SOAL: Second-order Online Active Learning*

Shuji Hao
Institute of High Performance Computing
Agency for Science Technology and Research
Singapore
haosj@ihpc.a-star.edu.sg

Peilin Zhao
Artificial Intelligence Department
Ant Financial
Hangzhou, China
peilin.zpl@antfin.com

Jing Lu
School of Information Systems
Singapore Management University
Singapore
jing.lu.2014@phdis.smu.edu.sg

Steven C. H. Hoi
School of Information Systems
Singapore Management University
Singapore
chhoi@smu.edu.sg

Chunyan Miao
School of Computer Engineering
Nanyang Technological University
Singapore
ascymiao@ntu.edu.sg

Chi Zhang
Interdisciplinary Graduate School
Nanyang Technological University
Singapore
czhang024@e.ntu.edu.sg

***Abstract*—This paper investigates the problem of online active learning for training classification models from sequentially arriving data. This is more challenging than conventional online learning tasks since the learner not only needs to figure out how to effectively update the classifier but also needs to decide when is the best time to query the label of an incoming instance given limited label budget. The existing online active learning approaches are often based on first-order online learning methods which generally fall short in slow convergence rate and sub-optimal exploitation of available information when querying the labeled data. To overcome the limitations, in this paper, we present a new framework of Second-order Online Active Learning (SOAL), which fully exploits both first-order and second-order information to achieve high learning accuracy with low labeling cost. We conduct both theoretical analysis and empirical studies for evaluating the proposed SOAL algorithm extensively. The encouraging results show clear advantages of the proposed algorithm over a family of state-of-the-art online active learning algorithms.**

I. Introduction

Online learning represents a family of efficient and scalable machine learning algorithms which are promising for large-scale learning tasks from big data streams [1]–[6]. Online learning typically works in a sequential manner. Consider online binary classification as an example. At time t, the learner receives an instance $\mathbf{x}_t$ from the environment, and then makes a prediction of its class label $\hat{y}_t = \text{sign}(f(\mathbf{x}_t))$. After making the prediction, it often assumes the true label $y_t \in \{+1, -1\}$ will be revealed from the environment, and then make an update of the classifier whenever necessary (e.g., wrongly classified $\hat{y}_t \neq y_t$ or other criteria). In contrast to traditional batch learning that often suffers from expensive re-training cost when new training data comes, online learning avoids re-training and learns incrementally from data streams in an efficient and scalable way.

Although various online learning algorithms have been proposed over the past decades [1], [7], conventional supervised online learning methods often assume the feedback (e.g., the class label in a classification task) is always revealed to the learner at the end of each iteration. However, this is not always true for many real applications where data streams could be unlabeled and the manually labeling cost could be quite expensive in many scenarios. For example, in the social media, data usually comes with a high speed and volume, which usually makes labeling all of the instances costly and nearly infeasible. This has raised the challenging problem "online active learning", which aims to maximize the learning efficacy while minimizing human labeling cost in stream data classification.

Some existing studies have attempted to address the challenge. A pioneering study is the "Perceptron-based active learning" [8], where the learner decides when to query by drawing a Bernoulli sampling of random variable $Z_t \in \{0, 1\}$ with parameter $\delta/(\delta + |p_t|)$, where $|p_t|$ is a form of prediction margin and $\delta > 0$ is a sampling parameter. If $Z_t = 1$, the learner will then place a query. The similar approach has also been used by the online Passive Aggressive (PA) active learning in another recent study [9]. Despite their simplicity, these algorithms often suffer some critical limitations. First, they often adopt first-order online learning algorithms for training. 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, particularly in the early rounds of online learning.

To overcome these limitations, we present a new Second-order Online Active Learning (SOAL) algorithm, which explores second-order online learning techniques for both training the classifiers and forming the query strategy. Specifically, the proposed SOAL adopts Adaptive Regularization Of Weights (AROW) algorithm [10] as the classifier, and devises an effective query strategy by exploiting both margin and second-order confidence information. We analyze the mistake bound of the SOAL algorithm in theory, and further validate its effectiveness via extensive empirical studies, which show the high scalability and efficacy of the proposed algorithms.

