Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence

# An Adaptive Gradient Method for Online AUC Maximization

**Yi Ding**[1], **Peilin Zhao**[2], **Steven C.H. Hoi**[3], **Yew-Soon Ong**[1]
[1]School of Computer Engineering, Nanyang Technological University, 639798, Singapore
[2]Institute for Infocomm Research, 1 Fusionopolis Way, #21-01 Connexis, 138632, Singapore
[3]School of Information Systems, Singapore Management University, 178902, Singapore
{ding0077,asysong}@ntu.edu.sg, zhaop@i2r.a-star.edu.sg, chhoi@smu.edu.sg

## Abstract

Learning for maximizing AUC performance is an important research problem in machine learning. Unlike traditional batch learning methods for maximizing AUC which often suffer from poor scalability, recent years have witnessed some emerging studies that attempt to maximize AUC by single-pass online learning approaches. Despite their encouraging results reported, the existing online AUC maximization algorithms often adopt simple stochastic gradient descent approaches, which fail to exploit the geometry knowledge of the data observed in the online learning process, and thus could suffer from relatively slow convergence. To overcome the limitation of the existing studies, in this paper, we propose a novel algorithm of Adaptive Online AUC Maximization (AdaOAM), by applying an adaptive gradient method for exploiting the knowledge of historical gradients to perform more informative online learning. The new adaptive updating strategy by AdaOAM is less sensitive to parameter settings due to its natural effect of tuning the learning rate. In addition, the time complexity of the new algorithm remains the same as the previous non-adaptive algorithms. To demonstrate the effectiveness of the proposed algorithm, we analyze its theoretical bound, and further evaluate its empirical performance on both public benchmark datasets and anomaly detection datasets. The encouraging empirical results clearly show the effectiveness and efficiency of the proposed algorithm.

## Introduction

AUC (Area Under ROC curve) (Hanley and McNeil 1982) is an important measure for characterizing machine learning performances in many real-world applications, such as ranking, and anomaly detection tasks, especially when misclassification costs are unknown. In general, AUC measures the probability for a randomly drawn positive instance to have a higher decision value than a randomly sample negative instance. Many efforts have been devoted recently to developing efficient AUC optimization algorithms for both batch and online learning tasks (Cortes and Mohri 2003; Calders and Jaroszewicz 2007; Joachims 2005; Rudin and Schapire 2009; Zhao et al. 2011; Gao et al. 2013).



Due to its high efficiency and scalability in real-world applications, online AUC optimization for streaming data has been actively studied in the research community in recent years. The key challenge for AUC optimization in online setting is that AUC is a metric represented by the sum of pairwise losses between instances from different classes, which makes conventional online learning algorithms unsuitable for direct use in many real world scenarios. To address this challenge, two core types of Online AUC Maximization (OAM) frameworks have been proposed recently. The first framework is based on the idea of buffer sampling (Zhao et al. 2011; Kar et al. 2013), which stores some randomly sampled historical examples in a buffer to represent the observed data for calculating the pairwise loss functions. The other framework focuses on one-pass AUC optimization (Gao et al. 2013), where the algorithm scan through the training data only once. The benefit of one-pass AUC optimization lies in the use of squared loss to represent the AUC loss function while providing proofs on its consistency with the AUC measure (Gao and Zhou 2012).

Although these algorithms have been shown to be capable of achieving fairly good AUC performances, they share a common trait of employing the online (stochastic) gradient descent technique, which fails to take advantage of the geometry property of the data observed from the online learning process, while recent studies have shown the importance of exploiting this information for online optimization (Duchi, Hazan, and Singer 2011). To overcome the limitation of the existing works, we propose a novel framework of Adaptive Online AUC maximization (AdaOAM), which considers the adaptive gradient optimization technique for exploiting the geometric property of the observed data to accelerate online AUC maximization tasks. Specifically, the technique is motivated by a simple intuition, that is, the frequently occurring features in online learning process should be assigned with low learning rates while the rarely occurring features should be given high learning rates. To achieve this purpose, we propose the AdaOAM algorithm by adopting the adaptive gradient updating framework proposed by (Duchi, Hazan, and Singer 2011) to control the learning rates for different features. We theoretically prove that the regret bound of the proposed algorithm is better than those of the existing non-adaptive algorithms. We also empirically compared the proposed algorithm with several

state-of-the-art online AUC optimization algorithms on both benchmark datasets and real-world online anomaly detection datasets. The promising results further validate the effectiveness and efficiency of the proposed algorithm.

The rest of this paper is organized as follows. We first review related works from both online learning and AUC optimization, and then present the formulations of the proposed approach and its theoretical analysis; we further discuss our obtained experimental results, and the sensitivity of the parameters, and finally conclude the paper with a brief summary of our work.

## Related Work

Our work is closely related to two topics in the context of machine learning, namely, online learning and AUC optimization. Below we briefly review some of the important related works in both areas.

**Online Learning.** Online learning has been extensively studied in the machine learning communities (Cesa-Bianchi and Lugosi 2006; Crammer et al. 2006; Zhao, Hoi, and Jin 2011; Hoi et al. 2013; Zhao et al. 2014), mainly due to its high efficiency and scalability to large-scale datasets. Differing from conventional batch learning methods that assume all training instances are available prior to the learning phase, online learning considers one instance each time to update the model sequentially and iteratively. Therefore, online learning is ideally appropriate for tasks in which data arrives sequentially. A number of first-order algorithms have been proposed including the well-known Perceptron algorithm (Rosenblatt 1958) and the Passive-Aggressive (PA) algorithm (Crammer et al. 2006). Although the PA introduces the concept of "maximum margin" for classification, it fails to control the direction and scale of parameter updates during online learning phase. In order to address this issue, recent years have witnessed some second-order online learning algorithms (Dredze, Crammer, and Pereira 2008; Crammer, Kulesza, and Dredze 2009; Orabona and Crammer 2010; