11-2017

Second-order online active learning and its applications

Shuji HAO
Institute of High Performance Computing

Jing LU
Singapore Management University, jing.lu.2014@phdis.smu.edu.sg

Peilin ZHAO
South China University of Technology

Chi ZHANG
Nanyang Technological University

Steven C. H. HOI
Singapore Management University, CHHOI@smu.edu.sg

See next page for additional authors

DOI: https://doi.org/10.1109/TKDE.2017.2778097

Citation

HAO, Shuji; LU, Jing; ZHAO, Peilin; ZHANG, Chi; HOI, Steven C. H.; and MIAO, Chunyan. Second-order online active learning and its applications. (2017). *IEEE Transactions on Knowledge and Data Engineering*. 30, (7), 1338-1351. Research Collection School Of Information Systems.
Available at: https://ink.library.smu.edu.sg/sis_research/4132

Author
Shuji HAO, Jing LU, Peilin ZHAO, Chi ZHANG, Steven C. H. HOI, and Chunyan MIAO

Second-order Online Active Learning and Its Applications

Shuji Hao, Jing Lu, Peilin Zhao, Chi Zhang, Steven C.H. Hoi and Chunyan Miao

Abstract—The goal of online active learning is to learn predictive models from a sequence of unlabeled data given limited label query budget. Unlike conventional online learning tasks, online active learning is considerably more challenging because of two reasons. Firstly, it is difficult to design an effective query strategy to decide when is appropriate to query the label of an incoming instance given limited query budget. Secondly, it is also challenging to decide how to update the predictive models effectively whenever the true label of an instance is queried. Most existing approaches for online active learning are often based on a family of first-order online learning algorithms, which are simple and efficient but fall short in the slow convergence and sub-optimal solution in exploiting the labeled training data. To solve these issues, this paper presents a novel framework of Second-order Online Active Learning (SOAL) by fully exploiting both the first-order and second-order information. The proposed algorithms are able to achieve effective online learning efficacy, maximize the predictive accuracy and minimize the labeling cost. To make SOAL more practical for real-world applications, especially for class-imbalanced online classification tasks (e.g., malicious web detection), we extend the SOAL framework by proposing the Cost-sensitive Second-order Online Active Learning algorithm named "SOAL$_{CS}$", which is devised by maximizing the sum of weighted sensitivity and specificity or minimizing the cost of weighted mistakes of different classes. We conducted both theoretical analysis and empirical studies, including an extensive set of experiments on a variety of large-scale real-world datasets, in which the promising empirical results validate the efficacy and scalability of the proposed algorithms towards large-scale online learning tasks.

Index Terms—Online Learning, Active Learning, Malicious websites detection.

1 Introduction

Online Learning is an active research area in machine learning for processing large-scale learning tasks. This area has been extensively studied in literature for its high efficiency and scalability in processing big data streams [1], [2], [3], [4], [5], [6], [7], [8], [9]. Different from the traditional batch-based machine learning algorithms which require the availability of all data before training the models, online learning typically works in a sequential manner. We take an online binary classification task as an example. At time t, the learner only receives one instance $\mathbf{x}_t$ from the environment and then makes a prediction of its class label $\hat{y}_t = \text{sign}(f(\mathbf{x}_t))$, where f is a classifier that maps the feature vector $\mathbf{x}_t$ to a real value classification score. After making the prediction, it usually assumes that the true label $y_t \in \{+1, -1\}$ will be revealed from the environment and then updates the classifier whenever necessary, for example, when the leaner makes a mistake ($\hat{y}_t \neq y_t$). In contrast to traditional batch learning, which often suffers from expensive re-training cost when new training data comes, online learning avoids re-training and learns incrementally from data streams, which makes them much more efficient.

Although a variety of online learning algorithms have been proposed over the past decades [10], [11], [12], [13], [14], [15], [16], [17], [18], conventional fully supervised online learning algorithms usually assume that the ground truth (e.g., the class labels in classification tasks) is always available to the learner at the end of each iteration. However, in many real applications, the dataset is usually large and unlabeled, and manually labeling all the instances is usually too expensive to afford meanwhile. For example, in the social media platforms, data stream usually comes with a high speed and volume, which makes it costly or nearly infeasible to label all of the instances. This has raised a challenging problem of how to minimize the number of instances to be labeled and train a well-performed learner meanwhile (i.e. designing effective query strategies which can automatically select a subset of most informative instances to label).

To address this challenge, researchers have proposed a serial of "Online Active Learning" algorithms [19], [20], [21], [22] in recent years. A pioneering study is the "Perceptron-based active learning" [23]. The learner in [23] decides when to query by drawing a Bernoulli random variable $Z_t \in \{0, 1\}$ with parameter $\delta/(\delta + |p_t|)$, where $|p_t|$ is the margin value (the distance of the instance to the prediction hyperplane) of $\mathbf{x}_t$ and $\delta > 0$ is a sampling parameter to control the labeling budget. If and only if $Z_t = 1$, the learner will then place a query to ask an external oracle to give true label of the current instance. The intuitive idea is to query the instances nearby the hyperplane as they are harder to be correctly predicted and thus more informative. This similar approach has also been used by the online Passive Aggressive (PA) learners in recent studies [7]. Despite their simplicity, these algorithms often suffer some critical limitations. First, they often adopt first-order based online learning algorithms as the prediction model, whose performance is usually limited as all dimensions share same

- *Shuji Hao is with the Institute of High Performance Computing, Agency for Science Technology and Research, Singapore.*
  E-mail: haosj@ihpc.a-star.edu.sg
- *Jing Lu and Steven C.H. Hoi are with School of Information Systems, Singapore Management University.*
  E-mail: jing.lu.2014@phdis.smu.edu.sg, chhoi@smu.edu.sg
- *Peilin Zhao is with School of Software Engineering, South China University of Technology, Guangzhou, China.*
  E-mail: peilinzhao@hotmail.com
- *Chi Zhang is with Interdisciplinary Graduate School, Nanyang Technological University, Singapore and Tencent Lab, China.*
  E-mail: czhang024@e.ntu.edu.sg
- *Chunyan Miao is with School of Computer Engineering, Nanyang Technological University, Singapore.*
  E-mail: ascymiao@ntu.edu.sg
- *Steven C.H. Hoi and Peilin Zhao are corresponding authors.*

Manuscript received ; revised .

learning rate when the model is updated. Second, as the margin $|p_t|$ only depends on the classifier $\mathbf{w}_t$, the query strategy would be sub-optimal when the classifier $\mathbf{w}_t$ is not precise. For example, in the early rounds of learning, the margin value may be not accurate as the classifier $\mathbf{w}_t$ is not trained well with sufficient samples.

To overcome these limitations, we present a new algorithm, Second-order Online Active Learning (SOAL), which explores second-order online learning techniques for both training the classifiers and forming the query strategy. Specifically, we devise a novel query strategy, which enables to query the most informative instances by exploiting both margin and second-order confidence information, and the proposed algorithm SOAL also takes advantages of the second-order information which enables each dimension of the model $\mathbf{w}$ to be update with different and adaptive learning rate. In addition, to tackle the issue of accuracy evaluation metric on the imbalanced tasks, such as malicious web sites detection [24] etc, we proposed the Cost-sensitive Second-order Online Active Learning (SOAL$_{CS}$) algorithm, which aims to maximize the sum of weighted sensitivity and specificity or minimize the cost of weighted mistakes.

When compared with the first-order based online active learning, our proposed algorithms are different in the following aspects: (1) most of the existing algorithms only update a single weight vector $\mathbf{w}$ during the online learning process, where not enough information is used for effective updates. While in our proposed algorithm, we learn not only the mean but also the possible distribution of the $\mathbf{w}$, which leads to a faster convergence rate. Specially, we will demonstrate that our proposed algorithm updates each dimension of the weight vector $\mathbf{w}$ with a different learning rate, depending on the confidence it has on this particular dimension; (2) existing active learning algorithms usually query instances with the smallest distance to the decision boundary, which however, might be misleading when the decision boundary itself is not well trained. While in the proposed SOAL algorithm, the variance of the distance to the decision boundary is also considered. Consequently, the instances selected for labeling in our proposed algorithm are more informative than those of the existing algorithms.

To evaluate the performance of the proposed algorithm SOAL, we conduct both theoretical analysis and empirical studies that investigate the algorithm in terms of accuracy, parameter sensitivity and scalability. Furthermore, we also apply the proposed Cost-sensitive algorithm (SOAL$_{CS}$) to several malicious websites detection datasets. Encouraging results show clear advantages of the proposed algorithm over a family of state-of-the-art online active learning algorithms. In summary, the main contributions of this work are as follows:

- We propose a novel second-order based online active learning algorithm ($SOAL$) for the binary classification problems, in which the proposed query strategy considers not only the uncertainty of the prediction but also the confidence of the classification model.
- To tackle the imbalance problem in practice, we also propose a novel second-order based online active learning ($SOAL_{CS}$) by considering different loss on different class.
- To evaluate the performance of the proposed algorithms, we first present theoretical analysis for the $SOAL$ algorithm, and then conduct empirical studies from several aspects, such as varied query ratio, parameter sensitivity, scalability etc.

It should be noted that a short version of this work has been published as a conference paper [25].

2 Related Work

Our work is related to three major groups of studies in machine learning literature: (i) online learning, (ii) active learning and (iii) cost-sensitive learning.

2.1 Online Learning

Online learning has been an active research topic in machine learning community [10], [11], [12], [13], [14], [15], [16], [26], in which a variety of online learning models has been proposed. Typically, based on the model updating strategy, the existing online learning algorithms can be categorized into two main groups: (i) first-order based online learning, where only the first-order feature information is exploited, (ii) Second-order based online learning, which maintains not only the first-order feature information but also the second-order information, such as the covariance matrix of the feature information.

In the first-order based online learning algorithms, one of the most well-known ones is the Perceptron algorithm [27], [28], which updates the learner by adding or subtracting the misclassified instance with a fixed weight to the current set of support vectors. Recently, several works also studied the first-order based online learning algorithms by maximizing the margin value. One pioneer work is the Relaxed Online Maximum Margin Algorithm (ROMMA) [29], which repeatedly chooses the classifier which can correctly classify the existing training instances with a large margin. Another work is the Passive-Aggressive algorithms (PA) [30], which updates the current model when the current instance is misclassified or its prediction value doesn't reach a predefined margin value. By examining the empirical performance of these first-order based online learning algorithms, we can observe that the large margin algorithms can generally outperform the Perceptron algorithm. However, the performance of these large margin algorithms is still restricted as only the first-order information is adopted.

In recent years, researchers have been actively designing second-order based online learning algorithms in order to overcome the limitations of first-order based algorithms. Generally, the performance of second-order based algorithms have been significantly improved by exploring the parameter confidence information (second-order information). One of the well-known second-order models is the Second-Order Perceptron algorithm (SOP) [31], which is usually viewed as a variant of the whitened Perceptron algorithm. The authors explore the online correlation matrices of the previously seen instances to achieve the whitened effect. Later, several large margin second-order online learning algorithms are also proposed, such as Confidence-Weighted (CW) learning [32], which maintains a Gaussian distribution over the model parameters and uses the covariance of the parameters to guide the update of each parameter. Although CW is promising both in theory and empirical studies, it may suffer from its aggressive hard margin update strategy in noisy data. To tackle this limitation, researchers have proposed improved versions, such as the Adaptive Regularization of Weights algorithm (AROW) [33] and Soft Confidence-Weighted algorithms [34] by employing an adaptive regularization for each training instance. In general, the second order algorithms can consistently converge faster and perform better than the first-order based algorithms.

2.2 Active Learning

The goal of active learning is to train a well-performed predictive model by actively selecting a small subset of informative instances whose labels will be queried. As active learning can largely reduce the labeling cost, it has been extensively studied in the batch-based learning scenarios [20], [21]. Existing active learning techniques could be generally grouped into four categories: (1) uncertainty-based query strategies [7], [22], [35], where instances with the lowest prediction confidence are queried; (2) disagreement-based query strategies [36], [37], [38], which query the instances on which the hypothesis space has the most disagreement degree on their predictions; (3) labeling the instances which could minimize the expected error and variance on the pool of unlabeled instances [39] and (4) exploiting the structure information among the instances [40]. More about batch-based active learning studies can be found in the comprehensive survey [19], [41].

