Online Adaptive Asymmetric Active Learning for Budgeted Imbalanced Data

Yifan Zhang[†*], Peilin Zhao[†*], Jiezhang Cao[†], Wenye Ma[‡], Junzhou Huang[‡], Qingyao Wu[†§], Mingkui Tan[†§]
[†]South China University of Technology; [‡]Tencent AI Lab
{sezyifan@mail,qyw@,mingkuitan@}scut.edu.cn,peilinzhao@hotmail.com,{wenyema,joehhuang}@tencent.com

ABSTRACT

This paper investigates Online Active Learning (OAL) for imbalanced unlabeled datastream, where only a budget of labels can be queried to optimize some cost-sensitive performance measure. OAL can solve many real-world problems, such as anomaly detection in healthcare, finance and network security. In these problems, there are two key challenges: the query budget is often limited; the ratio between two classes is highly imbalanced. To address these challenges, existing work of OAL adopts either asymmetric losses or queries (an isolated asymmetric strategy) to tackle the imbalance, and uses first-order methods to optimize the cost-sensitive measure. However, they may incur two deficiencies: (1) the poor ability in handling imbalanced data due to the isolated asymmetric strategy; (2) relative slow convergence rate due to the first-order optimization.

In this paper, we propose a novel Online Adaptive Asymmetric Active (OA3) learning algorithm, which is based on a new asymmetric strategy (merging both the asymmetric losses and queries strategies), and second-order optimization. We theoretically analyze its bounds, and also empirically evaluate it on four real-world online anomaly detection tasks. Promising results confirm the effectiveness and robustness of the proposed algorithm in various application domains.

ACM Reference Format:
Yifan Zhang[†*], Peilin Zhao[†*], Jiezhang Cao[†], Wenye Ma[‡], Junzhou Huang[‡], Qingyao Wu[†§], Mingkui Tan[†§]. 2018. Online Adaptive Asymmetric Active Learning for Budgeted Imbalanced Data. In *KDD 2018: 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, August 19–23, 2018, London, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3219819.3219948

1 INTRODUCTION

With rapid growth of data and fast development of computational resources, machine learning is able to address more and more practical problems, powering many aspects of modern society [2, 20, 26, 30, 32]. Nevertheless, many machine learning methods train a model off-line and require the availability of sufficient data before training, while the off-line data are required to follow the i.i.d. assumption [11, 19]. However, in practice, there are many applications where the data come in an online manner and the i.i.d. assumption may not hold. To address these limitations, online learning has emerged as a powerful learning tool [6, 9, 15, 29], which makes no assumptions about the distribution of data, and thus is data efficient and adaptable [11, 19].

Most existing online methods assume all the data are labeled, without considering the labeling cost as well as the budget control problem. However, in many applications, such as medical diagnosis [20] and malicious URL detection [32], the cost of manual annotation is often expensive. In these cases, it is important to find the samples, which deserve to be labeled, from data streams. To handle this task, online active learning (OAL) [5, 28] has emerged, and it seeks to achieve promising performance with only a small number of labels, which are actively queried on data streams. Many studies [5, 28, 30] have found that different query rules may result in very different learning performance, which implies that the active learning strategy is very important.

On the other hand, in order to save the operation cost, companies would expect to spend as few funds as possible for data annotation. In other words, we only have a limited budget for data annotation. Given this limited budget, we have to select the most informative samples to query labels so that they can help to train a promising model. For convenience, the overall structure of the OAL scheme that we will study is provided in Figure 1.

In addition, the class-imbalance problem seriously affects the algorithm performance in practical applications, such as cancer diagnosis [20], financial credit monitoring [26] and network fraud detection [32]. Existing OAL algorithms usually optimize the performance using *accuracy* or *mistake rate* as metrics, which, however, are not good at handling the imbalance issue [29], and thus limit the performance of corresponding methods. To overcome this limitation, researchers have suggested using more informative metrics, such as the weighted *sum* of *sensitivity* and *specificity*, and the weighted *misclassification cost* [29].

[*]Equal Contribution
[§]Corresponding author

KDD 2018, August 19–23, 2018, London, United Kingdom

ACM ISBN 978-1-4503-5552-0/18/08. . . $15.00
https://doi.org/10.1145/3219819.3219948

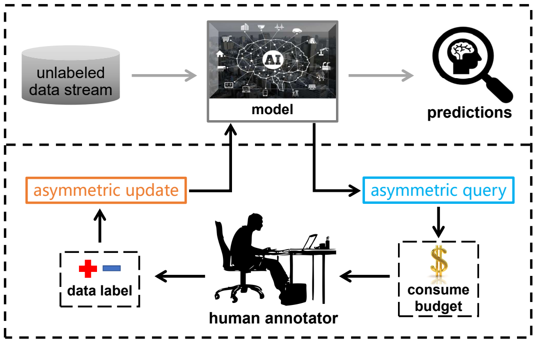


Figure 1: The scheme of Online Adaptive Asymmetric Active Learning under limited query budgets. The main task is to query a small subset of informative samples for labels, and use them to train a desirable model. Moreover, the query budget is given in advance.

Based on these metrics, a pioneering cost-sensitive online active learning algorithm (CSOAL) [32] was proposed to directly optimize asymmetrically cost-sensitive metrics for OAL. However, this method only adopts a symmetric query rule [5], which implies it ignores the imbalance problem in data selection. Thus, it is incapable of querying sufficient instances for the minority class (assumed as the positive class in this paper). Meanwhile, recent work [30] has discovered that using asymmetric query strategy helps to handle imbalanced data better, and thus proposed an online asymmetric active learning algorithm (OAAL) by assigning different query probabilities for the instances with different predictions. However, this method overlooks imbalance issues in optimization process, and tends to query more negative data due to the recommended parameter settings [30], which may lead to poor training on the positive data and thus poor performance in imbalance problems. In comparison, CSOAL is "asymmetric update plus symmetric query", and OAAL is "symmetric update plus asymmetric query". Both algorithms only consider the asymmetric strategy from one isolated perspective, which makes them difficult to achieve an optimal solution for imbalance problems.

Moreover, both algorithms have another limitation: they only consider first-order information of data streams, and may converge slowly when the scales of different dimensions vary significantly [6]. Thus, it would be difficult for them to achieve an optimal solution, especially when the labeled data are quite limited. Recent studies [10, 33, 34] have found second-order information (*i.e.,* the correlations between features) helps to enhance the performance of online algorithms significantly.

In this paper, we propose a novel online adaptive asymmetric active (**OA3**) learning algorithm by exploiting samples' second-order information. Based on the derived knowledge, we develop a new asymmetric strategy, considering both model optimization and label queries, which addresses imbalance issues better and improves model performance. We theoretically analyze the mistake bound and cost-sensitive metric bounds of the proposed algorithm, for the cases within budgets and over budgets, respectively. Note that the analytical thinking of exhausting budgets can be potentially used to improve the theoretical studies of other correlative algorithms. Moreover, we empirically validate the algorithmic practical value in real-world online anomaly detection. Encouraging results confirm the effectiveness and robustness for OA3 in various application domains. We also examine the influences of algorithm parameters. Extensive results validate the algorithm characteristics and recommend some candidate parameter settings for algorithm engineers.

The rest of this paper is organized as follows. We review the related work in Section 2 and present the algorithm in Section 3. Next, we theoretically analyze the algorithm in Section 4 and apply it to handle real-world online anomaly detection in Section 5. Finally, we conclude this paper.

2 RELATED WORK

Online learning has been a hot research topic in machine learning for many years. One pioneering method is Perceptron algorithm [15], which updates the predictive vector by adding the misclassified sample with a constant weight. Recently, many margin-based methods [9, 29] emerged, and show good performance. Despite their superiority, these methods only adopt samples' first-order information, which may result in slow convergence rate. To eliminate this limitation, many second-order methods [6, 10, 33, 34] were proposed, and significantly improve the performance with faster convergence.

Active learning aims to train a well-performed model by querying the labels of a small subset of informative data. It helps to reduce the labeling cost and attracts wide attention [3, 5, 12, 14, 28]. Particularly, one classic work [5] proposes a sampling method by drawing a Bernoulli random variable, which has been validated as effective and inspires many studies [30, 32]. Despite the effectiveness, most active algorithms [2, 7, 21, 23] assume that all the data are provided in advance, which is impractical in real-world problems. To address this limitation, OAL has emerged and raised wide research interest [13, 16, 25, 35].

Although OAL has been studied widely, only a few studies focus on class-imbalanced issues. One classic method is CSOAL [32], which adopts a cost-sensitive update rule to solve imbalance problems. Another famous method is OAAL [30], which uses an asymmetric query rule to handle imbalanced data. However, both pioneering methods only consider the asymmetric strategy from one isolated perspective, which restricts their abilities in imbalance problems. Moreover, both methods only consider first-order information of samples, which limits their performance. To address these limitations, we exploit samples' second-order information and develop a new asymmetric strategy, considering both optimization and label queries, to handle imbalance problems and accelerate convergence rate.

Finally, we highlight several differences of OA3 with several conceptually related methods, *i.e.* [16, 23]. In [23], a cost overlapped active learning algorithm is proposed to study the cost problem, but it is not an online method and does not consider class-imbalance issues. A second-order online active learning algorithm (SOAL) is proposed in [16]. This method adopts second-order information, but it ignores the budget limitations and class-imbalance problems. A cost-sensitive version of SOAL is presented in [16]. This method considers the asymmetric losses, but it lacks theoretical guarantees, and ignores budget limitations and asymmetric query strategies.

3 METHODS

In this section, we first present the scheme of the Online Adaptive Asymmetric Active (OA3) Learning algorithm. Then, relying on samples' second-order information, we propose a new asymmetric strategy, which consists of an asymmetric update rule and an asymmetric query rule.

3.1 Algorithm Scheme

Without loss of generality, we consider online binary classification under limited query budgets here. The main task is to learn a linear model[1]($i.e.,\ w \in \mathbb{R}^d$) on a sample stream $\{x_t|t=1,...,T\}$ under a limited budgets B, where T is the total sample size, and $x_t \in \mathbb{R}^d$ is a d-dimensional sample.

Formally, in the t-th round, the model receives a sample x_t, and then predicts its predictive margin $p_t{=}w_t^\top x_t$ and label $\hat{y}_t{=}\text{sign}(p_t)$, where w_t is the predictive vector learnt from previous $t{-}1$ samples. After that, the model needs to determine whether to query the true label $y_t \in \{-1,+1\}$. If it enjoys budget surpluses and decides to query, it will receive the true label y_t at the price of an unit budget, and then update itself according to the painful loss when making a mistake $\hat{y}_t \neq y_t$. Otherwise, the model will ignore x_t.

Inspired by the adaptive confidence weight technique [10], we exploit samples' second-order information. Assume that the model follows a multivariate Gaussian distribution [10], *i.e.,* $w{\sim}\mathcal{N}(\mu,\Sigma)$, where μ and Σ denote the mean and covariance of w, respectively. Without loss of generality, we can regard each mean value μ_i as the model's knowledge about feature i, and regard diagonal entries of covariance $\Sigma_{i,i}$ as the model's confidence about feature i. Generally, the smaller of $\Sigma_{i,i}$, the more confidence of model in mean weight μ_i. Then, given a definite Gaussian distribution, we predict the label by $\hat{y}_t{=}\text{sign}(\mu_t^\top x_t)$ [10, 33, 34].

We summarize the algorithm scheme in Algorithm 1.

Generally, we have two main challenges when designing this active algorithm.

- **How to update**: how to design an effective update strategy to achieve an optimal solution and faster convergence, which is described in Subsection 3.2.
- **When to query**: how to plan a valid query strategy to query the most informative samples in imbalanced settings, which is described in Subsection 3.3.

3.2 Adaptive Asymmetric Update Rule

In general, traditional online algorithms often optimize *accuracy* or *mistake rate*, which treats samples from different classes equally. These metrics, however, are impractical for imbalanced data, since models can easily obtain high performance by simply classifying all samples as the majority class.

Algorithm 1 Online Adaptive Asymmetric Active (OA3) Learning algorithm.

```
Require: budget $B$; learning rate $\eta$; regular parameter $\gamma$.
Initialize $\mu_1 = 0$, $\Sigma_1 = I$, $B_1 = 0$.
for $t = 1 \to T$ do
    Receive an example $x_t \in \mathbb{R}^d$;
    Compute $p_t = \mu_t^\top x_t$;
    Make the prediction $\hat{y}_t = \text{sign}(p_t)$;
    Draw a variable $Z_t = \boldsymbol{Query}(p_t) \in \{0,1\}$;
    if $Z_t = 1$ and $B_t < B$ then
        Query the true label $y_t \in \{-1,+1\}$;
        $B_{t+1} = B_t + 1$;
        $\mu_{t+1}, \Sigma_{t+1} = \boldsymbol{Update}(\mu_t, \Sigma_t; x_t, y_t)$.
    else
        $B_{t+1} = B_t, \mu_{t+1} = \mu_t, \Sigma_{t+1} = \Sigma_t$.
    end if
end for
```

To address this issue, one solution is to maximize the *sum* of weighted *sensitivity* and *specificity*:

$$sum = \alpha_p \times \frac{T_p - M_p}{T_p} + \alpha_n \times \frac{T_n - M_n}{T_n},$$

where T_p and T_n denote the total number of positive and negative samples, while M_p and M_n represent the number of false negatives and false positives, respectively. Besides, $\alpha_p, \alpha_n \in [0,1]$ denote the trade-off parameters between *sensitivity* and *specificity*, and $\alpha_p + \alpha_n = 1$. Note that when $\alpha_p = \alpha_n = 0.5$, the *sum* metric becomes the famous *balanced accuracy*.

In addition, another solution is to minimize the weighted *cost* of misclassification:

$$cost = c_p \times M_p + c_n \times M_n,$$

where $c_p, c_n{\in}[0,1]$ denote the cost weights for positive and negative instances, and $c_p{+}c_n{=}1$.

In general, either the higher *sum* value or the lower *cost* value, the better performance of the algorithm. Thus, we can adjust our focus to maximize *sum* or minimize *cost*, and both objectives are equivalent to minimizing [29]:

$$\sum_{y_t=+1} \rho \mathbb{I}_{(y_t \mu^\top x_t < 0)} + \sum_{y_t=-1} \mathbb{I}_{(y_t \mu^\top x_t < 0)},$$

where $\rho = \frac{\alpha_p T_n}{\alpha_n T_p}$ for maximizing *sum* metric, or $\rho = \frac{c_p}{c_n}$ for minimizing *cost* metric, while $\mathbb{I}_{(\cdot)}$ is an indicator function.