The rest of this paper is organized as follows. Section II

$*$ This work was done when the first author was with Nanyang Technological University, Singapore.

gives a brief overview of related work in literature. Section III presents the proposed SOAL algorithm and analyzes its mistake bound. Section IV discusses the results of our empirical studies, and finally Section V concludes this work.

II. Related Work

Our work is related to two major groups of studies in machine learning literature: online learning and active learning.

A. Online Learning

Online learning has been extensively studied in machine learning community [11]. Consider supervised online learning tasks, there are two major groups of studies: (i) first-order online learning, where only the first-order feature information is exploited. Example algorithms include Perceptron [11] and Passive-Aggressive (PA) algorithms [12], Online Gradient Descent, etc; (ii) Second-order online learning, where second-order information, such as covariance matrix of features, is exploited. Example algorithms include the Second-Order Perceptron (SOP) [13], Confidence-Weighted (CW) learning [14], and the Adaptive Regularization Of Weights (AROW) algorithm [10], etc. Most of these methods often assume a fully supervised learning settings, where the class label is always revealed to the learner at the end of each learning iteration, a scenario which is not always realistic for many real-world applications.

B. Active Learning

Active learning is a family of machine learning technique for actively querying informative unlabeled data to improve learning efficacy while reducing overall labeling cost. It has been extensively studied in machine learning literature [15], [16]. Existing active learning techniques could be generally grouped into four categories: uncertainty-based [17], searching through the hypothesis space [18], minimizing the expected error and variance on the pool of unlabeled instances [19], and exploiting the structure information [20] among the instances. More about batch active learning studies can be found in the comprehensive survey [21].

Batch-based active learning algorithms have shown promising results in reducing labeling cost on several applications, such as text classification, image recognitions and abnormal detection and so on. However, most of them require that all of the data should be prepared firstly before the active learning process. This makes them infeasible in some real-world applications, such as in online social media, data usually comes sequentially. To overcome this challenge, researchers has studied online active learning (OAL) [6], [9], [22] , also known as selective sampling, which aims to tackle the learning on data streams by combining both the efficiency of online learning and the effectiveness of active learning. However, the existing works often suffer from two major limitations. First, the learning efficacy (in terms of accuracy) of these algorithms is limited as most of them adopt first-order based online learning algorithms. Second, their active query strategy often strongly rely on the weight vector, which may be not 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.

III. Second-order Online Active Learning

In general, there are two open challenges with an online active learning task: (i) how to devise an effective query strategy that queries the most informative unlabeled example for training; and (ii) how to update the classifier more effectively once a query has been placed and the feedback is revealed to the learner. In the following, we present a new framework of Second-order Online Active Learning.

A. Problem Formulation

Without loss of generality, consider a learning for a typical online binary classification task. At time t, 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 i-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 $f(\mathbf{w}_t) = \mathrm{sign}(\mathbf{w}_t^\top \mathbf{x}_t)$, where $\mathbf{w}_t \in \mathbb{R}^d$ is the weight vector at the t-th round.

Unlike regular supervised online learning, when receiving $\mathbf{x}_t$, online active learning needs to decide whether to query its true label y_t. If the true label is queried, the algorithm can adopt regular online learning techniques to update the model $\mathbf{w}_t$. Otherwise, the algorithm will ignore the instance and process the next instance. In this way, online active learning aims to query the true labels of a small fraction of informative instances, and at the same time achieves a comparable accuracy with the regular online learning algorithms which query the true labels of all the instances.

In this article, we assume the model follows a Gaussian distribution [23], i.e., $\mathbf{w} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and predicts the label $\hat{y}_t = \mathrm{sign}(p_t)$, where $p_t = \mathbf{w}^\top \mathbf{x}_t$. In practice, however, it is often easier to simply use the average weight vector $\boldsymbol{\mu} = \mathbb{E}[\mathbf{w}]$ to make predictions. 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.