Batch-based active learning algorithms are effective in reducing labeling cost in several applications, such as text classification, image recognitions and abnormal detection. However, these algorithms typically require that all of the data should be collected firstly before the active learning process. This makes them infeasible in some real-world applications, such as in online social media platforms, where data usually comes in a sequential manner. To overcome this challenge, researchers have studied online active learning (OAL) [2], [7], [22], [42], also known as selective sampling, which aims to learn predictive models from a sequence of unlabeled data given limited label query budget. These online active learning algorithms typically adopt first-order based query strategies, such as margin-based query strategy. This makes the algorithms suffer from two major limitations. First, the performance (in terms of accuracy) of these algorithms is usually limited as most of them adopt first-order based predictive models. Second, their active query strategies often strongly rely on the predictive model $\mathbf{w}_t$, which may not be precise in the early rounds of online learning. The work in this paper aims to tackle these limitations by proposing a new online active learning method going beyond the existing first-order learning approaches.

2.3 Cost-sensitive Classification

Cost-Sensitive classification, has been widely used in malicious web detection, credit fraud detection and medical diagnosis, where the cost of misclassify a malicious or fraud target (false-negative) is much higher than that of a false-positive. Traditional classification algorithms would be inappropriate since they adopt accuracy as evaluation metric and treat the cost of a false-negative and a false-positive equally. To overcome this limitation, researchers have investigated a variate of cost-sensitive metrics. Two of the most well-known algorithms are called the weighted sum of *sensitivity* and *specificity* [43] and the weighted *misclassification cost* [44]. In the past decades, several batch learning algorithms have been proposed for these cost-sensitive metrics [45], [46], [47]. Besides, a few cost-sensitive online learning algorithms are also proposed recently. However, these algorithms either are based on first-order learning algorithms [48], or assume that all the instances are well labeled [48]. In this article, we propose a new second-order online active learning algorithm not only to reduce the labeling cost but also to improve the cost-sensitive performance.

3 Second-order Online Active Learning (SOAL)

Generally, there are two open challenges when designing an online active learning algorithm. (i) "When to query", i.e. how to design an effective query strategy that can query the most informative unlabeled examples for training. (ii) "How to update", i.e. how to update the learner effectively whenever a query has been placed and the feedback is revealed to the learner. In this section, we present a new framework of Second-order Online Active Learning to solve both the "When to query" and the "How to update" challenges.

3.1 Problem Formulation

In this work, we consider a typical online binary classification task. A learner iteratively learns from a sequence of training instances $\{(\mathbf{x}_t, y_t) \,|t = 1, \ldots, T\}$, where $\mathbf{x}_t \in \mathbb{R}^d$ is the feature vector of the t-th instance and $y_t \in \{-1, +1\}$ is its true class label. The goal of online binary classification is to learn a linear classifier $\hat{y}_t = \text{sign}(\mathbf{w}_t^\top \mathbf{x}_t)$, where $\mathbf{w}_t \in \mathbb{R}^d$ is the weight vector at the t-th round.

Unlike regular supervised online learning, when receiving $\mathbf{x}_t$, an online active learning algorithm needs to decide whether to query the true label y_t or not. If the algorithm decides to query the true label, an external oracle (e.g. an expert in this task who is able to give correct label) will be asked to give the true label. Once the true label is revealed, the algorithm may suffer some positive loss and adopt regular online learning techniques to update the model $\mathbf{w}_t$. Otherwise, the algorithm will ignore the instance and process the next one. In this way, online active learning aims to query a small fraction of informative instances for the true labels and at the same time achieve a comparable accuracy with the regular online learning algorithms which query all of the instances for true labels.

In this article, we assume that the classifier $\mathbf{w}$ follows a Gaussian distribution [33], [34], [48], [49], i.e., $\mathbf{w} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. The values μ_i and $\Sigma_{i,i}$ encode the model's knowledge of and confidence in the weight for i-th feature w_i: the smaller the value of $\Sigma_{i,i}$ is, the more confident the learner is in the mean weight value μ_i. The covariance term $\Sigma_{i,j}$ captures interactions between w_i and w_j. In practice, it is often easier to simply use the expectation of weight vector $\boldsymbol{\mu} = \mathbb{E}[\mathbf{w}]$ as the classifier to make predictions.

3.2 SOAL Algorithm

The proposed algorithm SOAL mainly consists of two parts: 1) "How to update" presents the updating rule of the classifier $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ whenever the true label of an instance is revealed; 2) "When to query" presents the proposed second-order based query strategy which decides when to query an unlabeled instance for the true label. We discuss each part in detail as follows.

3.2.1 How to Update

The idea to design the learning object function is three folds: 1) the learnt new model shall suffer small loss on the current training instance; 2) the learnt new model shall not make too large updating step from the previous model [49]; 3) the learnt model shall be more confident on its prediction on the future instances which are same or similar as the current training instance. Specifically, at the t-th round, if the true label y_t of $\mathbf{x}_t$ is revealed, we will update the model to make sure that it suffers small loss on t-th instance and has high confidence on its prediction. Formally, we want to update the Gaussian distribution by minimizing the following objective function

$$\mathcal{C}_t(\boldsymbol{\mu},\boldsymbol{\Sigma}) = D_{KL}(\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})\|\mathcal{N}(\boldsymbol{\mu}_t,\boldsymbol{\Sigma}_t)) + \eta\mathbf{g}_t^\top\boldsymbol{\mu} + \tfrac{1}{2\gamma}\mathbf{x}_t^\top\boldsymbol{\Sigma}\mathbf{x}_t, \quad (1)$$

The first term is to keep the new model not far away from the previous model. The second term is to minimize the (linearized) loss of the new model on the current example. The final term is to minimize the variance of prediction margin value.

In Eq. (1),

$$\begin{aligned}&D_{KL}(\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})\|\mathcal{N}(\boldsymbol{\mu}_t,\boldsymbol{\Sigma}_t))\\ =&\frac{1}{2}\log\left(\frac{\det\boldsymbol{\Sigma}_t}{\det\boldsymbol{\Sigma}}\right)+\frac{1}{2}\mathrm{Tr}(\boldsymbol{\Sigma}_t^{-1}\boldsymbol{\Sigma})+\frac{1}{2}\|\boldsymbol{\mu}_t-\boldsymbol{\mu}\|^2_{\boldsymbol{\Sigma}_t^{-1}}-\frac{d}{2},\end{aligned} \quad (2)$$

$\mathbf{g}_t = \partial\ell_t(\boldsymbol{\mu}_t) = -y_t\mathbf{x}_t$, $\eta > 0$ and $\gamma > 0$ are two positive regularization parameters. $\ell_t(\boldsymbol{\mu}_t) = \max(0, 1 - y_t\boldsymbol{\mu}_t^T\mathbf{x}_t)$ is the hinge loss function adopted.

When $\ell_t(\boldsymbol{\mu}_t) > 0$, we solve the above minimization in the following two steps:

- Update the confidence matrix parameters:

$$\boldsymbol{\Sigma}_{t+1} = \arg\min_{\boldsymbol{\Sigma}}\mathcal{C}_t(\boldsymbol{\mu},\boldsymbol{\Sigma});$$

- Update the mean parameters:

$$\boldsymbol{\mu}_{t+1} = \arg\min_{\boldsymbol{\mu}}\mathcal{C}_t(\boldsymbol{\mu},\boldsymbol{\Sigma});$$

For the first step, by setting the derivative $\partial_{\boldsymbol{\Sigma}}\mathcal{C}_t(\boldsymbol{\mu},\boldsymbol{\Sigma}_{t+1}) = 0$, we can derive the closed-form update:

$$\boldsymbol{\Sigma}_{t+1} = \boldsymbol{\Sigma}_t - \frac{\boldsymbol{\Sigma}_t\mathbf{x}_t\mathbf{x}_t^\top\boldsymbol{\Sigma}_t}{\gamma + \mathbf{x}_t^\top\boldsymbol{\Sigma}_t\mathbf{x}_t}, \quad (3)$$

where the Woodbury identity is used.

For the second step, by setting $\partial_{\boldsymbol{\mu}}\mathcal{C}_t(\boldsymbol{\mu}_{t+1},\boldsymbol{\Sigma}) = 0$, we can derive the closed-form update:

$$\boldsymbol{\mu}_{t+1} = \boldsymbol{\mu}_t - \eta\boldsymbol{\Sigma}_t\mathbf{g}_t,$$

Since the update of the mean relies on the confidence parameter, we try to update the mean based on the updated covariance matrix $\boldsymbol{\Sigma}_{t+1}$, i.e.,

$$\boldsymbol{\mu}_{t+1} = \boldsymbol{\mu}_t - \eta\boldsymbol{\Sigma}_{t+1}\mathbf{g}_t, \quad (4)$$

which should be more accurate than the update in Equation (4).

In order to handle high-dimensional data, we can only keep the diagonal elements of $\boldsymbol{\Sigma}$ and the updating rules in Equation (3) and (4) becomes

$$\begin{aligned}\boldsymbol{\Sigma}_{t+1} &= \boldsymbol{\Sigma}_t - \frac{\boldsymbol{\Sigma}_t\odot\mathbf{x}_t\odot\mathbf{x}_t\odot\boldsymbol{\Sigma}_t}{\gamma + (\mathbf{x}_t\odot\boldsymbol{\Sigma}_t)^\top\mathbf{x}_t}, & (5)\\ \boldsymbol{\mu}_{t+1} &= \boldsymbol{\mu}_t - \eta\boldsymbol{\Sigma}_{t+1}\odot\mathbf{g}_t, & (6)\end{aligned}$$

where $\odot$ denotes the element-wise multiplication.

Remark: By comparing the above updating equation with first-order based updating rules, such as Eq. (3) in [30], we can observe that the above updating rule assigns different feature dimension with **different** learning rate via $\boldsymbol{\Sigma}$, so that the less confident weights will be updated more aggressively (the diagonal value of $\boldsymbol{\Sigma}_{t+1}$ would be big). However, the updating rules in the first-order based algorithms [30] assign different feature dimension with **same** learning rate, thus less confident weights will be updated equally as the confident weights.

3.2.2 When to Query

In the "How to Update" section, we solved the challenge of how to update the classifier $\boldsymbol{\mu}$ whenever we receive the true label y_t of $\mathbf{x}_t$, in this section, we propose a novel second-order based query strategy to solve the "When to Query" challenge by considering two factors as follows.

The first factor is the margin value $|p_t| = |\boldsymbol{\mu}_t^\top\mathbf{x}_t|$, which represents how far the instance is away from the current classifier hyperplane $\mathbf{w}_t$. The smaller the value of $|p_t|$ is, the more uncertain the classifier is about its prediction on the instance $\mathbf{x}_t$, and the instance should have a higher chance to be queried for the true label.

This margin value has been extensively adopted in existing online active learning algorithms [2], [7], [22]. However, we can observe that the margin value p_t is directly depending on the precision of learned classifier $\boldsymbol{\mu}_t$. If $\boldsymbol{\mu}_t$ is precise, p_t would be accurate. However, when $\boldsymbol{\mu}_t$ is not precise, such as in the early rounds of learning process, p_t would not be precise and thus affects the query strategy. To overcome this limitation, our proposed query strategy not only considers this margin value p_t, but also considers a second factor which describes how confident the model is on its prediction.

Specifically, the second factor is defined as

$$c_t = \frac{1}{2}\frac{-\eta}{\frac{1}{v_t}+\frac{1}{\gamma}}, \quad (7)$$

where $\eta > 0$, $\gamma > 0$ are two fixed hyper-parameters and $v_t = Var[\boldsymbol{\mu}_t^\top\mathbf{x}_t] = \mathbf{x}_t^\top\boldsymbol{\Sigma}_t\mathbf{x}_t$ is the only variable, which models the variance of the margin value of $\mathbf{x}_t$. In other words, v_t characterizes how often the instances which are similar as $\mathbf{x}_t$ have been seen by the classifier $\boldsymbol{\mu}$ in the past t-th round. Specifically,

- when c_t is small (v_t is large), the classifier has not been well trained on the instances which are similar to $\mathbf{x}_t$ so far and it's necessary to place *high probability* to query the true label;
- when c_t is large (v_t is small), the classifier has been well trained on the instances which are similar to $\mathbf{x}_t$ so far and we should place a *low probability* to query the true label y_t .