However, this objective is non-convex. To facilitate the optimization, we replace the indicator function by its convex surrogate, *i.e.,* the following modified hinge loss function:

$$\ell_t(\mu) = \big(\rho \mathbb{I}_{(y=+1)} {+} \mathbb{I}_{(y=-1)}\big) \max\big\{0, 1{-}y_t(\mu^\top x_t)\big\}. \quad (1)$$

Given a Gaussian distribution, we naturally recast the object function by optimizing the following unconstraint objective [10]:

$$D_{KL}\big(\mathcal{N}(\mu,\Sigma)||\mathcal{N}(\mu_t,\Sigma_t)\big) + \eta \ell_t(\mu) + \frac{1}{2\gamma} x_t^\top \Sigma x_t, \quad (2)$$

[1] We focus on the linear model for simplification, while one can generalize it to non-linear models in a Reproducing Kernel Hilbert Space.

where η is the learning rate, γ is the regularized parameter and D_{KL} denotes the Kullback-Leibler divergence, *i.e.,*

$$\begin{aligned}&D_{KL}\big(\mathcal{N}(\mu,\Sigma)||\mathcal{N}(\mu_t,\Sigma_t)\big)\\ &=\frac{1}{2}\log\Big(\frac{\det\Sigma_t}{\det\Sigma}\Big)+\frac{1}{2}\mathrm{Tr}(\Sigma_t^{-1}\Sigma)+\frac{1}{2}||\mu_t-\mu||^2_{\Sigma_t^{-1}}-\frac{d}{2}.\end{aligned}$$

Specifically, the objective (2) helps to reach the trade-off between distribution divergence (first term), loss function (second term) and model confidence (third term). In other words, the objective tends to make the least adjustment to minimize the painful loss and optimize the model confidence.

Nevertheless, this objective dose not have a closed-form solution. Thus, we replace the loss term $\ell_t(\mu)$ by its first order Taylor expansion $\ell_t(\mu_t)+g_t^\top(\mu-\mu_t)$, where $g_t=\partial\ell_t(\mu_t)$. Then, we obtain the final objective by removing constant terms:

$$f_t(\mu,\Sigma)=D_{KL}\big(\mathcal{N}(\mu,\Sigma)||\mathcal{N}(\mu_t,\Sigma_t)\big)+\eta g_t^\top\mu+\frac{1}{2\gamma}x_t^\top\Sigma x_t,$$

which is much easier to be solved.

We handle this optimization by decomposing the derived objective into two parts w.r.t. μ and Σ, respectively. Thus, we can perform the updates of μ and Σ independently:

- Update the predictive vector:
$$\mu_{t+1}=\arg\min_{\mu} f_t(\mu,\Sigma);$$
- If $\ell_t(\mu_t)\neq 0$, update the covariance matrix:
$$\Sigma_{t+1}=\arg\min_{\Sigma} f_t(\mu,\Sigma).$$

For the update of predictive vector, setting the derivative $\partial_\mu f_t(\mu_{t+1},\Sigma)$ to zero will give:

$$\Sigma_t^{-1}(\mu_{t+1}-\mu_t)+\eta g_t=0 \implies \mu_{t+1}=\mu_t-\eta\Sigma_t g_t.$$

For the update of covariance matrix, setting the derivative $\partial_\Sigma f_t(\mu,\Sigma_{t+1})$ to zero gives:

$$-\Sigma_{t+1}^{-1}+\Sigma_t^{-1}+\frac{x_t x_t^\top}{\gamma}=0 \implies \Sigma_{t+1}^{-1}=\Sigma_t^{-1}+\frac{x_t x_t^\top}{\gamma},$$

where we can easily prove the covariance matrix Σ_t is non-singular. Then, based on Woodbury identity [18], we have:

$$\Sigma_{t+1}=\Sigma_t-\frac{\Sigma_t x_t x_t^\top\Sigma_t}{\gamma+x_t^\top\Sigma_t x_t}.$$

Note that the update of the predictive vector μ relies on the confidence parameter Σ, so we update μ_t based on the updated covariance Σ_{t+1} instead of the previous one Σ_t, which will be more accurate and aggressive [33, 34]:

$$\mu_{t+1}=\mu_t-\eta\Sigma_{t+1}g_t.$$

We summarize the adaptive asymmetric update strategy in Algorithm 2.

Time Complexity Analysis. The time complexities of the updates of μ and Σ are both $\mathcal{O}(Td^2)$, so the overall time complexity of this update strategy is $\mathcal{O}(Td^2)$, where d is the dimensionality of the data. Nevertheless, the update efficiency of OA3 is slower than first-order algorithms $\big(\mathcal{O}(Td)\big)$, especially when handling high-dimensional datasets. To promote the efficiency, we propose the **diagonal version** of the update strategy, which accelerates the efficiency to $\mathcal{O}(Td)$. Specifically, in this version, only the diagonal entries of Σ were maintained and updated in each round. To save space, we do not describe the diagonal version in the pseudo code.

Algorithm 2 Adaptive Asymmetric Update Strategy: ***Update***$(\mu_t,\Sigma_t;x_t,y_t)$.

```
Require: $\rho=\frac{\alpha_p T_n}{\alpha_n T_p}$ for "sum" or $\rho=\frac{c_p}{c_n}$ for "cost";
Receive a sample $(x_t,y_t)$;
Compute the loss $\ell_t(\mu_t)$, based on Equation (1);
if $\ell_t(\mu_t)>0$ then
    $\Sigma_{t+1}=\Sigma_t-\frac{\Sigma_t x_t x_t^\top\Sigma_t}{\gamma+x_t^\top\Sigma_t x_t}$;
    $\mu_{t+1}=\mu_t-\eta\Sigma_{t+1}g_t$, where $g_t=\partial\ell_t(\mu_t)$.
else
    $\mu_{t+1}=\mu_t, \Sigma_{t+1}=\Sigma_t$.
end if
Output $\mu_{t+1},\Sigma_{t+1}$.
```

Remark. *We employ the adaptive asymmetric update rule for OA3 to pursue high performance with faster convergence. Nevertheless, it is not the only choice, where many other classic techniques can be used, such as online gradient descent [29] and online margin-based strategies [6, 9].*

3.3 Asymmetric Query Strategy

In a pioneering study [5], a classic sampling method was proposed to query labels, based on a Bernoulli random variable Z_t. Particularly, the sampling probability depends on the absolute value of the predictive margin $|p_t|$. Specifically, the formula is:

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

where $\delta>0$ is the query bias. In specific, the lower absolute value of the predictive margin $|p_t|$, the higher probability to query the label of the corresponding sample. That is, when $|p_t|$ is small, the sample x_t would be close to the predictive boundary, and thus be difficult to be classified. So, querying its label will be more valuable.

However, this symmetric query rule ignores the imbalance issue and treats all predictions for the two imbalanced classes equally, which limits its potential to solve imbalance problems. To address this limitation, inspired by recent work [30], we employ an asymmetric strategy to query labels:

$$Pr\big(Z_t=1\big)=\begin{cases}\frac{\delta_+}{\delta_++|p_t|}, & \text{if } p_t\geq 0;\\ \frac{\delta_-}{\delta_-+|p_t|}, & \text{if } p_t<0;\end{cases}$$

where $\delta_+>0$ and $\delta_->0$ denote the query biases for positive and negative predictions, respectively.

Directly using this asymmetric strategy makes the judgement heavily depend on the margin p_t, which relies on the precision of the model μ_t. However, when the model is not precise, query decisions of the model will be uninformative.

To address this issue, we employ samples' second-order information to enhance the robustness of the query judgements. We first define the variance of a model on the sample x_t as $v_t=x_t^\top\Sigma_t x_t$, which represents the familiarity of the model with the current sample through previous experience. Based on v_t, we then define the query confidence:

$$c_t = -\frac{1}{2}\frac{\eta\rho_{max}}{\frac{1}{v_t}+\frac{1}{\gamma}}, \tag{3}$$

where ρ_{max}= $\max\{1,\rho\}$. We highlight that this equation is helpful for the theoretical analysis. Moreover, the confidence c_t directly depends on the variance v_t. From this equation, we can observe that when the model has been well trained on some instances similar to the current sample x_t (*i.e.,* low variance v_t), the model would be confident of this sample (*i.e.,* large confidence c_t).

By combining the predictive margin and the confidence, we obtain the final query parameter:

$$q_t = |p_t| + c_t. \tag{4}$$

Moreover, the learning rate η and regularized parameter γ in Equation (3) can be understood as the trade-off factors in Equation (4).

Based on above analyses, we propose an improved asymmetric query strategy:

$$Pr\big(Z_t = 1\big) = \begin{cases} \frac{\delta_+}{\delta_+ + q_t}, & \text{if } p_t \geq 0; \\ \frac{\delta_-}{\delta_- + q_t}, & \text{if } p_t < 0. \end{cases}$$

In particular, when q_t>0, the query decisions of the model is very confident, so we directly draw a Bernoulli variable based on this equation. Besides, when q_t≤0, the query decisions is unconfident of the current sample, so we decide to query the true label whatever the value of p_t, *i.e.,* obtaining Z_t=1 by setting q_t=0 (see the above equation).

We summarize the proposed asymmetric query strategy in Algorithm 3.

Algorithm 3 Asymmetric Query Strategy: ***Query***(p_t).

```
Require: $\rho_{max} = \max\{1,\rho\}$; query bias ($\delta_+$,$\delta_-$) for positive
    and negative predictions.
Compute the variance $v_t = x_t^\top \Sigma_t x_t$;
Compute the query parameter $q_t = |p_t| - \frac{1}{2}\frac{\eta\rho_{max}}{\frac{1}{v_t}+\frac{1}{\gamma}}$;
if $q_t \leq 0$ then
    Set $q_t = 0$;
end if
if $p_t \geq 0$ then
    $p_t^+ = \frac{\delta_+}{\delta_+ + q_t}$;
    Draw a Bernoulli variable $Z_t \in \{0,1\}$ with $p_t^+$.
else
    $p_t^- = \frac{\delta_-}{\delta_- + q_t}$;
    Draw a Bernoulli variable $Z_t \in \{0,1\}$ with $p_t^-$.
end if
Output $Z_t$.
```

We can obtain the expected number of queried samples without budget limitations as follows.

Proposition 1. *Based on the proposed asymmetric query strategy, the expected number of requested samples without a budget is:*

$$\sum \mathbb{I}_{(q_t \leq 0)} + \sum_{\substack{q_t > 0 \\ p_t \geq 0}} \frac{\delta_+}{\delta_+ + q_t} + \sum_{\substack{q_t > 0 \\ p_t < 0}} \frac{\delta_-}{\delta_- + q_t}.$$

4 THEORETICAL ANALYSIS

In this section, we analyze the proposed algorithm in terms of its mistake bound and two cost-sensitive metric bounds, for the cases within budgets and over budgets, respectively. Before that, we first show a lemma, which facilitates the analysis within budgets. Due to page limitation, all the proofs are put into Appendix A[2].

For convenience, we introduce the following notations:

$$M_t = \mathbb{I}_{(\hat{y}_t \neq y_t)},\ \rho = \frac{\alpha_p T_n}{\alpha_n T_p} \text{ or } \frac{c_p}{c_n},$$

ρ_t=$\rho\mathbb{I}_{(y_t=+1)}$+$\mathbb{I}_{(y_t=-1)}$, ρ_{max}= $\max\{1,\rho\}$, ρ_{min}= $\min\{1,\rho\}$.

Lemma 1. *Let $(x_1, y_1), ..., (x_T, y_T)$ be a sequence of input samples, where $x_t \in \mathbb{R}^d$ and $y_t \in \{-1,+1\}$ for all t. Let T_B be the round that runs out of the budgets, i.e., $B_{T_B+1} = B$. For any $\mu \in \mathbb{R}^d$ and any $\delta > 0$, OA3 algorithm satisfies:*

$$\sum_{t=1}^{T_B} M_t Z_t(\delta + q_t) \leq \frac{\delta}{\rho_{min}} \sum_{t=1}^{T_B} \ell_t(\mu) + \frac{1}{\eta\rho_{min}} \mathrm{Tr}(|\Sigma_{T_B+1}^{-1}|)\times \big[M(\mu) + (1-\delta)^2||\mu||^2\big],$$

where $M(\mu) = \max_t ||\mu_t - \mu||^2$.

Based on Lemma 1, we obtain the following three theorems for the case **within budgets**.

Theorem 1. *Let $(x_1, y_1), ..., (x_T, y_T)$ be a sequence of input samples, where $x_t \in \mathbb{R}^d$ and $y_t \in \{-1,+1\}$ for all t. Let T_B be the round that runs out of the budgets, i.e., $B_{T_B+1} = B$. For any $\mu \in \mathbb{R}^d$, the expected mistake number of OA3 within budgets is bounded by:*

$$\mathbb{E}\Big[\sum_{t=1}^{T_B} M_t\Big] = \mathbb{E}\Big[\sum_{\substack{t=1 \\ y_t=+1}}^{T_B} M_t + \sum_{\substack{t=1 \\ y_t=-1}}^{T_B} M_t\Big]$$
$$\leq \frac{1}{\rho_{min}}\Big[\sum_{t=1}^{T_B} \ell_t(\mu) + \frac{1}{\eta} D(\mu) \mathrm{Tr}(|\Sigma_{T_B+1}^{-1}|)\Big],$$

where $D(\mu)$= $\max\Big\{\frac{M(\mu)+(1-\delta_+)^2||\mu||^2}{\delta_+}, \frac{M(\mu)+(1-\delta_-)^2||\mu||^2}{\delta_-}\Big\}$.

This mistake bound helps to analyze the weighted *sum* performance under limited budgets.

Theorem 2. *Under the same condition in Theorem 1, by setting $\rho = \frac{\alpha_p T_n}{\alpha_n T_p}$, the proposed OA3 within budgets satisfies for any $\mu \in \mathbb{R}^d$:*

$$\mathbb{E}\Big[sum\Big] \geq 1 - \frac{\alpha_n \rho_{max}}{T_n \rho_{min}}\Big[\sum_{t=1}^{T_B} \ell_t(\mu) + \frac{1}{\eta} D(\mu) \mathrm{Tr}(|\Sigma_{T_B+1}^{-1}|)\Big].$$

[2] The **Supplementary Materials** with **all Appendices** can be obtained in https://github.com/Vanint/OA3.

Remark. *By setting $\alpha_p = \alpha_n = 0.5$, we can easily obtain the bound of the balanced accuracy.*

Note that α_n cannot be set to zero, because $\rho = \frac{\alpha_p T_n}{\alpha_n T_p}$. One restriction is that we could not acquire $\frac{T_n}{T_p}$ in advance in real-world tasks. To overcome this limitation, we can choose *cost* metric as an alternative, where $\rho = \frac{c_p}{c_n}$. Then, engineers need not worry $\frac{T_n}{T_p}$ any more. Next, we bound the cumulative *cost* performance under limited budgets.

Theorem 3. *Under the same condition in Theorem 1, by setting $\rho{=}\frac{c_p}{c_n}$, the proposed OA3 within budgets satisfies for any $\mu \in \mathbb{R}^d$:*

$$\mathbb{E}\Big[cost\Big] \le \frac{c_n \rho_{max}}{\rho_{min}} \Big[\sum_{t=1}^{T_B} \ell_t(\mu) + \frac{1}{\eta} D(\mu) \mathrm{Tr}(|\Sigma_{T_B+1}^{-1}|)\Big].$$

Note that c_n cannot be set to zero, since $\rho{=}\frac{c_p}{c_n}$.

By now, we have analyzed OA3 algorithm within budgets. Next, we analyze OA3 for the case **over budgets**.

Theorem 4. *Let $(x_1, y_1), ..., (x_T, y_T)$ be a sample stream, where $x_t \in \mathbb{R}^d$ and $y_t \in \{-1, +1\}$. Let T_B be the round that uses up the budgets, i.e., $B_{T_B+1}{=}B$. For any $\mu{\in}\mathbb{R}^d$, the expected mistakes of OA3 over budgets is bounded by:*

$$\mathbb{E}\Big[\sum_{T_B+1}^{T} M_t\Big] \le \sum_{T_B+1}^{T} \Big[\frac{\ell_t(\mu)}{\rho_{min}} + y_t x_t^\top \mu_{T_B+1}\Big],$$

where μ_{T_B+1} is the predictive vector of model, trained by all the previous queried samples.

Now, we bound the weighted *sum* and misclassification *cost* after running out of budgets.

Theorem 5. *Under the same condition in Theorem 4, by setting $\rho = \frac{\alpha_p T_n}{\alpha_n T_p}$, the sum performance of OA3 over budgets satisfies for any $\mu \in \mathbb{R}^d$:*

$$\mathbb{E}\Big[sum\Big] \ge 1 - \frac{\alpha_n \rho_{max}}{T_n} \sum_{T_B+1}^{T} \Big[\frac{\ell_t(\mu)}{\rho_{min}} + y_t x_t^\top \mu_{T_B+1}\Big].$$

Theorem 6. *Under the same condition in Theorem 4, by setting $\rho{=}\frac{c_p}{c_n}$, the misclassification cost of OA3 over budgets satisfies for any $\mu \in \mathbb{R}^d$:*

$$\mathbb{E}\Big[cost\Big] \le c_n \rho_{max} \sum_{T_B+1}^{T} \Big[\frac{\ell_t(\mu)}{\rho_{min}} + y_t x_t^\top \mu_{T_B+1}\Big].$$

Remark. *This analytical thinking of exhausting budgets helps to optimize the theoretical studies of budgeted OAL, and can be potentially used to improve CSOAL [32], OAAL [30] and SOAL [16].*

5 EXPERIMENTAL RESULTS

The proposed OA3 algorithm can be potentially applied to a wide range of real-world tasks in data mining. To evaluate the algorithm performance and parameter sensitivity, we apply OA3 to handle four **online anomaly detection** tasks.[3]

5.1 Application Domains and Datasets

We first exhibit the related application domains we will study:

- **Finance**: In finance area, the credit risk detection problem is very critical. Here, the task is to discriminate the customers with bad credit risks from normal ones on the "german" dataset.
- **Bioinformatics**: Anomaly detection in bioinformatics with "Cod-RNA" dataset is to detect the anomaly non-coding RNAs from some large sequenced genomes.
- **Web Security**: We apply our algorithm to classify the abnormal web page on "w8a" dataset, where the data are highly sparse.
- **Medical Imaging**: We address medical image anomaly detection with "KDDCUP08" cancer dataset[4]. The main goal is to detect the breast cancer from X-ray images.

The statistics of all datasets are summarized in Table 1. The first three datasets are obtained from LIBSVM [8][5].

Table 1: Datasets for Online Anomaly Detection

Dataset	#Examples	#Features	#Pos:#Neg
german	1000	24	1:2.3
Cod-RNA	271617	8	1:2.0
w8a	64700	300	1:32.5
KDDCUP08	102294	117	1:163.19

5.2 Experimental Testbed and Setup

For adequate evaluations, we compare **OA3** and the diagonal variant (**OA3**$_{diag}$) with several state-of-the-art online active methods: (1) Online Passive-aggressive Active Algorithm (**PAA**) [25]; (2) Online Asymmetric Active Algorithm (**OAAL**) [30]; (3) Cost-Sensitive Online Active Algorithm (**CSOAL**) [32]; (4) Second-order Online Active Algorithm (**SOAL**) [16] and its cost-sensitive variant (**SOAL-CS**) [16].

Besides, for examining our asymmetric query strategy, we also compare OA3 and OA3$_{diag}$ with two variants of OA3 with weak query rules: (1) OA3 with the "First come first served" strategy (**OA3-F**), which is the pure updating version without query strategies; (2) OA3 with the random query strategy (**OA3-R**).

Since each data point arrives sequentially in online manner, each sample is normalized by $x_t \leftarrow \frac{x_t}{\|x_t\|_2}$. When budgets have run out, the update of corresponding method will stop.

For fair comparisons, we use the same settings for all algorithms. We set $\alpha_p{=}\alpha_n{=}0.5$ for *sum*, and $c_p{=}0.9$ and $c_n{=}0.1$ for *cost*. The value of ρ is set to $\frac{\alpha_p T_n}{\alpha_n T_p}$ for *sum*, or $\frac{c_p}{c_n}$ for *cost*. In addition, query biases (δ, δ_+, δ_-) and learning rates η for all algorithms are selected from $[10^{-5}, 10^{-4}, ..., 10^{4}, 10^{5}]$ using cross validations. By default, the regularization parameter γ is set to 1 for all second-order algorithms (*i.e.,* SOAL and OA3 based algorithms).

For stable evaluations, we run experiments over 20 random permutations for each dataset. Results are averaged over these 20 runs and 4 metrics are employed to evaluate the

[3] The **codes** will be released in https://github.com/Vanint/OA3.

[4] http://www.kdd.org/kdd-cup/view/kdd-cup-2008/Data.

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

performance: *sensitivity*, *specificity*, weighted *sum* of sensitivity and specificity, and weighted *cost* of misclassification.

Last, all algorithms were implemented in MATLAB on a 3.40GHz Windows machine.

5.3 Evaluation on Fixed Query Budgets

In this subsection, we evaluate all algorithms under fixed budgets. Figures 2 and 3 record the results in terms of *sum* and *cost*, respectively. Table 2 shows more detailed results. By evaluating the performance in terms of cost-sensitive metrics, we draw several observations below.

First, OAAL (with asymmetric query) and CSOAL (with asymmetric update) outperform PAA (with symmetric update and query) in most cases. This comparison demonstrates the importance of taking imbalance issues into considerations when conducting online active learning for imbalanced data.

Second, all second-order algorithms (namely, SOAL and OA3 based algorithms) outperform all first-order algorithms (*i.e.*, PAA, OAAL and CSOAL) in most cases. This demonstrates the effectiveness of samples' second-order information.

Next, our algorithms outperform all baselines by at least 1.2% *sum* and 8.8% relative *cost*, respectively. This implies that our proposed strategy surpasses both the isolated asymmetric strategies. This also demonstrates the superiority of our query strategy, compared to the "First come first served" strategy and the random query strategy.

Moreover, OA3 and OA3$_{diag}$ show high performance with smaller standard deviations than other algorithms, which implies that the proposed algorithms are more stable and robust on the task of online anomaly detection. More comparisons on the robustness are put in Appendix B due to the page limitation. The additional comparison in the appendix also confirms this claim.

From comparisons in terms of *sensitivity* and *specificity*, our proposed algorithms achieve the best *sensitivity* on all datasets and produce fairly good *specificity* on most datasets. This shows our algorithms pay more attention to minority samples, which are usually more important in practical tasks.

