} \ell_t(y_t u^\top x_t) + \frac{c^2}{2} \|u\|^2
\]

\( \square \)