By combining these two terms together, we can compute the term

$$\rho_t = |p_t| + c_t. \quad (8)$$

This equation servers as a *soft* version of margin-based query strategy, where the query decision not only depends on the margin value (uncertainty), but also considers the correctness or confidence of this predicted margin value.

There are two cases to be considered. When $\rho_t \le 0$, i.e. the model is extremely not confident on the trained classifier, we **always query** the label of instance no matter how large $|p_t|$ is. Compared to the traditional query strategy [7], [23] where a large value of $|p_t|$ always results in a **small query probability** no matter how unreliable the current classifier is, our proposed strategy is more reasonable.

When $\rho_t > 0$, i.e. the model is confident on the trained classifier (c_t is large), the margin value $|p_t|$ computed based on the trained weight vector is reliable. In this situation, we draw a Bernoulli random variable $Z_t \in \{0, 1\}$ of parameter $\frac{\delta}{\delta+\rho_t}$, where $\delta > 0$ is a smoothing parameter. Here, ρ_t contains both the first-order information p_t and the second-order information v_t, which is more reliable than the margin value p_t alone. Formally,

- If $\rho_t \le 0$, query y_t;

- Else $\rho_t > 0$, draw a Bernoulli random variable $Z_t \in \{0, 1\}$ with $\Pr(Z_t = 1) = \frac{\delta}{\delta+\rho_t}$;
  - If $Z_t = 1$, query true label y_t;
  - Else $Z_t = 0$, discard $\mathbf{x}_t$.

Remark: By comparing to the margin-based query strategies in previous studies [7], [22], our proposed strategy not only considers the margin value p_t (which describes how far the instance is away from the classifier hyperplane), but also considers the confidence value c_t (which describes how well the classifier is trained on the instances which are similar as current instance $\mathbf{x}_t$). In precious studies, if p_t is small, the margin-based query strategy would make a query with a high probability no matter how large c_t is (the classifier is already well trained on the instances which are similar to $\mathbf{x}_t$ so far and querying $\mathbf{x}_t$ is not necessary); however, in our proposed strategy shown in Eq. (8), even p_t is small, the chance to make a query would be reduced if c_t is large. Thus, our proposed query method would be more effective than the margin-based strategy.

Finally, Algorithm 1 summarizes the proposed algorithm.

Algorithm 1 SOAL:Second-order Online Active Learning.

```
Input: learning rate $\eta$; regularization parameter $\gamma$, smoothing parameter $\delta$.
Initialize: $\boldsymbol{\mu}_1 = \mathbf{0}$, $\boldsymbol{\Sigma}_1 = I$.
for $t = 1, \ldots, T$ do
  Receive $\mathbf{x}_t \in \mathbb{R}^d$;
  Compute $p_t = \boldsymbol{\mu}_t^\top \mathbf{x}_t$;
  Make prediction $\hat{y}_t = \mathrm{sign}(p_t)$;
  Compute $\rho_t = |p_t| + c_t$, where $c_t = \frac{1}{2}\frac{-\eta}{\frac{1}{v_t}+\frac{1}{\gamma}}$;
  if $\rho_t > 0$ then
    Draw Bernoulli random variable $Z_t \in \{0, 1\}$ of parameter $\frac{\delta}{\delta+\rho_t}$;
  else
    $Z_t = 1$;
  end if
  if $Z_t = 1$ then
    Query $y_t \in \{-1, +1\}$;
    Compute $\ell_t(\boldsymbol{\mu}_t) = [1 - y_t \mathbf{x}_t^\top \boldsymbol{\mu}_t]_+$;
    if $\ell_t > 0$ then
      $\boldsymbol{\Sigma}_{t+1} = \boldsymbol{\Sigma}_t - \frac{\boldsymbol{\Sigma}_t \mathbf{x}_t \mathbf{x}_t^\top \boldsymbol{\Sigma}_t}{\gamma + \mathbf{x}_t^\top \boldsymbol{\Sigma}_t \mathbf{x}_t}, \boldsymbol{\mu}_{t+1} = \boldsymbol{\mu}_t - \eta \boldsymbol{\Sigma}_{t+1} \mathbf{g}_t$, or
      $\boldsymbol{\Sigma}_{t+1} = \boldsymbol{\Sigma}_t - \frac{\boldsymbol{\Sigma}_t \odot \mathbf{x}_t \odot \mathbf{x}_t \odot \boldsymbol{\Sigma}_t}{\gamma + (\mathbf{x}_t \odot \boldsymbol{\Sigma}_t)^\top \mathbf{x}_t}, \boldsymbol{\mu}_{t+1} = \boldsymbol{\mu}_t - \eta \boldsymbol{\Sigma}_{t+1} \odot \mathbf{g}_t$;
    end if
  end if
end for
```

3.3 Theoretical Analysis

To be concise, we introduce two notations:

$$M_t = \mathbb{I}(\hat{y}_t \neq y_t), L_t = \mathbb{I}(\ell_t(\boldsymbol{\mu}_t) > 0, \hat{y}_t = y_t).$$

Next we would analyze the performance of the proposed algorithm in terms of expected mistake bound $\mathbb{E}[\sum_{t=1}^{T} M_t]$.

Theorem 1. *Let $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_T, y_T)$ be a sequence of input examples, where $\mathbf{x}_t \in \mathbb{R}^d$ and $y_t \in \{-1, +1\}$ for all t. If the SOAL algorithm is run on this sequence of examples, then the expected number of prediction mistakes made is bounded from above by the following inequality, for any vector $\boldsymbol{\mu} \in \mathbb{R}^d$,*

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{T} M_t\right] \\
&\leq \mathbb{E}\left[\sum_{t=1}^{T} Z_t \ell_t(\boldsymbol{\mu})\right] + \frac{D_{\boldsymbol{\mu}} + (1-\delta)^2 \|\boldsymbol{\mu}\|^2}{\eta\delta} \mathrm{Tr}(\boldsymbol{\Sigma}_{T+1}^{-1}) \\
&\quad + \frac{1}{\delta}\mathbb{E}\left[\sum_{\rho_t<0} \frac{\eta\gamma v_t}{(\gamma + v_t)}\right] + \frac{2}{\delta}\mathbb{E}\left[\sum_{\rho_t>0} L_t\right] - \mathbb{E}\left[\sum_{t=1}^{T} L_t\right]
\end{aligned}$$

where $\delta > 0$, $D_{\boldsymbol{\mu}} = \max_{t \leq T} \|\boldsymbol{\mu}_t - \boldsymbol{\mu}\|^2$.

Remark: First, when $\gamma = 1$, $\mathbb{E}\sum_{\rho_t<0} \frac{\gamma v_t}{(\gamma+v_t)} \leq \sum_{i=1}^{d} \ln(1+\lambda_i)$, where the right-hand side is used in the Theorem 3 of [22], which implies our term is better.

Second, since

$$\mathbb{E}\sum_{\rho_t<0} \frac{\gamma v_t}{(\gamma + v_t)} \leq \mathbb{E}\sum_{t} \frac{\gamma v_t}{(\gamma + v_t)} \leq \gamma \mathbb{E} \ln(\left|\boldsymbol{\Sigma}_{T+1}^{-1}\right|),$$

if $\eta = \sqrt{\frac{(D_{\boldsymbol{\mu}} + (1-\delta)^2 \|\boldsymbol{\mu}\|^2) \mathrm{Tr}(\boldsymbol{\Sigma}_{T+1}^{-1})}{\gamma \ln |\boldsymbol{\Sigma}_{T+1}^{-1}|}}$, we have the following expected mistake bound,

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{T} M_t\right] \\
&\leq \mathbb{E}\sum_{t=1}^{T} Z_t \ell_t(\boldsymbol{\mu}) + \frac{2}{\delta}\mathbb{E}\left[\sum_{\rho_t>0} L_t\right] - \mathbb{E}\left[\sum_{t=1}^{T} L_t\right] \\
&\quad + \frac{2}{\delta}\sqrt{D_{\boldsymbol{\mu}} + (1-\delta)^2 \|\boldsymbol{\mu}\|^2} \sqrt{\gamma \mathrm{Tr}(\boldsymbol{\Sigma}_{T+1}^{-1}) \ln \left|\boldsymbol{\Sigma}_{T+1}^{-1}\right|}.
\end{aligned}$$

4 Cost-sensitive Second-order Online Active Learning algorithms

In the algorithm SOAL 1, we proposed maximizing the accuracy in classification problems based on the assumption that the numbers of the instances from the two classes are roughly balanced. However, this assumption is usually hard to meet. For example, in the abnormal detection problems, the number of abnormal instances is usually limited. In these problems, it would be infeasible to maximize the accuracy as a trivial learner which simply classifies all samples as normal could still achieve a high accuracy. Thus, more appropriate performances metric should be adopted. We first propose to maximize the weighted sum of *sensitivity* and *specificity*,

$$sum = \alpha_p \times \frac{T_p}{T_p + F_n} + \alpha_n \times \frac{T_n}{T_n + F_p}, \tag{9}$$

where T_p and F_n are the number of true positives and false negatives, T_n and F_p denote the number of true negatives and false positives, $\alpha_p + \alpha_n = 1$ and $0 \leq \alpha_p, \alpha_n \leq 1$, which are two parameters that controls the trade-off between sensitivity and specificity. It should be noted that when $\alpha_p = \alpha_n = 0.5$, the corresponding *sum* equals the *accuracy* metric used in balanced datasets. Generally, we pursue a higher *sum* value when designing the classification models in imbalanced datasets. An alternative evaluation metric is to evaluate the total mis-classification cost,

$$cost = c_p \times F_n + c_n \times F_p, \tag{10}$$

where $c_p + c_n = 1$ and $0 \le c_p, c_n \le 1$ are the mis-classification cost parameters for positive and negative classes, respectively. In general, the lower the *cost* value, the better the classification performance.

Based on these two metrics, the objective of a classification algorithm is either to maximize the *sum* or minimize *cost* as shown in [48], [50], which can be unified into minimizing the following objective:

$$\sum_{y_t=+1} \theta \mathbb{1}(y_t \boldsymbol{\mu} \cdot \mathbf{x}_t < 0) + \sum_{y_t=-1} \mathbb{1}(y_t \boldsymbol{\mu} \cdot \mathbf{x}_t < 0), \qquad (11)$$

where $\mathbb{1}(x)$ is an indicator function. When $\theta = \frac{\alpha_p T_n}{\alpha_n T_p}$, the objective function equals to maximize the *sum* metric, and when $\theta = \frac{c_p}{c_n}$, it equals to minimize the *cost* metric. And it should be noted this objective function is not convex, thus we replace it by its convex surrogate:

$$\begin{aligned} &\ell^{CS}(\boldsymbol{\mu}; (\mathbf{x}, y)) \\ &= \max\left(0, (\theta \mathbb{1}(y=1) + \mathbb{1}(y=-1)) - y(\boldsymbol{\mu} \cdot \mathbf{x})\right). \end{aligned} \qquad (12)$$

Based on the cost-sensitive loss ℓ^{CS} defined, we assume the model follows a Gaussian distribution as described in Section (3.1), and the updating rule of the model could be obtained by minimizing the following cost-sensitive object function

$$\begin{aligned} &\mathcal{C}_t^{CS}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \\ &= D_{KL}(\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \| \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)) + \eta \mathbf{g}_t^\top \boldsymbol{\mu} + \frac{1}{2\gamma} \mathbf{x}_t^\top \boldsymbol{\Sigma} \mathbf{x}_t, \end{aligned}$$

where g_t is the gradient of cost-sensitive loss function ℓ^{CS} over $\boldsymbol{\mu}$ variable.

When $\ell^{CS}(\boldsymbol{\mu}_t; (\mathbf{x}_t, y_t)) > 0$, we update the $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ iteratively by setting the derivative of $\mathcal{C}_t^{CS}$ over $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ to zero, respectively, and this can give us the closed-form updating rule as follows:

$$\boldsymbol{\Sigma}_{t+1} = \boldsymbol{\Sigma}_t - \frac{\boldsymbol{\Sigma}_t \mathbf{x}_t \mathbf{x}_t^\top \boldsymbol{\Sigma}_t}{\gamma + \mathbf{x}_t^\top \boldsymbol{\Sigma}_t \mathbf{x}_t}, \qquad (13)$$

$$\boldsymbol{\mu}_{t+1} = \boldsymbol{\mu}_t - \eta \boldsymbol{\Sigma}_{t+1} \mathbf{g}_t, \qquad (14)$$

For the high dimensional tasks, we also can adopt the diagonal version of $\boldsymbol{\Sigma}$ as follows:

$$\begin{aligned} \boldsymbol{\Sigma}_{t+1} &= \boldsymbol{\Sigma}_t - \frac{\boldsymbol{\Sigma}_t \odot \mathbf{x}_t \odot \mathbf{x}_t \odot \boldsymbol{\Sigma}_t}{\gamma + (\mathbf{x}_t \odot \boldsymbol{\Sigma}_t)^\top \mathbf{x}_t}, && (15) \\ \boldsymbol{\mu}_{t+1} &= \boldsymbol{\mu}_t - \eta \boldsymbol{\Sigma}_{t+1} \odot \mathbf{g}_t, && (16) \end{aligned}$$

where $\odot$ denotes the element-wise multiplication.

It should be noted that there is no much difference of the updating rules between the cost-insensitive algorithm in Section 3 and the cost-sensitive algorithm defined here, and the only difference is how the loss function is defined. In the cost-insensitive algorithm, the loss will treat the positive and the negative equally. While in the cost-sensitive algorithm, the false negative one would suffer more loss such that the model can make fewer mistakes on positive ones in future.

It should also be noted that this cost-sensitive algorithm is fully-supervised, which makes it quite expensive to query all of the instances labels, especially for the abnormal detection problems. To alleviate this labeling cost, we adopt the second-order query strategy proposed in Section (3.2).

Finally, Algorithm 2 summarizes the proposed cost-sensitive second-order based online active learning algorithm.

Algorithm 2 SOAL$_{CS}$: Cost-sensitive Second-order Online Active Learning.

```
Input: learning rate $\eta$; regularization parameter $\gamma$, bias parameter $\theta = \frac{\alpha_p T_n}{\alpha_n T_p}$ for sum and $\theta = \frac{c_p}{c_n}$ for cost, smoothing parameter $\delta$.
Initialize: $\boldsymbol{\mu}_1 = \mathbf{0}$, $\boldsymbol{\Sigma}_1 = I$.
for $t = 1, \dots, T$ do
  Receive $\mathbf{x}_t \in \mathbb{R}^d$;
  Compute $p_t = \boldsymbol{\mu}_t^\top \mathbf{x}_t$;
  Make prediction $\hat{y}_t = \mathrm{sign}(p_t)$;
  Compute $\rho_t = |p_t| + c_t$, where $c_t = \frac{1}{2} \frac{-\eta}{\frac{1}{v_t} + \frac{1}{\gamma}}$;
  if $\rho_t > 0$ then
    Draw Bernoulli random variable $Z_t \in \{0, 1\}$ of parameter $\frac{\delta}{\delta + \rho_t}$
  else
    $Z_t = 1$;
  end if
  if $Z_t = 1$ then
    Query $y_t \in \{-1, +1\}$;
    Compute $\theta_t = \theta \mathbb{1}(y = 1) + \mathbb{1}(y = -1)$;
    Compute $\ell_t(\boldsymbol{\mu}_t) = [\theta_t - y_t \mathbf{x}_t^\top \boldsymbol{\mu}_t]_+$;
    if $\ell_t^{CS} > 0$ then
      $\boldsymbol{\Sigma}_{t+1} = \boldsymbol{\Sigma}_t - \frac{\boldsymbol{\Sigma}_t \mathbf{x}_t \mathbf{x}_t^\top \boldsymbol{\Sigma}_t}{\gamma + \mathbf{x}_t^\top \boldsymbol{\Sigma}_t \mathbf{x}_t}$, $\boldsymbol{\mu}_{t+1} = \boldsymbol{\mu}_t - \eta \boldsymbol{\Sigma}_{t+1} \mathbf{g}_t$, or
      $\boldsymbol{\Sigma}_{t+1} = \boldsymbol{\Sigma}_t - \frac{\boldsymbol{\Sigma}_t \odot \mathbf{x}_t \odot \mathbf{x}_t \odot \boldsymbol{\Sigma}_t}{\gamma + (\mathbf{x}_t \odot \boldsymbol{\Sigma}_t)^\top \mathbf{x}_t}$, $\boldsymbol{\mu}_{t+1} = \boldsymbol{\mu}_t - \eta \boldsymbol{\Sigma}_{t+1} \odot \mathbf{g}_t$;
    end if
  end if
end for
```

5 Experiments

5.1 Compared Algorithms and Experimental Testbed

To evaluate the proposed algorithms, we compare it with several state-of-the-art algorithms, which are listed as follows:

- APE: the Active PErceptron algorithm [23];
- APAII: the state-of-the-art first-order Active Passive-Aggressive algorithm [7];
- ASOP": the state-of-the-art Second-Order Active Perceptron algorithm [22];
- SOL": the passive version of SOAL algorithm which queries all of the instances;
- SORL": the random version of SOAL algorithm with random query strategy;
- SOAL-M": the margin-based SOAL algorithm which adopts the same query strategy as in APE, APAII and ASOP;
- SOAL": our proposed Second-order Online Active Learning in Algorithm 1.

To examine the performance of proposed algorithm, we conduct extensive experiments on a variety of benchmark datasets from machine learning repositories. Table 1 shows the details of datasets used in the following experiments. All of these datasets can be freely downloaded from LIBSVM website [1] and UCI machine learning repository [2].

All the compared algorithms learn a linear classifier for the binary classification tasks (The multi-class datasets are changed into binary datasets with one-vs-all strategy). The parameters of each algorithm are searched from $10^{[-5:5]}$ through cross validation for all datasets. The smoothing parameter (determining the query

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

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

TABLE 1: Summary of datasets in the experiments.

Dataset	*#Instances*	*#Features*
a8a	32,561	123
covtype	116,405	54
HIGGS	11,000,000	28
kddcup99	494,012	41
letter	20,000	16
magic04	19,002	10
optdigits	5,620	64
satimage	6,435	36
w8a	64,700	300

ratio) δ is set as $2^{[-10:10]}$ in order to examine varied querying ratios. All the experiments are conducted over 20 runs of different random permutations for each dataset. All the results are reported by averaging over these 20 runs. The algorithms are evaluated with three metrics, accuracy, parameter sensitivity and time complexity.

All of the algorithms are implemented with C++ language, and all of following experiments are conducted in an Ubuntu OS 64-bit PC with Intel Core i7-4770 CPU @ 3.40GHz × 8 and 16 GB memory.

5.2 Evaluation of Varied Query Ratio

In this experiment, we investigate the performance of proposed algorithm SOAL with varied query ratio by setting the parameter δ to different value. Fig. 1 summarizes the average performance on different datasets in terms of accuracy. Based on the results, we can make several observations.

First, in general, second-order based algorithms (SOAL-M and SOAL) can outperform the first-order based algorithms (APE and APAII). This is consistent with the results found in [32], [33] and confirms the necessity of considering second-order information, such as the co-variance matrix, to improve the predictive performance. Second-order based Active Perceptron (ASOP) algorithm usually performs better than the first-order based Active Perceptron (APE) algorithm, which is consistent with the finding in [22]. However, on half of the cases, ASOP algorithm even performs worse than the first-order algorithm APAII, one possible reason is that ASOP is more sensitive to noise.

Second, both the proposed algorithm SOAL and its variant SOAL-M algorithm can consistently achieve better performance than the random query strategy algorithm SORL. This observation indicates that both the margin-based query strategy in SOAL-M and our proposed query strategy in SOAL are effective in identifying more informative instances to label thus can greatly reduce the cost in labeling. This also indicates that the random query strategy can not effectively identify the informative instances to train the model.

Third, compared to the margin-based query strategy in SOAL-M, our proposed strategy in SOAL can consistently achieve the highest accuracy with varied query ratio on all of the datasets. The reason is that we not only consider the margin value of the instance, but also consider the confidence of model. This makes SOAL can identify the instances on which the model has low uncertainty on its predication and low confidence on the learned classifier, such as in the early rounds of online learning. Besides, we observe that the SOAL can achieve comparable performance as SOL by querying less than 20% of the instances. It should be noted that SOL is a fully-supervised online learning algorithm which uses 100% of the query ratio, to make the algorithm clear, we draw a straight line in the Fig. 1.

Last, on some datasets, for example, *HIGGS* and *kddcup99* the active learning algorithms SOAL even can outperform the fully-supervised algorithm SOL. We guess that these datasets might contain many noisy labels. It also should be noted that ASOP algorithm performs worse when the query ratio increases on some datasets, such as *a8a*, *HIGGS* and *magic04*. One possible reason is that ASOP algorithm may suffer the overfitting issue on these datasets.

5.3 Evaluation of Parameter Sensitivity

In the previous section, the parameters η and γ in SOAL are searched from $10^{[-5:5]}$ via cross validation. In this section, we evaluate the sensitivity of algorithms to these parameters.

Fig. 2 shows the experiment results on *a8a*, *covtype* and *HIGGS* datasets. For each dataset, x-axis and y-axis correspond to parameters η and γ, respectively, and different colour corresponds to different performance in terms of accuracy. From the figure, we can observe that parameter η should be neither too small or too large when γ is fixed. This is consistent with our theoretical analysis in Theorem 1. When η is too small, the second term in Theorem 1 will become the dominant term and thus the performance decreases. When η is too large, the third term in Theorem 1 will become dominant and thus make the performance worse. Typically, η should be searched around 1.

In Fig. 2, we can also observe that parameter γ should be decreased when η increases in order to achieve a high accuracy (yellow colour). This observation is also consistent with the theoretical analysis in Theorem 1. When keeping the other terms fixed, we can roughly get $\gamma \sim \frac{1}{\eta^2}$ relationship between γ and η.

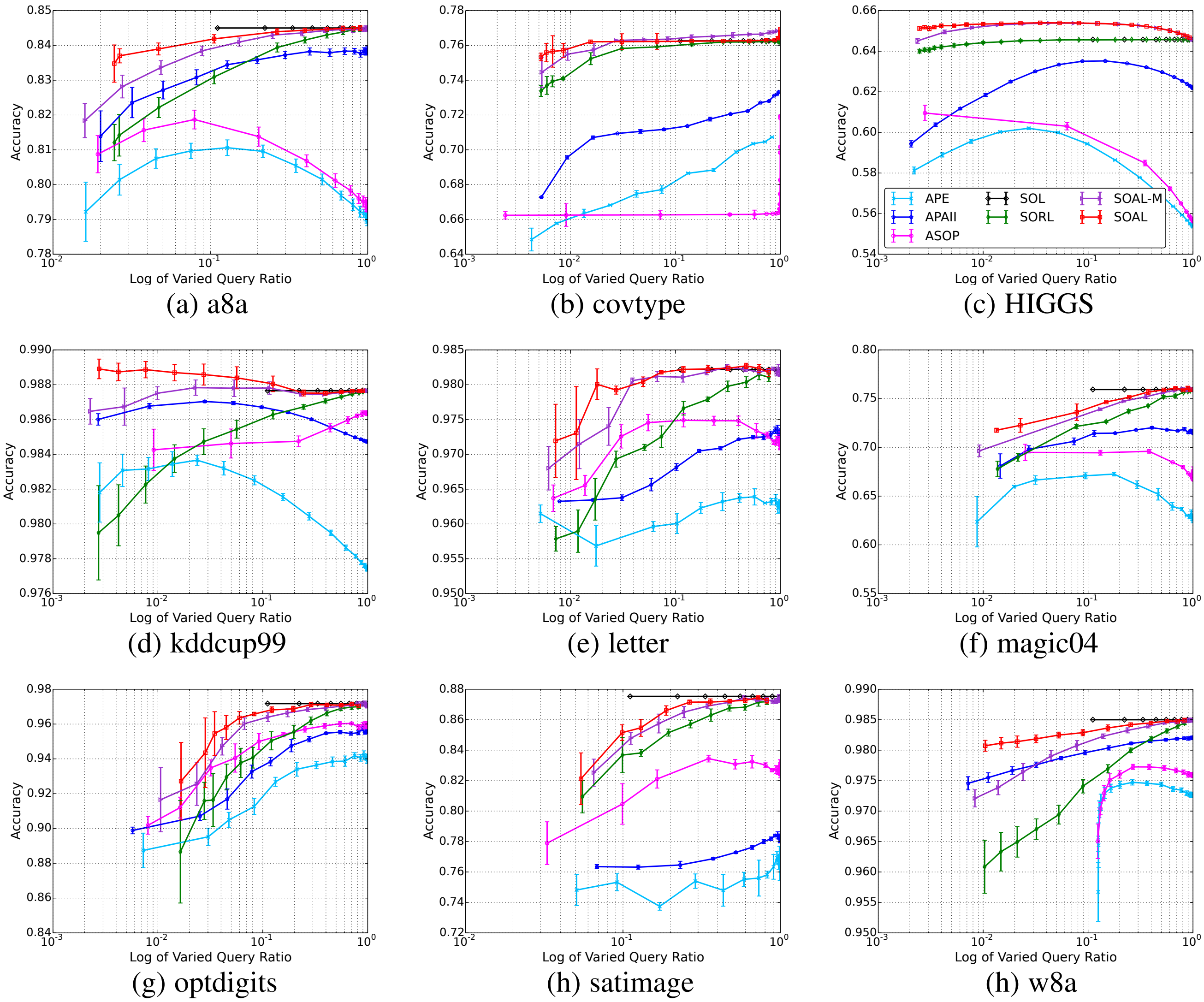
In practice, we can either adopt a grid search for both the η and γ or find the best η first followed by searching best γ around $\frac{1}{\eta^2}$.

5.4 Evaluation of Scalability and Efficiency

Time complexity is usually a major concern for large-scale problems. To evaluate the scalability of the proposed algorithm SOAL, we conducted this experiment to show the time cost corresponding to the log of varied query ratio on three datasets in Fig. 3. Similar observations also could be made on the other datasets.

First, as expected, the first-order based algorithms APE and APAII are the most efficient ones among all algorithms, which only cost less than 0.5 seconds when being trained on all of the instances. This confirms that the first-order online learning scheme is efficient and easy to be scalable to large scale applications. And we also observe that the second-order based algorithms (ASOP, SOL, SORL, SOAL-M and SOAL) typically cost more time due to the computation of the second-order information $\mathbf{\Sigma}$. Among them, AOSP usually requires more time, which is almost two times of the other second-order algorithms (SOL, SORL, SOAL-M and SOAL). Moreover, the proposed algorithm SOAL costs more time than its random variant SORL and the margin-based SORL-M algorithms due to the computation of the query strategy shown in Equation 8.

Second, compared to the passive version SOL, the time complexity of both the random algorithm (SORL) and active algorithms (SOAL-M and SOAL) is smaller when query ratio is less than 100%. The reason is that we will skip updating the model if the label of an instance is not queried. When query ratio increases, the time cost of these algorithms slowly converges to the SOL as expected. This indicates that the proposed algorithm

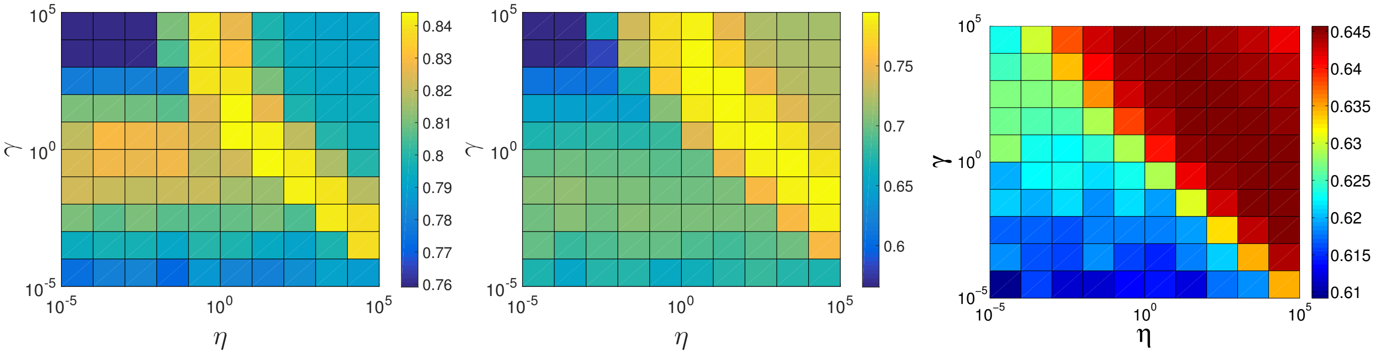


(a) a8a (b) covtype (c) HIGGS

(d) kddcup99 (e) letter (f) magic04

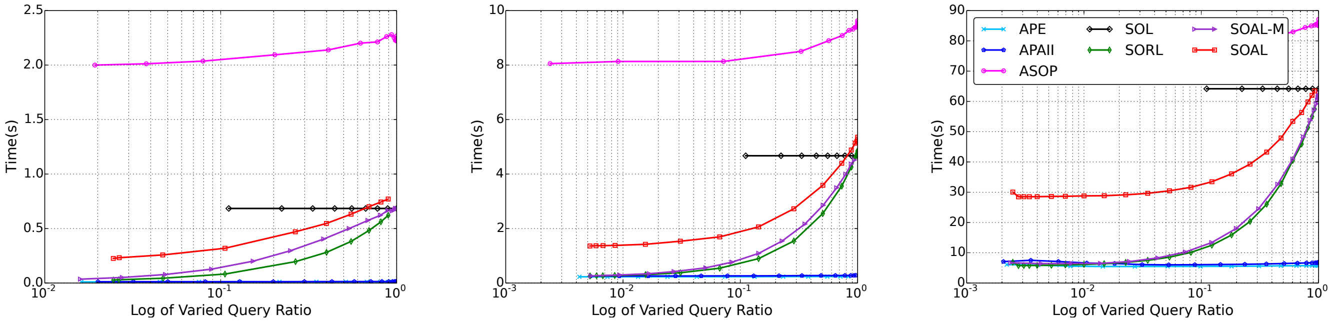
(g) optdigits (h) satimage (h) w8a

Fig. 1: Evaluation of accuracy with respect to log of varied query ratio.



(a) a8a (b) covtype (b) HIGGS

Fig. 2: Evaluation of accuracy against the parameters η and γ in Algorithm 1.



(a) a8a (b) covtype (b) HIGGS

Fig. 3: Evaluation of time cost (seconds) with respect to log of varied query ratio.

SOAL can not only reduce the labeling cost shown in Fig. 1, but also speed up the training process by updating the model only with the queried instance.

Third, when query ratio is around 100%, the time cost of SOAL exceeds the one of SOL as SOAL needs extra time to compute the query strategy. However, the extra time cost could be almost ignored considering the high efficiency of the online learning scheme.

5.5 Application on Malicious Web Classification

In this section, we evaluate the proposed Cost-sensitive Second-order Online Active Learning algorithm (SOAL$_{CS}$) shown in Algorithm 2. To examine its performance, we conduct experiments on two large-scale benchmark datasets for malicious detection problem as follows:

1) "URL" [51]: the task in URL dataset is to classify the malicious URLs from the normal ones. The features in each URL are composed by two parts: (a) Lexical features, the textual properties of the URL itself (not the content of the page), such as length of the hostname, the length of the entire URL, the number of the dots in the URL and so on; (b) Host-based features, such as the IP address properties, WHOIS properties, domain name properties, Geographic properties and so on. In the end, each URL is described by 3231961 features.
2) "webspam" [52]: this dataset is taken from a subset of the one used in Pascal Large Scale Learning Challenge. The web spam pages are obtained by extract the URLs from the email spam corpora SpamArchive [3]. The normal web pages are extracted by traversing the well-known websites, such as news, sports and so on. For each instance, we treat continuous 1 bytes as a word, and use word count as the feature value. In the end, we obtain 254 features for each website.

It should be noted that the original URL and webspam datasets were created in purpose to make them somehow balanced, in which the number of malicious samples is roughly similar to the number of normal ones. In the following experiments, we sample two subsets of these two datasets in order to make them more realistic, in which we randomly sample instances to make sure that the ration between the number of positive instances and the number of negative instances equals to the number shown in the table. Table 2 shows the details of these two subsets, in which T_p/T_n denotes the ratio of number of malicious samples over the number of normal ones.

TABLE 2: Summary of the datasets

$Dataset$	$\#Instances$	$\#Features$	T_p/T_n
URL	1,620,187	3,231,961	1:99
webspam	140,000	254	1:63

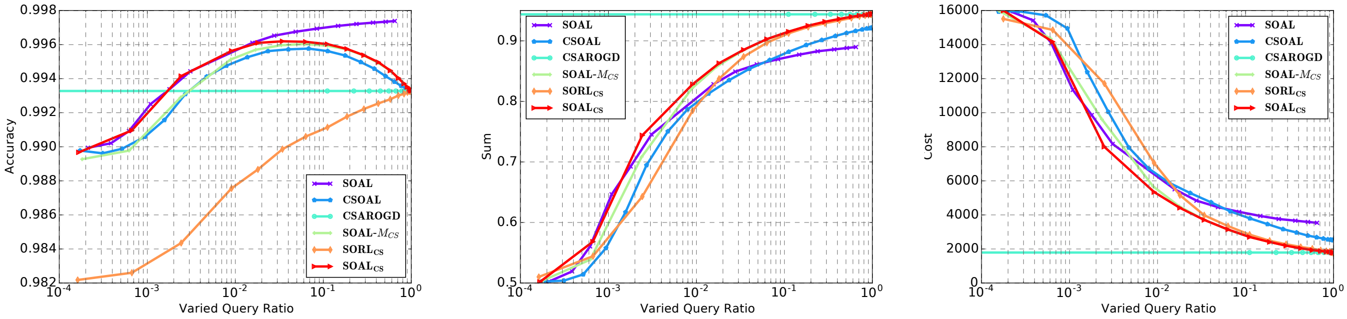
Based on previous evaluation, we know that the proposed algorithm SOAL can consistently achieve the best performance in terms of accuracy, thus here we only consider the SOAL among the algorithms which adopt the accuracy as evaluation metric. Besides, we also consider the following state-of-the-art cost-sensitive algorithms:

- CSOAL [2]: the state-of-the-art first-order based Cost-sensitive Online Active Learning algorithm, which adopts the margin-based query strategy [7], [22] to decide when to query the instance;
- ARCSOGD [48]: the state-of-the-art second-order based Cost-sensitive fully-supervised Online Learning algorithm, which queries all of the instances for labels;
- SOAL-M_{CS}: a variant of SOAL$_{CS}$ algorithm which adopts the margin-based query strategy [7], [22] and the cost-sensitive loss function defined in Eq. (12);
- SORL$_{CS}$: a variant of SOAL$_{CS}$ algorithm which adopts the random query strategy;
- SOAL$_{CS}$: our proposed Cost-sensitive Second-order based Online Active Learning method shown in Algorithm 2, which adopts the query strategy shown in Section 3.2.2 to decide when to query the instance and the cost-sensitive loss defined in Eq. 12.

To make a fair comparison, all algorithms adopt the same experimental setup. For the evaluation metric *sum*, we set $\alpha_p = \alpha_n = 0.5$ for all cost-sensitive algorithms, while for *cost*, we set $c_p = 0.9$ and $c_n = 0.1$. The parameter C in CSOAL, parameters η and γ in $ARCSOGD$, SOAL$_{CS}$, SOAL-M_{CS}, SORL$_{CS}$ and SOAL are selected by cross validation from $[10^{-5}, 10^{-4}, \dots, 10^{5}]$ for each dataset. The smoothing parameter δ in CSOAL, SOAL-M_{CS} and SOAL$_{CS}$ is set as $2^{[-10:2:10]}$ in order to achieve varied querying ratio.

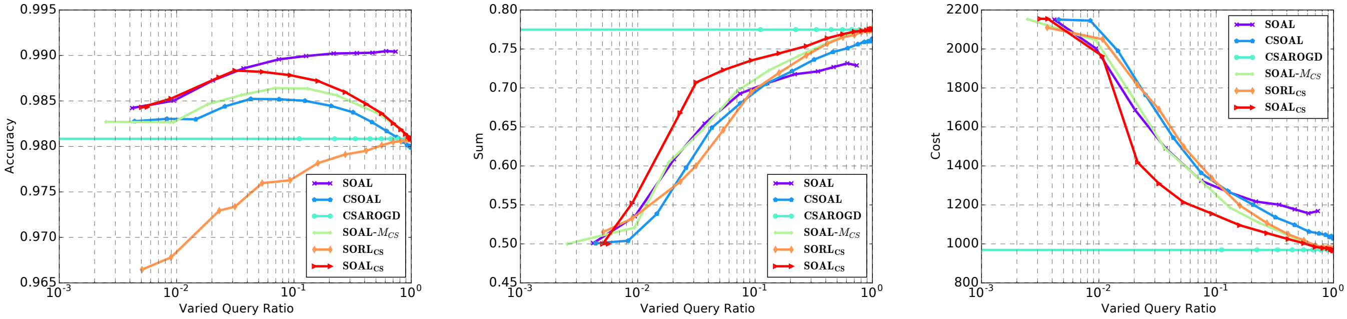
All the experiments are conducted over 10 random permutations on each dataset. The results are reported by averaging over these 10 runs. We evaluate the online classification performance by three metrics: the accuracy which treats the positives and the

[3] ftp://spamarchiev.org/pub/archieves



(a) Accuracy (b) Sum (b) Cost

Fig. 4: Evaluation of Accuracy, Sum, Cost against the varied query ratios on *URL* dataset.



(a) Accuracy (b) Sum (b) Cost

Fig. 5: Evaluation of Accuracy, Sum, Cost against the varied query ratios on *webspam* dataset.

negatives equally, the weighted *sum* defined in Eq. 9 and the *cost* defined in Eq. 10.

By varying the δ value, we can achieve the performance of algorithm with different query ration. Fig. 4 and Fig. 5 show the results on datasets *URL* and *webspam*, respectively. As the feature dimension of "URL" is too high to run the experiments in a single PC, so second-order algorithms shown in Fig. 4 are their diagonal versions with updating rule as describe in Eq. (15) and Eq. (16). Based on the results, we can make several observations.

First, we observe that when we aim to achieve best *accuracy* shown in Fig. 4 (a) and Fig. 5 (a), the proposed algorithm SOAL which adopts regular hinge loss can achieve the best performance. When the query ratio is small, the proposed cost-sensitive algorithm SOAL$_{CS}$ can roughly achieve similar accuracy with the SOAL algorithm. However, when the query ratio increases, the performance of SOAL algorithm consistently increases and achieves the best performance lastly. This indicates that the accuracy in the imbalanced dataset could be a misleading evaluation metric. Thus, this motivates us to investigate the other proper metrics, such as *sum* and *cost*. It should also be noted that the proposed SOAL$_{CS}$ algorithm can consistently outperform the other cost-sensitive algorithms even with accuracy as evaluation metric, which again validates the query strategy proposed in Section 3.2.2 is effective in querying informative instances.

Second, when we adopt the evaluation metric *sum* of sensitivity and specificity shown in Fig. 4 (b) and Fig. 5 (b), we observe that cost-sensitive algorithm SOAL$_{CS}$, SORL$_{CS}$ can achieve better performance than SOAL when the query ratio is larger than 1%. It should also be noted that first-order based cost-sensitive algorithm CSOAL also can outperform SOAL when query ratio is larger than 10%. These observations indicate that it's necessary to import cost-sensitive strategy which put a larger weight on the malicious samples. Furthermore, it can be observed that the second-order cost-sensitive algorithms SOAL$_{CS}$, SORL$_{CS}$ and ARCSOGD can outperform the first-order algorithm CSOAL. This indicates its effectiveness to consider the second-order information when designing learning models. We also observe that the proposed algorithm SOAL$_{CS}$ can consistently outperform the margin-based algorithm SOAL-M_{CS} and the random version SORL$_{CS}$. This again verifies the effectiveness of the proposed active learning strategy.

Third, similar observations with metric *cost* can be made in Fig.4 (c) and Fig. 5 (c) as the metric *sum*. Besides, it should be noted that the fully-supervised cost-sensitive algorithm ARCSOGD can achieve the best performance in terms of both *sum* and *cost*, however, ARCSOGD requires to query 100% of the instances, which is costly and time consuming. For example, in the *URL* dataset, we have more than 1.5 million instances and labeling all of these instances would be extreme costly. Our proposed active learning algorithm SOAL$_{CS}$ can achieve comparable performance as ARCSOGD with less than 50% query ratio.

6 Conclusion

This paper proposed SOAL — a framework of Second-order Online Active Learning in order to address the open challenge of real-life online learning from unlabeled data streams given limited label query budget. By adopting an effective second-order online learning framework, we proposed to build an effective label query strategy by carefully considering not only the prediction margin of an incoming unlabeled instance but also the confidence of the learner. To further tackle the cost-sensitive learning problems for class-imbalanced applications such as malicious web detection, we extended the SOAL framework by proposing a new cost-sensitive and second-order online active learning algorithm SOAL$_{CS}$ to explicitly optimize the cost-sensitive metrics. We theoretically analyzed the mistake bounds of the proposed SOAL algorithm and conducted a set of extensive experiments to examine its empirical effectiveness in terms of accuracy, parameter sensitivity and scalability. We also successfully applied the proposed SOAL$_{CS}$ algorithm to two large-scale malicious web classification tasks. The experimental results showed that our algorithms consistently outperform several state-of-the-art approaches. Future work will explore some hyper-parameter learning strategy for automatically re-adjusting the parameters γ and η in the online learning process, and active learning in the transfer learning domain [53], multi-task learning [54] etc.

ACKNOWLEDGMENTS

Steven C.H. Hoi and Chunyan Miao are supported by the National Research Foundation, Prime Ministers Office, Singapore under its International Research Centres in Singapore Funding Initiative.

APPENDIX
PROOF OF THEOREM 1

To facilitate the proof, we first present a lemma as follows.

Lemma 1. *Let* $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_T, y_T)$ *be a sequence of input examples, where* $\mathbf{x}_t \in \mathbb{R}^d$ *and* $y_t \in \{-1, +1\}$ *for all t. If the SOAL algorithm is run on this sequence of examples, then the following bound holds for any* $\mathbf{w} \in \mathbb{R}^d$,

$$\begin{aligned}
&Z_t\Big[M_t(\delta+|p_t|)+L_t(\delta-|p_t|)\Big]\\
&\le \frac{1}{2\eta}Z_t\Big[\|\boldsymbol{\mu}_t-\delta\boldsymbol{\mu}\|^2_{\boldsymbol{\Sigma}^{-1}_{t+1}}-\|\boldsymbol{\mu}_{t+1}-\delta\boldsymbol{\mu}\|^2_{\boldsymbol{\Sigma}^{-1}_{t+1}}\\
&\qquad +\|\boldsymbol{\mu}_t-\boldsymbol{\mu}_{t+1}\|^2_{\boldsymbol{\Sigma}^{-1}_{t+1}}+\delta\ell_t(\boldsymbol{\mu})\Big],
\end{aligned}$$

where $\delta > 0$.

Proof. When $Z_t = 0$, it is easy to verify the inequality in the theorem, using the fact $\ell_t \ge 0$.

When $Z_t = 1$, it is easy to observe that

$$\begin{aligned}
\boldsymbol{\mu}_{t+1} &= \arg\min_{\boldsymbol{\mu}} h_t(\boldsymbol{\mu})\\
h_t(\boldsymbol{\mu}) &= \frac{1}{2}\|\boldsymbol{\mu}_t-\boldsymbol{\mu}\|^2_{\boldsymbol{\Sigma}^{-1}_{t+1}}+\eta\mathbf{g}_t^\top\boldsymbol{\mu}.
\end{aligned}$$

Because h_t is convex, we have the following inequality $\forall\boldsymbol{\mu}$,

$$\begin{aligned}
0 &\le \partial h_t(\boldsymbol{\mu}_{t+1})^\top(\boldsymbol{\mu}-\boldsymbol{\mu}_{t+1})\\
&= \big[\boldsymbol{\Sigma}^{-1}_{t+1}(\boldsymbol{\mu}_{t+1}-\boldsymbol{\mu}_t)+\eta\mathbf{g}_t\big]^\top(\boldsymbol{\mu}-\boldsymbol{\mu}_{t+1}).
\end{aligned}$$

Re-arranging the above inequality will result in

$$\begin{aligned}
&\eta\mathbf{g}_t^\top(\boldsymbol{\mu}_{t+1}-\boldsymbol{\mu})\\
&\le \Big(\boldsymbol{\mu}_{t+1}-\boldsymbol{\mu}_t\Big)^\top\boldsymbol{\Sigma}^{-1}_{t+1}(\boldsymbol{\mu}-\boldsymbol{\mu}_{t+1}\Big)\\
&= \frac{1}{2}\Big[\|\boldsymbol{\mu}_t-\boldsymbol{\mu}\|^2_{\boldsymbol{\Sigma}^{-1}_{t+1}}-\|\boldsymbol{\mu}_{t+1}-\boldsymbol{\mu}\|^2_{\boldsymbol{\Sigma}^{-1}_{t+1}}\Big]\\
&\quad -\frac{1}{2}\Big[\|\boldsymbol{\mu}_t-\boldsymbol{\mu}_{t+1}\|^2_{\boldsymbol{\Sigma}^{-1}_{t+1}}\Big]
\end{aligned} \tag{17}$$

Now, we would provide a lower bound for $\mathbf{g}_t^\top(\boldsymbol{\mu}_{t+1}-\boldsymbol{\mu})$,

$$\begin{aligned}
&\mathbf{g}_t^\top(\boldsymbol{\mu}_{t+1}-\boldsymbol{\mu})\\
&= \mathbf{g}_t^\top(\boldsymbol{\mu}_t-\boldsymbol{\mu})+\mathbf{g}_t^\top(\boldsymbol{\mu}_{t+1}-\boldsymbol{\mu}_t)\\
&= (L_t+M_t)(-y_t\mathbf{x}_t^\top\boldsymbol{\mu}_t)+(L_t+M_t)y_t\mathbf{x}_t^\top\boldsymbol{\mu}\\
&\quad -\frac{1}{\eta}\|\boldsymbol{\mu}_{t+1}-\boldsymbol{\mu}_t\|^2_{\boldsymbol{\Sigma}^{-1}_{t+1}},
\end{aligned} \tag{18}$$

where the second equality used the facts $\mathbf{g}_t = (L_t+M_t)(-y_t\mathbf{x}_t)$ and $\partial h_t(\boldsymbol{\mu}_{t+1}) = 0$ i.e.,

$$\boldsymbol{\Sigma}^{-1}_{t+1}(\boldsymbol{\mu}_{t+1}-\boldsymbol{\mu}_t)+\eta\mathbf{g}_t = 0. \tag{19}$$

Combining the above equality (18) with the facts

$$\begin{aligned}
M_t(-y_t\mathbf{x}_t^\top\boldsymbol{\mu}_t) &= M_t|y_t\mathbf{x}_t^\top\boldsymbol{\mu}_t| = M_t|p_t|\\
L_t(-y_t\mathbf{x}_t^\top\boldsymbol{\mu}_t) &= L_t(-|y_t\mathbf{x}_t^\top\boldsymbol{\mu}_t|) = -L_t|p_t|\\
y_t\mathbf{x}_t^\top\boldsymbol{\mu}+\delta\ell_t(\boldsymbol{\mu}/\delta) &\ge y_t\mathbf{x}_t^\top\boldsymbol{\mu}+\delta(1-y_t\mathbf{x}_t^\top\boldsymbol{\mu}/\delta) = \delta,
\end{aligned}$$

we get the following bound for $\mathbf{g}_t^\top(\boldsymbol{\mu}_{t+1}-\boldsymbol{\mu})$,

$$\begin{aligned}
&\mathbf{g}_t^\top(\boldsymbol{\mu}_{t+1}-\boldsymbol{\mu})\\
&\ge (M_t|p_t|-L_t|p_t|)+(L_t+M_t)[\delta-\delta\ell_t(\boldsymbol{\mu}/\delta)]\\
&\quad -\frac{1}{\eta}\|\boldsymbol{\mu}_{t+1}-\boldsymbol{\mu}_t\|^2_{\boldsymbol{\Sigma}^{-1}_{t+1}}\\
&= [M_t(\delta+|p_t|)+L_t(\delta-|p_t|)]-\frac{1}{\eta}\|\boldsymbol{\mu}_{t+1}-\boldsymbol{\mu}_t\|^2_{\boldsymbol{\Sigma}^{-1}_{t+1}}\\
&\quad -(L_t+M_t)\delta\ell_t(\boldsymbol{\mu}/\delta).
\end{aligned} \tag{20}$$

Combining the above two inequalities (17) and (20), will give the following important inequality

$$\begin{aligned}
&[M_t(\delta+|p_t|)+L_t(\delta-|p_t|)]\\
&\le \frac{1}{2\eta}\Big[\|\boldsymbol{\mu}_t-\boldsymbol{\mu}\|^2_{\boldsymbol{\Sigma}^{-1}_{t+1}}-\|\boldsymbol{\mu}_{t+1}-\boldsymbol{\mu}\|^2_{\boldsymbol{\Sigma}^{-1}_{t+1}}-\|\boldsymbol{\mu}_t-\boldsymbol{\mu}_{t+1}\|^2_{\boldsymbol{\Sigma}^{-1}_{t+1}}\Big]\\
&\quad +\frac{1}{\eta}\|\boldsymbol{\mu}_{t+1}-\boldsymbol{\mu}_t\|^2_{\boldsymbol{\Sigma}^{-1}_{t+1}}+\delta\ell_t(\boldsymbol{\mu}/\delta)\\
&= \frac{1}{2\eta}\Big[\|\boldsymbol{\mu}_t-\boldsymbol{\mu}\|^2_{\boldsymbol{\Sigma}^{-1}_{t+1}}-\|\boldsymbol{\mu}_{t+1}-\boldsymbol{\mu}\|^2_{\boldsymbol{\Sigma}^{-1}_{t+1}}+\|\boldsymbol{\mu}_t-\boldsymbol{\mu}_{t+1}\|^2_{\boldsymbol{\Sigma}^{-1}_{t+1}}\Big]\\
&\quad +\delta\ell_t(\boldsymbol{\mu}/\delta)
\end{aligned}$$

Replacing $\boldsymbol{\mu}$ with $\delta\boldsymbol{\mu}$ concludes the proof. □

We now proof the proposed Theorem 1 as follows.

Proof. Firstly, according to the equality (19), we have

$$\begin{aligned}
&\|\boldsymbol{\mu}_t-\boldsymbol{\mu}_{t+1}\|^2_{\boldsymbol{\Sigma}^{-1}_{t+1}}\\
&= \eta^2\mathbf{g}_t^\top\boldsymbol{\Sigma}_{t+1}\mathbf{g}_t\\
&= \eta^2(M_t+L_t)\mathbf{x}_t^\top\boldsymbol{\Sigma}_{t+1}\mathbf{x}_t\\
&= \eta^2(M_t+L_t)\Big(\mathbf{x}_t^\top\boldsymbol{\Sigma}_t\mathbf{x}_t-\frac{\mathbf{x}_t^\top\boldsymbol{\Sigma}_t\mathbf{x}_t\mathbf{x}_t^\top\boldsymbol{\Sigma}_t\mathbf{x}_t}{\gamma+\mathbf{x}_t^\top\boldsymbol{\Sigma}_t\mathbf{x}_t}\Big)\\
&= \eta^2(M_t+L_t)\frac{\gamma v_t}{\gamma+v_t},
\end{aligned}$$

where we used the updating rule of $\boldsymbol{\Sigma}$. Plugging it into the inequality in the Lemma 1 and re-arranging it will give

$$Z_t\Bigg[M_t\left(\delta+|p_t|-\frac{\eta\gamma v_t}{2(\gamma+v_t)}\right)+L_t\left(\delta-|p_t|-\frac{\eta\gamma v_t}{2(\gamma+v_t)}\right)\Bigg]$$
$$\le\frac{1}{2\eta}Z_t\left[\|\boldsymbol{\mu}_t-\delta\boldsymbol{\mu}\|^2_{\boldsymbol{\Sigma}^{-1}_{t+1}}-\|\boldsymbol{\mu}_{t+1}-\delta\boldsymbol{\mu}\|^2_{\boldsymbol{\Sigma}^{-1}_{t+1}}\right]+\delta\ell_t(\boldsymbol{\mu})$$

Summing the above inequality over $t=1,2,\ldots,T$ and using the definition of ρ_t can give

$$\begin{aligned}&\sum_{t=1}^{T}Z_t\Big[M_t(\delta+\rho_t)+L_t(\delta+\rho_t-2|p_t|)\Big]\\&\le\frac{1}{2\eta}\sum_{t=1}^{T}Z_t\Big[\|\boldsymbol{\mu}_t-\delta\boldsymbol{\mu}\|^2_{\boldsymbol{\Sigma}^{-1}_{t+1}}-\|\boldsymbol{\mu}_{t+1}-\delta\boldsymbol{\mu}\|^2_{\boldsymbol{\Sigma}^{-1}_{t+1}}\Big]\\&\quad+\delta\sum_{t=1}^{T}Z_t\ell_t(\boldsymbol{\mu})\end{aligned}\tag{21}$$

Now, we would like to bound the right-hand side of the above inequality. Firstly, we bound the first term as

$$\begin{aligned}&\sum_{t=1}^{T}Z_t\left[\|\boldsymbol{\mu}_t-\delta\boldsymbol{\mu}\|^2_{\boldsymbol{\Sigma}^{-1}_{t+1}}-\|\boldsymbol{\mu}_{t+1}-\delta\boldsymbol{\mu}\|^2_{\boldsymbol{\Sigma}^{-1}_{t+1}}\right]\\&\le\|\boldsymbol{\mu}_1-\delta\boldsymbol{\mu}\|^2_{\boldsymbol{\Sigma}_2^{-1}}+\sum_{t=2}^{T}\left[\|\boldsymbol{\mu}_t-\delta\boldsymbol{\mu}\|^2_{\boldsymbol{\Sigma}^{-1}_{t+1}}-\|\boldsymbol{\mu}_t-\delta\boldsymbol{\mu}\|^2_{\boldsymbol{\Sigma}_t^{-1}}\right]\\&\le\|\boldsymbol{\mu}_1-\delta\boldsymbol{\mu}\|^2\mathrm{Tr}(\boldsymbol{\Sigma}_2^{-1})+\sum_{t=2}^{T}\|\boldsymbol{\mu}_t-\delta\boldsymbol{\mu}\|^2\mathrm{Tr}(\boldsymbol{\Sigma}^{-1}_{t+1}-\boldsymbol{\Sigma}_t^{-1})\\&=\max_{t\le T}\|\boldsymbol{\mu}_t-\delta\boldsymbol{\mu}\|^2\mathrm{Tr}\left(\boldsymbol{\Sigma}^{-1}_{T+1}\right)\\&\le2(D_{\boldsymbol{\mu}}+(1-\delta)^2\|\boldsymbol{\mu}\|^2)\mathrm{Tr}(\boldsymbol{\Sigma}^{-1}_{T+1})\end{aligned}\tag{22}$$

where $D_{\boldsymbol{\mu}}=\max_{t\le T}\|\boldsymbol{\mu}_t-\boldsymbol{\mu}\|^2$. Plugging the above upper bound (22) into the inequality (21), we can get

$$\begin{aligned}&\sum_{t=1}^{T}Z_t\left[M_t(\delta+\rho_t)+L_t(\delta+\rho_t-2|p_t|)\right]\\&\le\frac{1}{\eta}\left(D_{\boldsymbol{\mu}}+(1-\delta)^2\|\boldsymbol{\mu}\|^2\right)\mathrm{Tr}\left(\boldsymbol{\Sigma}^{-1}_{T+1}\right)\\&\quad+\delta\sum_{t=1}^{T}Z_t\ell_t(\boldsymbol{\mu})\end{aligned}\tag{23}$$

When $\rho_t>0$, using $\mathbb{E}_tZ_t=\delta/(\delta+\rho_t)$, we have

$$\begin{aligned}&\mathbb{E}\Big\{Z_t\left[M_t(\delta+\rho_t)+L_t(\delta+\rho_t-2|p_t|)\right]\Big\}\\&=\delta\mathbb{E}[M_t]+\delta\mathbb{E}\left[L_t\left(1-2|p_t|/(\delta+\rho_t)\right)\right]\\&\ge\delta\mathbb{E}[M_t]+(\delta-2)\mathbb{E}[L_t].\end{aligned}$$

When $\rho_t\le0$, i.e.,$|p_t|\le\frac{\eta\gamma v_t}{2(\gamma+v_t)}$, using $\mathbb{E}_tZ_t=1$, we have

$$\begin{aligned}&\mathbb{E}\Big\{Z_t[M_t(\delta+\rho_t)+L_t(\delta+\rho_t-2|p_t|)]\Big\}\\&\ge\mathbb{E}M_t\left(\delta-\frac{\eta\gamma v_t}{2(\gamma+v_t)}\right)+\mathbb{E}L_t\left(\delta-\frac{\eta\gamma v_t}{\gamma+v_t}\right)\end{aligned}$$

To summarize,

$$\begin{aligned}&\sum_{t=1}^{T}\mathbb{E}\Big\{Z_t\left[M_t(\delta+\rho_t)+L_t(\delta+\rho_t-2|p_t|)\right]\Big\}\\&\ge\delta\mathbb{E}\left[\sum_{t=1}^{T}M_t\right]+\delta\mathbb{E}\left[\sum_{t=1}^{T}L_t\right]-2\mathbb{E}\left[\sum_{\rho_t>0}L_t\right]\\&\quad-\mathbb{E}\left[\sum_{\rho_t<0}M_t\frac{\eta\gamma v_t}{2(\gamma+v_t)}\right]-\mathbb{E}\left[\sum_{\rho_t<0}L_t\frac{\eta\gamma v_t}{\gamma+v_t}\right]\end{aligned}$$

Taking expectation of the inequality (23) and combining with the above inequality conclude the proof. □

References

[1] T. Yang, M. Mahdavi, R. Jin, and S. Zhu, "Regret bounded by gradual variation for online convex optimization," *Machine Learning*, Oct. 2013.

[2] P. Zhao and S. C. Hoi, "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 - KDD '13*, (New York, New York, USA), p. 919, ACM Press, 2013.

[3] H. B. Ammar, U. EDU, E. Eaton, P. Ruvolo, O. EDU, M. E. Taylor, and W. EDU, "Online multi-task learning for policy gradient methods," *ICML 2014*, 2014.

[4] C. Zhang, P. Zhao, S. Hao, Y. C. Soh, and B. S. Lee, "Rom: A robust online multi-task learning approach," in *2016 IEEE 16th International Conference on Data Mining (ICDM'16),*, pp. 1341–1346, IEEE, 2016.

[5] C. C. Cao, L. Chen, and H. V. Jagadish, "From labor to trader: Opinion elicitation via online crowds as a market," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, (New York, NY, USA), pp. 1067–1076, ACM, 2014.

[6] G. Li, S. C. Hoi, K. Chang, W. Liu, and R. Jain, "Collaborative online multitask learning," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 26, no. 8, pp. 1866–1876, 2014.

[7] J. Lu, Z. Peilin, and C. S. Hoi, "Online passive aggressive active learning and its applications," *Machine Learning*, vol. 103(2), pp. 141–183, 2016.

[8] D. Sahoo, S. C. Hoi, and B. Li, "Online multiple kernel regression," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 293–302, ACM, 2014.

[9] S. Hao, P. Zhao, Y. Liu, S. C. H. Hoi, and C. Miao, "Online multi-task relative similarity learning," in *The 26th International Joint Conference on Artificial Intelligence (IJCAI'17)*, 2017.

[10] S. C. Hoi, J. Wang, and P. Zhao, "LIBOL: a library for online learning algorithms," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 495–499, 2014.