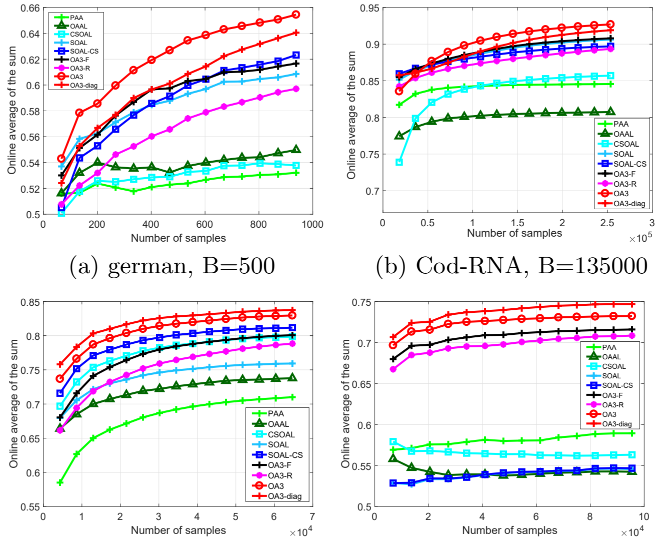
Last, we compare the efficiency of proposed methods. From Table 2, OA3$_{diag}$ is much faster than OA3 with quite small performance decrease. It implies that OA3$_{diag}$ is a good choice to balance the performance and efficiency in real-world online anomaly detections, especially on high-dimensional problems.

5.4 Evaluation on Varying Query Budgets

We compare the performance of all algorithms with varying query budgets. Due to the page limitation, we only present results in terms of *sum* in Figure 4. The results in terms of *cost* are put in Appendix B.

From Figure 4 and results in the appendix, our algorithms achieve good performance over a wide range of budgets in terms of both metrics, which demonstrates the effectiveness and robustness of our algorithms. Furthermore, it suggests that our algorithms can help real-world companies with different labeling budgets to handle real-world tasks.

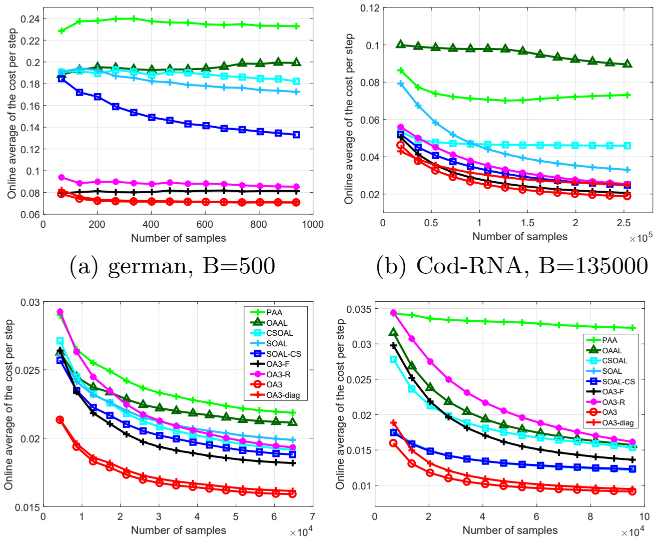
In addition, when the query budget falls, the performance differences between algorithms decrease and the standard deviations of each algorithm increase. This implies that the randomness of samples plays an important role in algorithm performance, especially when the budget is limited, which validates the importance and necessity of the data selection.



(a) german, B=500 (b) Cod-RNA, B=135000

(c) w8a, B=32000 (d) KDDCUP08, B=50000

Figure 2: Evaluation of sum with fixed budget.



(a) german, B=500 (b) Cod-RNA, B=135000

(c) w8a, B=32000 (d) KDDCUP08, B=50000

Figure 3: Evaluation of cost with fixed budget.

5.5 Evaluation Between Query Biases

In this subsection, we examine the influences of query biases on the performance of OA3 under limited budgets, where all query biases (δ_+, δ_-) are selected from $[10^{-5}, 10^{-4}, ..., 10^4, 10^5]$ while other parameters are fixed. Due to the page limitation, we only exhibit the results in terms of *sum* in Figure 5, and put the results in terms of *cost* in Appendix B.

Table 2: Evaluation of the online anomaly detection performance

Algorithm	“*sum*” on german (B=500)				“*cost*” on german (B=500)	
	Sum(%)	Sensitivity(%)	Specificity (%)	Time(s)	Cost	Time(s)
PAA	53.230 ± 2.153	15.667 ± 10.975	**90.793 ± 7.817**	0.003	233.405 ± 25.178	0.003
OAAL	55.005 ± 2.153	44.617 ± 19.391	65.393 ± 18.514	0.003	200.180 ± 32.281	0.003
CSOAL	53.832 ± 1.974	28.900 ± 10.314	78.764 ± 9.187	0.003	179.900 ± 13.007	0.003
SOAL	61.137 ± 1.952	38.667 ± 6.976	83.607 ± 3.933	0.009	171.900 ± 16.762	0.008
SOAL-CS	62.692 ± 2.093	46.483 ± 7.702	78.900 ± 4.310	0.010	131.890 ± 10.572	0.010
OA3-F	61.923 ± 3.071	61.567 ± 8.055	62.279 ± 7.706	0.007	80.875 ± 7.623	0.009
OA3-R	60.199 ± 1.815	60.433 ± 3.207	59.964 ± 2.943	0.008	84.495 ± 5.211	0.008
OA3	**65.774** ± 1.513	**66.983 ± 4.513**	64.564 ± 4.700	0.010	**70.520 ± 0.724**	0.010
OA3$_{diag}$	**64.490 ± 1.153**	64.867 ± 5.441	64.114 ± 5.741	0.008	**70.710 ± 1.036**	0.008

Algorithm	“*sum*” on Cod-RNA (B=135000)				“*cost*” on Cod-RNA (B=135000)	
	Sum(%)	Sensitivity(%)	Specificity (%)	Time(s)	Cost	Time(s)
PAA	84.571 ± 1.709	78.495 ± 5.337	90.647 ± 3.417	0.671	19912.565 ± 6200.220	0.677
OAAL	80.751 ± 0.103	73.419 ± 0.194	88.083 ± 0.053	0.665	24133.825 ± 6839.576	0.674
CSOAL	85.773 ± 0.184	84.240 ± 0.753	87.306 ± 1.076	0.653	12457.405 ± 489.362	0.666
SOAL	90.729 ± 0.190	87.758 ± 0.634	**93.699 ± 0.348**	1.258	8814.795 ± 948.552	1.160
SOAL-CS	89.776 ± 0.101	90.340 ± 0.403	89.211 ± 0.475	1.265	6627.710 ± 168.682	1.302
OA3-F	90.887 ± 0.155	92.382 ± 0.367	89.392 ± 0.460	1.007	5504.965 ± 413.125	0.873
OA3-R	89.535 ± **0.033**	90.735 ± 0.075	88.334 ± 0.082	1.026	6718.880 ± **50.371**	0.914
OA3	**92.813** ± 0.114	**93.814 ± 0.751**	91.812 ± 0.801	1.245	**5023.110 ± 80.403**	1.371
OA3$_{diag}$	**92.065** ± 0.329	92.319 ± 1.169	91.812 ± 0.907	1.168	6723.005 ± 2205.813	1.168