B. SOAL Algorithm

The proposed algorithm mainly consists of two parts: 1) how to update the model when the true label is obtained, and 2) when to query the label of an unlabeled instance. We discuss each issue in detail below.

1) How to Update: At the t-th round, if the true label y_t of $\mathbf{x}_t$ is provided, we will update the distribution by minimizing the following objective function

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

where

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

$\mathbf{g}_t = \partial \ell_t(\boldsymbol{\mu}_t) = -y_t\mathbf{x}_t$, and $\ell_t(\boldsymbol{\mu}_t) = \max(0, 1 - y_t\boldsymbol{\mu}_t^T\mathbf{x}_t)$ is the hinge loss function adopted. η, γ are positive parameters. This objective has three terms. 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 optimize the confidence of the model.

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 to zero, i.e., $\partial_{\boldsymbol{\Sigma}}\mathcal{C}_t(\boldsymbol{\mu}, \boldsymbol{\Sigma}_{t+1}) = 0$, we can derive the closed-form update:

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

where the Woodbury identity is used.

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

$$\boldsymbol{\mu}_{t+1} = \boldsymbol{\mu}_t - \eta\boldsymbol{\Sigma}_t\mathbf{g}_t, \tag{2}$$

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.,

$$\boldsymbol{\mu}_{t+1} = \boldsymbol{\mu}_t - \eta\boldsymbol{\Sigma}_{t+1}\mathbf{g}_t, \tag{3}$$

which should be more accurate than the update in Equation (2). To intuitively explain the above updating Equation (3), let us assume $\boldsymbol{\Sigma}_{t+1}$ is a diagonal matrix. Then, this update rule actually assigns different feature dimension with different learning rate, so that the less confident weights will be updated more aggressively. In order to handle high-dimensional data, we can only keep the diagonal elements of $\boldsymbol{\Sigma}$, and the updating rules in Equation (1) and (3) becomes

$$\begin{aligned} \boldsymbol{\Sigma}_{t+1} &= \boldsymbol{\Sigma}_t - \frac{\boldsymbol{\Sigma}_t \odot \mathbf{x}_t \odot \mathbf{x}_t \odot \boldsymbol{\Sigma}_t}{\gamma + (\mathbf{x}_t \odot \boldsymbol{\Sigma}_t)^\top\mathbf{x}_t}, \quad (4) \\ \boldsymbol{\mu}_{t+1} &= \boldsymbol{\mu}_t - \eta\boldsymbol{\Sigma}_{t+1} \odot \mathbf{g}_t, \quad (5) \end{aligned}$$

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

2) When to Query: For each incoming instance $\mathbf{x}_t$, we consider two factors when designing a query strategy for active learning.

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 decision boundary of the current classifier $\mathbf{w}_t$. The larger the value of $|p_t|$ is, the more certain the classifier is about its prediction on the instance $\mathbf{x}_t$.

The second factor is the confidence value of the model

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

Algorithm 1 SOAL:Second-order Online Active Learning.

Input: learning rate η; regularization parameter γ,
Initialize: $\boldsymbol{\mu}_1 = 0$, $\boldsymbol{\Sigma}_1 = I$.
for $t = 1, \dots, T$ **do**
  Receive $\mathbf{x}_t \in \mathbb{R}^d$;
  Compute $p_t = \boldsymbol{\mu}_t^\top\mathbf{x}_t$, $v_t = \mathbf{x}_t^\top\boldsymbol{\Sigma}_t\mathbf{x}_t$ and $c_t = \frac{-\eta\gamma v_t}{2\gamma + 2v_t}$;
  Compute $\rho_t = |p_t| + c_t$;
  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**
      Update model with Eq. (1) or (4), and (3) or (5)
    end if
  end if
end for