[11] P. Ruvolo and E. Eaton, "Online multi-task learning via sparse dictionary optimization," in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada.* (C. E. Brodley and P. Stone, eds.), pp. 2062–2068, AAAI Press, 2014.

[12] M. Aleksandrov, H. Aziz, S. Gaspers, and T. Walsh, "Online fair division: Analysing a food bank problem," in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015* (Q. Yang and M. Wooldridge, eds.), pp. 2540–2546, AAAI Press, 2015.

[13] X. Guo, "Online robust low rank matrix recovery," in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015* (Q. Yang and M. Wooldridge, eds.), pp. 3540–3546, AAAI Press, 2015.

[14] K. Hayakawa, E. H. Gerding, S. Stein, and T. Shiga, "Online mechanisms for charging electric vehicles in settings with varying marginal electricity costs," in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015* (Q. Yang and M. Wooldridge, eds.), pp. 2610–2616, AAAI Press, 2015.

[15] F. Jahedpari, "Artificial prediction markets for online prediction," in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015* (Q. Yang and M. Wooldridge, eds.), pp. 4371–4372, AAAI Press, 2015.

[16] J. Veness, M. Hutter, L. Orseau, and M. G. Bellemare, "Online learning of k-cnf boolean functions," in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015* (Q. Yang and M. Wooldridge, eds.), pp. 3865–3873, AAAI Press, 2015.

[17] J. Wan, P. Wu, S. C. H. Hoi, P. Zhao, X. Gao, D. Wang, Y. Zhang, and J. Li, "Online learning to rank for content-based image retrieval," in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015* (Q. Yang and M. Wooldridge, eds.), pp. 2284–2290, AAAI Press, 2015.

[18] B. Wang and J. Pineau, "Online boosting algorithms for anytime transfer and multitask learning," in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[19] B. Settles, "Active learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 6, pp. 1–114, June 2010.

[20] S. Tong, *Active learning: theory and applications*. PhD thesis, Stanford University, 2001.

[21] S. Hanneke, "Theory of disagreement-based active learning," *Foundations and Trends® in Machine Learning*, vol. 7, no. 2-3, pp. 131–309, 2014.

[22] N. Cesa-Bianchi, C. Gentile, and L. Zaniboni, "Worst-case analysis of selective sampling for linear classification," *The Journal of Machine Learning Research*, vol. 7, pp. 1205–1230, Dec. 2006.

[23] N. Cesa-Bianchi, C. Gentile, and L. Zaniboni, "Worst-case analysis of selective sampling for linear-threshold algorithms," in *Advances in Neural Information Processing Systems 17 [Neural Information Processing Systems, NIPS 2004, December 13-18, 2004, Vancouver, British Columbia, Canada]*, pp. 241–248, 2004.

[24] D. Sahoo, C. Liu, and S. C. Hoi, "Malicious url detection using machine learning: A survey," *arXiv preprint arXiv:1701.07179*, 2017.

[25] S. Hao, P. Zhao, J. Lu, S. C. Hoi, C. Miao, and C. Zhang, "Soal: Second-order online active learning," in *Data Mining (ICDM), 2016 IEEE 16th International Conference on*, pp. 931–936, IEEE, 2016.

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

[27] H. Block, "The perceptron: A model for brain functioning. i," *Reviews of Modern Physics*, vol. 34, no. 1, 1962.

[28] M. Mohri and A. Rostamizadeh, "Perceptron mistake bounds," *arXiv preprint arXiv:1305.0208*, 2013.

[29] Y. Li and P. M. Long, "The relaxed online maximum margin algorithm," *Machine Learning*, vol. 46, no. 1-3, pp. 361–387, 2002.

[30] K. Crammer, O. Dekel, J. Keshet, and S. Shalev-shwartz, "Online passive-aggressive algorithms," *The Journal of Machine Learning*, vol. 7, pp. 551–585, 2006.

[31] N. Cesa-Bianchi, A. Conconi, and C. Gentile, "A second-order perceptron algorithm," *SIAM Journal on Computing*, vol. 34, pp. 640–668, Jan. 2005.

[32] M. Dredze, K. Crammer, and F. Pereira, "Confidence-weighted linear classification," in *Proceedings of the 25th international conference on Machine learning*, pp. 264–271, ACM, 2008.

[33] K. Crammer, A. Kulesza, and M. Dredze, "Adaptive regularization of weight vectors," *Machine Learning*, vol. 91, pp. 155–187, Mar. 2013.

[34] J. Wang, P. Zhao, and S. C. Hoi, "Exact soft confidence-weighted learning," in *Proceedings of the 29th International Conference on Machine Learning (ICML-12)* (J. Langford and J. Pineau, eds.), (New York, NY, USA), pp. 121–128, ACM, 2012.

[35] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *The Journal of Machine Learning Research*, pp. 45–66, 2002.

[36] Y. Freund and Y. Mansour, "Learning under persistent drift," in *Computational Learning Theory*, pp. 109–118, 1997.

[37] S. Hanneke and L. Yang, "Minimax analysis of active learning," *Journal of Machine Learning Research*, vol. 16, pp. 3487–3602, 2015.

[38] S. Hanneke, "The optimal sample complexity of pac learning," *Journal of Machine Learning Research*, vol. 17, no. 38, pp. 1–15, 2016.

[39] Y. Guo and D. Schuurmans, "Discriminative batch mode active learning," in *Advances in neural information processing systems*, pp. 593–600, 2008.

[40] V. S. Sheng, F. Provost, and P. G. Ipeirotis, "Get another label? improving data quality and data mining using multiple, noisy labelers," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 614–622, ACM, 2008.

[41] S. Hanneke, "Activized learning : Transforming passive to active with improved label complexity ∗," *Journal of Machine Learning Research*, vol. 13, pp. 1469–1587, 2012.

[42] K. Fujii and H. Kashima, "Budgeted stream-based active learning via adaptive submodular maximization," in *Advances in Neural Information Processing Systems 29* (D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, eds.), pp. 514–522, Curran Associates, Inc., 2016.

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

[44] R. Akbani, S. Kwek, and N. Japkowicz, "Applying support vector machines to imbalanced datasets," in *European conference on machine learning*, pp. 39–50, Springer, 2004.

[45] M. Tan, "Cost-sensitive learning of classification knowledge and its applications in robotics," *Machine Learning*, vol. 13, no. 1, pp. 7–33, 1993.

[46] A. C. Lozano and N. Abe, "Multi-class cost-sensitive boosting with p-norm loss functions," in *Proceedings of the 14th ACM SIGKDD interna-*

tional conference on Knowledge discovery and data mining, pp. 506–514, ACM, 2008.

[47] Y. Li, J. T.-Y. Kwok, and Z.-H. Zhou, "Cost-sensitive semi-supervised support vector machine," in *Proceedings of the National Conference on Artificial Intelligence*, vol. 1, p. 500, 2010.

[48] P. Zhao, F. Zhuang, M. Wu, X.-L. Li, and S. C. Hoi, "Cost-sensitive online classification with adaptive regularization and its applications," in *Data Mining (ICDM), 2015 IEEE International Conference on*, pp. 649–658, IEEE, 2015.

[49] K. Crammer, A. Kulesza, and M. Dredze, "Adaptive regularization of weight vectors," in *Advances in Neural Information Processing Systems*, pp. 414–422, 2009.

[50] W. Jialei, Z. Peilin, and S. Hoi, "Cost-sensitive online classification," *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, vol. 26, no. 10, pp. 2425–2438, 2015.

[51] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond blacklists: learning to detect malicious web sites from suspicious urls," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1245–1254, ACM, 2009.

[52] S. Webb, J. Caverlee, and C. Pu, "Introducing the webb spam corpus: Using email spam to identify web spam automatically.," in *CEAS*, 2006.

[53] J. Wang, Y. Chen, S. Hao, and W. Feng, "Balanced distribution adaptation for transfer learning," in *ICDM 2017*, 2017.

[54] C. Zhang, P. Zhao, S. Hao, and S. Hoi, "Distributed multi-task classification: A decentralized online learning approach," *Machine Learning*, 2017.

Shuji Hao is a Scientist in the Institute of High Performance Computing (IHPC), Agency for Science Technology and Research (A*STAR), Singapore. Prior to joining IHPC, he finished PhD in Nanyang Technological University (NTU) in 2016. During his PhD study, he has been dedicated to designing large-scale sustainable machine learning algorithms, such as online learning, active learning, and applications such as image search, text classification, and abnormal detection. He currently focused on automatic machine learning research, such as reinforcement learning for automatic hyper-parameter search, and human-in-loop image search and so on. He has published several research papers as the first authors in top tier computer conferences.

Jing LU is a Ph.D student in the School of Information Systems (SIS), Singapore Management University (SMU), Singapore. Prior to joining SMU, she received her Bachelor's Degree in Honor School, Harbin Institute of Technology, China in 2012 and worked as a Project Officer in School of Computer Science, Nanyang Technological University, Singapore 2012-2014. During her past PHD study, she has been dedicated to her research area of online learning for addressing the emerging challenges of big data analytics particularly for dealing with real-time data stream analytics. She has published several research papers as the first authors in top tier journals and high-impact conferences. Most of her research publications addressed the key open challenges in the area of machine learning and big data analytics fields.

Peilin Zhao is currently a Professor in the School of Software Engineering, South China University of Technology, China. Previously, he has worked in the Department of Statistics, Rutgers University, USA, the Institute for Infocomm Research (I2R), A*STAR, Singapore, and Ant Financial Services Group, China. He received his PHD degree from School of Computer Engineering, Nanyang Technological University and his Bachelor degree from Department of Mathematics, Zhejiang University. His research Interests are Large Scale Machine Learning and its applications to Big Data Analytics. In his research areas, he has published over 70 papers in top conferences and journals. In addition, he has served as member or reviewer for many premier international conferences and journals.

Chi Zhang received the B.S. degree in physics from University of Science and Technology, China, 2014, and he is currently pursuing the Ph.D. degree in Interdisciplinary Graduate School at Nanyang Technological University. Since 2014, he has been working on the research of large-scale machine learning and optimization theory. His current research activities lie in the areas of online learning, multi-task learning stochastic optimization and.

Steven C. H. Hoi is currently an Associate Professor of the School of Information Sytems, Singapore Management Unviersity, Singapore. Prior to joining SMU, he was Associate Professor with Nanyang Technological University, Singapore. He received his Bachelor degree from Tsinghua University, P.R. China, in 2002, and his Ph.D degree in computer science and engineering from The Chinese University of Hong Kong, in 2006. His research interests are machine learning and data mining and their applications to multimedia information retrieval (image and video retrieval), social media and web mining, and computational finance, etc, and he has published over 150 refereed papers in top conferences and journals in these related areas. He has served as Editor in Chief for Neurocomputing Journal, general co-chair for ACM SIGMM Workshops on Social Media (WSM'09, WSM'10, WSM'11), program co-chair for the fourth Asian Conference on Machine Learning (ACML'12), book editor for "Social Media Modeling and Computing", guest editor for ACM Transactions on Intelligent Systems and Technology (ACM TIST), technical PC member for many international conferences, and external reviewer for many top journals and worldwide funding agencies, including NSF in US and RGC in Hong Kong.

Chunyan Miao is a Full Professor in the Division of Information Systems and Director of the Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly (LILY), School of Computer Engineering, Nanyang Technological University (NTU), Singapore. She received her PhD degree from NTU and was a Postdoctoral Fellow/Instructor in the School of Computing, Simon Fraser University (SFU), Canada. She visited Harvard and MIT, USA, as a Tan Chin Tuan Fellow, collaborating on a large NSF funded research program in social networks and virtual worlds. She has been an Adjunct Associate Professor/Associate Professor/Founding Faculty member with the Center for Digital Media which is jointly managed by The University of British Columbia (UBC) and SFU. Her current research is focused on human-centered computational/ artificial intelligence and interactive media for the young and the elderly. Since 2003, she has successfully led several national research projects with a total funding of about 10 Million dollars from both government agencies and industry, including NRF, MOE, ASTAR, Microsoft Research and HP USA. She is the Editor-in-Chief of the International Journal of Information Technology published by the Singapore Computer Society.