Algorithm	“*sum*” on w8a (B=37000)				“*cost*” on w8a (B=37000)	
	Sum(%)	Sensitivity(%)	Specificity (%)	Time(s)	Cost	Time(s)
PAA	71.003 ± 0.731	50.936 ± 1.501	**91.070 ± 0.053**	0.713	1414.960 ± 14.954	0.705
OAAL	73.770 ± 0.813	57.584 ± 1.783	89.957 ± 0.256	0.675	1367.570 ± 21.170	0.663
CSOAL	79.853 ± 0.549	70.083 ± 1.219	89.622 ± 0.180	0.698	1239.370 ± 10.884	0.716
SOAL	75.926 ± 0.516	61.247 ± 1.035	90.606 ± 0.047	4.615	1286.010 ± 12.753	5.620
SOAL-CS	81.171 ± **0.269**	72.447 ± 0.552	89.896 ± 0.050	6.052	1216.880 ± 12.102	5.920
OA3-F	80.066 ± 1.452	**84.721 ± 0.233**	75.411 ± 2.985	18.751	1176.575 ± 27.748	6.393
OA3-R	78.832 ± 1.275	83.539 ± 0.493	74.125 ± 2.402	17.742	1250.995 ± 15.045	6.449
OA3	**82.948** ± 0.829	**84.553 ± 0.424**	81.344 ± 1.857	27.915	**1029.405 ± 5.691**	15.781
OA3$_{diag}$	**83.736 ± 0.339**	82.957 ± 0.290	84.515 ± 0.678	5.658	**1044.470 ± 6.712**	4.318

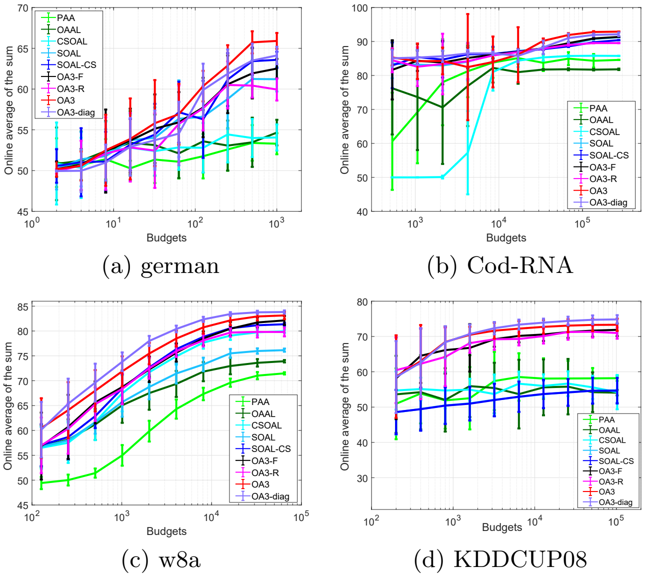
Algorithm	“*sum*” on KDDCUP08 (B=50000)				“*cost*” on KDDCUP08 (B=50000)	
	Sum(%)	Sensitivity(%)	Specificity (%)	Time(s)	Cost	Time(s)
PAA	58.937 ± 2.936	47.175 ± 6.195	70.698 ± 1.116	0.510	3295.890 ± 221.181	0.510
OAAL	54.274 ± 5.538	46.091 ± 10.465	62.457 ± 3.662	0.508	1585.150 ± 146.450	0.514
CSOAL	56.383 ± 1.566	24.197 ± 3.765	**88.569 ± 0.935**	0.507	1546.870 ± 49.744	0.459
SOAL	54.602 ± 3.593	35.682 ± 6.906	73.522 ± 1.899	11.843	1240.105 ± 144.159	5.534
SOAL-CS	54.689 ± 3.760	35.827 ± 7.231	73.552 ± 1.784	11.553	1252.795 ± 159.430	5.475
OA3-F	71.662 ± 1.740	89.599 ± 2.267	53.725 ± 3.681	5.456	1376.830 ± 45.394	5.162
OA3-R	70.950 ± 1.600	88.884 ± 1.785	53.016 ± 2.810	5.565	1617.190 ± 47.160	5.195
OA3	**73.270** ± 1.755	90.602 ± 2.308	55.938 ± 3.589	10.330	**930.405** ± 82.508	2.878
OA3$_{diag}$	**74.733 ± 1.242**	**91.677 ± 1.325**	57.789 ± 2.006	4.024	**962.765 ± 14.522**	1.735

The best results (*i.e.*, deep red color in Figure 5) are often obtained when $\delta_+ \in \{10^2, 10^3, 10^4, 10^5\}$ and $\delta_- \in \{1, 10\}$. This suggests, when querying more samples with the positive prediction (*i.e.*, $\delta_+ \geq \delta_-$), OA3 achieves the optimal result. The main reason is when paying more attention to positive predictions, the model will query more positive samples that are more informative in imbalanced tasks.

This observation is different from the discussions in OAAL [30]. In [30], the authors argued that δ_- should be larger than δ_+. The main difference is that our algorithm considers asymmetric strategies in both the optimization and query; While OAAL considers only the asymmetric strategy in the query. Thus, our method tends to query more positive samples due to the algorithm characteristics. Moreover, this observation also provides a candidate setting of query biases.

In addition, when both δ_+ and δ_- are large (*i.e.*, the upper right corners in Figure 5), our algorithms achieve fairly good performance. Specifically, in this setting, the algorithm tends to query every observed sample, and it degrade to the “First come first served” strategy. This demonstrates our algorithms with weak query strategy can also obtain fairly good performance. Moreover, when both δ_+ and δ_- are small (*i.e.*, the bottom left corner), the model tends to ignore the samples, so the algorithm performance will decrease significantly.

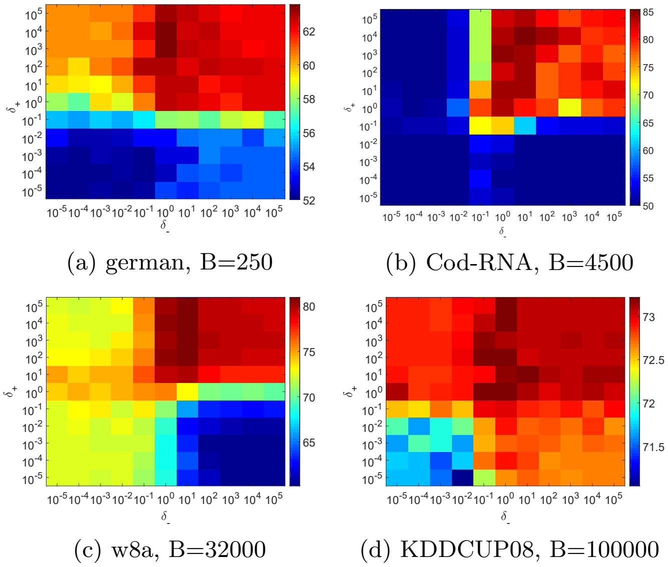
Furthermore, our algorithm with large δ_+ and small δ_- (*i.e.*, the upper left corners) achieves better performance than it with small δ_+ and large δ_- (*i.e.*, the bottom right corners), which validates the effectiveness of querying more positive predictions, *i.e.*, more minority samples.



(a) german (b) Cod-RNA

(c) w8a (d) KDDCUP08

Figure 4: Evaluation of sum with varying budgets.



(a) german, B=250 (b) Cod-RNA, B=4500

(c) w8a, B=32000 (d) KDDCUP08, B=100000

Figure 5: Evaluation of query biases.

5.6 Further Observations

We also evaluate the sensitivity w.r.t. other parameters in Appendix B. Here are some observations.

- **Learning rate** (η): when selecting learning rate from $[10^{-1}, 1, 10^{1}]$, OA3 achieves the best result on most datasets, and thus our algorithm performs stably on a wide range of learning rate. This experiment also suggests a candidate choice of the learning rate.
- **Regularized parameter** (γ): in general, the optimal γ changes for different datasets. Nevertheless, in most cases, the default setting $\gamma = 1$ makes the algorithm achieve the best or relatively good performance.
- **Cost weights** $((\alpha_p, \alpha_n), (c_p, c_n))$: our algorithms consistently outperform other algorithms under different settings of cost weights. This further demonstrates the effectiveness of the proposed algorithms.