where $v_t = Var[\mathbf{w}_t^\top\mathbf{x}_t] = \mathbf{x}_t^\top\boldsymbol{\Sigma}_t\mathbf{x}_t$ models the variance of the model on $\mathbf{x}_t$. In other words, v_t characterizes how often $\mathbf{x}_t$ is seen by the classifier till the t-th round. When v_t is large (c_t is small, the model is not confident on the classifier), it means 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 of $\mathbf{x}_t$. When v_t is small (c_t is large, and the model is more confident on the classifier), it means the classifier has well trained on the instances which are similar to $\mathbf{x}_t$ so far, and we should place a low chance to query the true label of $\mathbf{x}_t$.

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

$$\rho_t = |p_t| + c_t. \tag{6}$$

There are two cases to be considered. When $\rho_t \leq 0$, which means the model is least confident on the trained classifier, we will **always query** the label of instance no matter how large $|p_t|$ is. Compared to the traditional query strategy [8], [9], a large value of $|p_t|$ always results in a **small query probability**, no matter how unreliable the trained classifier is, our proposed strategy is more reasonable.

When $\rho_t > 0$, 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 more 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 $\rho_t \leq 0$, query y_t;
- Else $\rho_t > 0$, draw a Bernoulli random variable $Z_t \in \{0, 1\}$ of parameter $\frac{\delta}{\delta + \rho_t}$;

– If $Z_t = 1$, query x_t for true label y_t;
– Else $Z_t = 0$, discard x_t.

In summary, the proposed query strategy balances the trade-off between the uncertainty of instance and the confidence of model on the trained classifier.

Finally, Algorithm 1 summarizes the proposed algorithm.

C. 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 $\boldsymbol{\mu} \in \mathbb{R}^d$,*

$$\begin{aligned} &\mathbb{E}\left[\sum_{t=1}^T M_t\right] \\ &\leq \mathbb{E}\left[\sum_{t=1}^T Z_t \ell_t(\boldsymbol{\mu})\right] + \frac{D_{\boldsymbol{\mu}} + (1-\delta)^2 \|\boldsymbol{\mu}\|^2}{\eta\delta} \mathrm{Tr}(\boldsymbol{\Sigma}_{T+1}^{-1}) \qquad (7) \\ &\quad + \frac{1}{\delta}\mathbb{E}\left[\sum_{\rho_t<0} \frac{\eta\gamma v_t}{(\gamma + v_t)}\right] + \frac{2}{\delta}\mathbb{E}\left[\sum_{\rho_t>0} L_t\right] - \mathbb{E}\left[\sum_{t=1}^T L_t\right] \end{aligned}$$

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

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

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,

$$\begin{aligned} &\mathbb{E}\left[\sum_{t=1}^T M_t\right] \\ &\leq \mathbb{E}\sum_{t=1}^T Z_t \ell_t(\boldsymbol{\mu}) + \frac{2}{\delta}\mathbb{E}\left[\sum_{\rho_t>0} L_t\right] - \mathbb{E}\left[\sum_{t=1}^T L_t\right] \qquad (8) \\ &\quad + \frac{2}{\delta}\sqrt{D_{\boldsymbol{\mu}} + (1-\delta)^2\|\boldsymbol{\mu}\|^2}\sqrt{\gamma \mathrm{Tr}(\boldsymbol{\Sigma}_{T+1}^{-1}) \ln|\boldsymbol{\Sigma}_{T+1}^{-1}|}. \end{aligned}$$

IV. EXPERIMENTS

A. 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 [8];
- “APAII”: the state-of-the-art first-order Active Passive-Aggressive algorithm [9];
- “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 adopt the query strategy same 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 I 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].

TABLE I: Summary of datasets in the experiments.

Dataset	# Instances	# Features	Source
a8a	32,561	123	LIBSVM
covtype	116,405	54	LIBSVM
HIGGS	11,000,000	28	LIBSVM
kddcup99	494,012	41	UCI
letter	20,000	16	LIBSVM
magic04	19,002	10	UCI

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

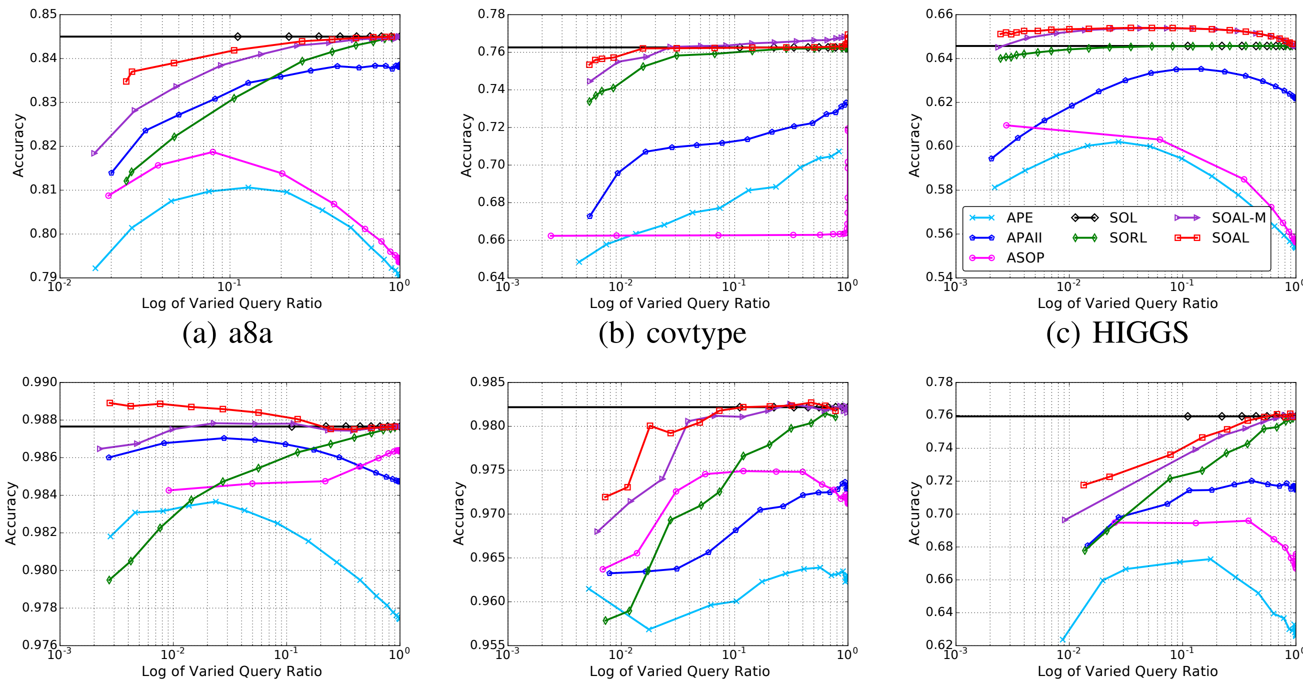
All of the algorithms are implemented with C++ language, and all of following experiments are conducted in a Ubuntu OS 64 bit PC with Intel Core i7-4770 CPU @ 3.40GHz $\times$ 8 and 16 GB memory.

B. Evaluation of Varied Query Ratio

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

First, on most of the cases, second-order based OAL algorithms (SOAL-M and SOAL) can outperform the first-order based OAL algorithms (APE and APAII). This is consistent with the results found in [10], [14] and confirms the necessity

[1] http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/
[2] http://www.ics.uci.edu/~mlearn/MLRepository.html



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

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

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

of considering second-order information to improve the predictive performance. Active Second-order 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 even performs worse than the first-order algorithm APAII, which makes it usually as a baseline.

Second, compared to the algorithm SORL which adopt the random query strategy, both the margin-based algorithm SOAL-M and SOAL algorithms proposed in this article can consistently achieve better performance. This observation indicate that both the margin-based query strategy in SOAL-M and our proposed query strategy in SOAL are effective to identify more informative instances to label thus can greatly reduce the cost in labeling.