6 CONCLUSIONS

In this paper, we have proposed a novel online adaptive asymmetric active learning algorithm to handle imbalanced datastream under limited query budgets. Relying on samples' second-order information, we develop a new asymmetric s-trategy, which merges both the asymmetric losses and queries strategies. We theoretically analyze the mistake and cost-sensitive metric bounds of the proposed algorithm, for the cases within budgets and over budgets. We also empirically evaluate the proposed algorithm in real-world anomaly detection tasks. Promising results not only confirm the effectiveness and robustness of the proposed algorithm, but also validate the favorable characteristics of parameter sensitivity.

7 ACKNOWLEDGEMENTS

This work was supported by National Natural Science Foundation of China (NSFC) 61502177 and 61602185, and Recruitment Program for Young Professionals, and Guangdong Provincial Scientific and Technological funds 2017B090901008, 2017A010101011, 2017B090910005, and Fundamental Research Funds for the Central Universities D2172500, D2172480, and Pearl River S&T Nova Program of Guangzhou 201806010-081 and CCF-Tencent Open Research Fund RAGR20170105.

REFERENCES

[1] P. Bachman, A. Sordoni, A. Trischler. Learning algorithms for active learning. *In 34th International Conference on Machine Learning*, 2017, pp. 301–310.

[2] N. Abe, B. Zadrozny, J. Langford. Outlier detection by active learning. *In SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2006, pp. 504-509.

[3] C. Aggarwal, X. Kong, Q. Gu, J. Han, P. Yu. Active learning: a survey, *Data Classification: Algorithms and Applications*, 2014.

[4] J. Attenberg, F. Provost. Why label when you can search? Alternatives to active learning for applying human resources to build classification models under extreme class imbalance. *In SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2010, pp. 423-432.

[5] N. Cesa-Bianchi, C. Gentile, L. Zaniboni. Worst-case analysis of selective sampling for linear classification. *Journal of Machine Learning Research*, 2006, No. 7, pp. 1205-1230.

[6] N. Cesa-Bianchi, A. Conconi, C. Gentile. A second-order perceptron algorithm. *SIAM Journal on Computing*, 2005, No. 3, pp. 640-668.

[7] S. Chakraborty, V. Balasubramanian, A. Sankar, S. Panchanathan, J. Ye. Batchrank: A novel batch mode active learning framework for hierarchical classification. *In SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015, pp. 99-108.

[8] C. C. Chang, C. J. Lin. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2011, Vol. 2, No. 3, pp. 27.

[9] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, Y. Singer. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 2006, pp. 551-585.

[10] K. Crammer, A. Kulesza, M. Dredze. Adaptive regularization of weight vectors. *In Advances in Neural Information Processing Systems*, 2009, pp. 414-422.

[11] M. Dundar, B. Krishnapuram, J. Bi, R. B. Rao. Learning classifiers when the training data is not IID. *In International Joint Conference on Artificial Intelligence*, 2007, pp. 756-761.

[12] M. Fang, X. Zhu, B. Li, W. Ding, X. Wu. Self-taught active learning from crowds. *In IEEE International Conference on Data Mining*, 2012, pp. 858-863.

[13] Z. Ferdowsi, R. Ghani, R. Settimi. Online active learning with imbalanced classes. *In IEEE International Conference on Data Mining.* 2013, pp. 1043-1048.

[14] K. Fujii, H. Kashima. Budgeted stream-based active learning via adaptive submodular maximization. *In Advances in Neural Information Processing Systems*, 2016, pp. 514-522.

[15] Y. Freund, R. E. Schapire. Large margin classification using the perceptron algorithm. *Machine learning*, 1999, No. 3, pp. 277-296.

[16] S. Hao, J. Lu, P. Zhao, C. Zhang, S. C. Hoi, C. Miao. Second-order online active learning and its applications. *IEEE Transactions on Knowledge and Data Engineering*, 2017.

[17] S. Hao, P. Zhao, J. Lu, S. C. Hoi, C. Miao, C. Zhang. Soal: Second-order online active learning. *In IEEE International Conference on Data Mining*, 2016, pp. 931-936.

[18] R. Horn, C. Johnson. Matrix analysis. *Cambridge University Express*, 1990.

[19] G. Hulten, L. Spencer, P. Domingos. Mining time-changing data streams. *In SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2001, pp. 97-106.

[20] S. C. Hoi, R. Jin, J. Zhu, M. R. Lyu. Batch mode active learning and its application to medical image classification. *In International Conference on Machine Learning*, 2006, pp. 417-424.

[21] S. J. Huang, J. L. Chen, X. Mu, Z. H. Zhou. Cost-Effective active learning from diverse labelers. *In International Joint Conference on Artificial Intelligence*, 2017, pp. 1879-1885.

[22] K. Konyushkova, R. Sznitman, P. Fua. Learning active learning from data. *In Advances in Neural Information Processing Systems*, 2017, pp. 4228–4238.

[23] A. Krishnamurthy, A. Agarwal, T. Huang, D. Hal and J. Langford. Active learning for cost-sensitive classification. *In International Conference on Machine Learning*, 2017, pp. 1915–1924.

[24] Y. Li, P. M. Long. The relaxed online maximum margin algorithm. *In Advances in Neural Information Processing Systems*, 2000, pp. 498-504.

[25] J. Lu, P. Zhao, S. C. Hoi. Online passive-aggressive active learning. *Machine Learning*, 2016, Vol. 103, No. 2, pp. 141-183.

[26] S. O. Moepya, S. S. Akhoury, F. V. Nelwamondo. Applying cost-sensitive classification for financial fraud detection under high class-imbalance. *In IEEE International Conference on Data Mining*, 2014, pp. 183-192.

[27] F. Nan, V. Saligrama. Adaptive classification for prediction under a budget. *In Advances in Neural Information Processing Systems*, 2017, pp. 4730–4740.

[28] V. S. Sheng, F. Provost, P. G. Ipeirotis. Get another label? improving data quality and data mining using multiple, noisy labelers. *In SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2008, pp. 614-622.

[29] J. Wang, P. Zhao and S. C. Hoi. Cost-sensitive online classification. *IEEE Transactions on Knowledge and Data Engineering*, 2014, vol. 26, no. 10, pp. 2425-2438.

[30] X. Zhang, T. Yang, P. Srinivasan. Online asymmetric active learning with imbalanced data. *In SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 2055-2064.

[31] Y. Zhang, G. Shu, Y. Li. Strategy-updating depending on local environment enhances cooperation in prisoners dilemma game. *Applied Mathematics and Computation*, 2017, vol. 301, pp. 224-232.

[32] P. Zhao, S. C. Hoi. Cost-sensitive online active learning with application to malicious URL detection. *In SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2013, pp. 919-927.

[33] P. Zhao, F. Zhuang, M. Wu, X. Li, and S. C. Hoi. Cost-sensitive online classification with adaptive regularization and its applications. *In IEEE International Conference on Data Mining*, 2015, pp. 649-658.

[34] P. Zhao, Y. Zhang, M. Wu, S. C. Hoi, M. Tan, J. Huang. Adaptive cost-sensitive online classification. *IEEE Transactions on Knowledge and Data Engineering*, 2018.

[35] I. Zliobaite, A. Bifet, B. Pfahringer, G. Holmes. Active learning with drifting streaming data. *IEEE Transactions on Neural Networks and Learning Systems*, 2014, Vol. 25, No. 1, pp. 27-39.