Third, compared to the margin-based query strategy in SOAL-M, our proposed strategy in SOAL can consistently achieve higher accuracy with varied query ratio on all of the datasets. Compared to SOAL-M, we have imported a second term to evaluate how well the classifier is trained on t-th iteration. By considering both the margin-value and the confidence of us on the model, 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.

These observations make the proposed SOAL algorithm attractive in building real-world large-scale applications. Considering recommendation in social media for an example, data is usually coming with a high speed and volume. To label every data is usually costly and impossible. The proposed SOAL can quickly identify more informative instances to query thus greatly reducing the labeling cost by querying only a few of the instances. Meanwhile, the proposed algorithm also enjoy the efficacy of second-order online learners.

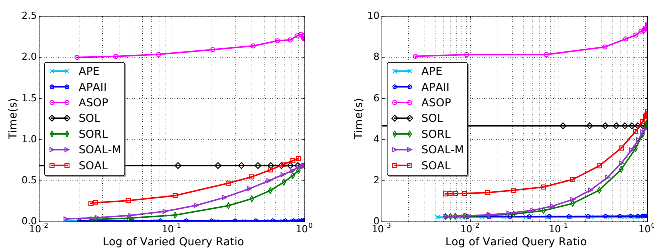
C. Evaluation of Scalability and Efficiency

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 two datasets in Fig. 2. 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 version SORL and the margin-based SORL-M algorithms due to the computation of the query strategy shown in Equation 6.

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 to update the model if the label of an instance is not queried. When query ratio increases, the time cost of these algorithms slowly approach to the SOL as expected. This indicated that the proposed algorithm SOAL can not only reduce the labeling cost shown in Fig. 1, but also can speed up the training process by updating the model with the queried instance alone.

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.



(a) a8a (b) covtype

Fig. 2: Evaluation of time cost (seconds) with respect to log of varied query ratio.

V. Conclusion

In this paper, we proposed a new framework of online active learning, named as Second-order Online Active Learning (SOAL) to overcome the labeling challenge of existing algorithms. We theoretically analyzed the mistake bound of the proposed SOAL algorithm and conducted a set of extensive experiments to examine its empirical effectiveness. For future work, we plan to explore some self-tuned learning strategy for automatically re-adjusting the parameters γ and η on the learning process. Besides, we also plan to investigate online active learning for AUC maximization [24], active learning for parallel computing [25].

Acknowledgement

This research is supported by the National Research Foundation, Prime Ministers Office, Singapore under its IDM Futures Funding Initiative.

References

[1] 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.

[2] S. Hao, P. Zhao, S. C. Hoi, and C. Miao, "Learning relative similarity from data streams: Active online learning approaches," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pp. 1181–1190, ACM, 2015.

[3] S. Hao, S. C. Hoi, C. Miao, and P. Zhao, "Active crowdsourcing for annotation," in *2015 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, vol. 2, pp. 1–8, IEEE, 2015.

[4] 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.

[5] 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)*, pp. 121–128, 2012.

[6] P. Zhao and S. C. Hoi, "Cost-sensitive online active learning with application to malicious url detection," in *Proceedings of the 19^{th} ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '13*, (New York, New York, USA), p. 919, ACM Press, 2013.

[7] 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.

[8] 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.

[9] J. Lu, P. Zhao, and S. C. Hoi, "Online passive aggressive active learning and its applications," in *Journal of Machine Learning Research - Proceedings Track (ACML2014)*, 2014.

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

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

[12] 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.

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

[14] 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.

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

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

[17] 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.

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

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

[20] 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.

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

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

[24] 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.

[25] S. Hao, M. Zhou, Z. Wu, P. Xu, and H. Zhang, "Highly parallel surface crack simulation," in *Computer Graphics, Imaging and Visualization (CGIV), 2011 Eighth International Conference on*, pp. 82–86, IEEE, 2011.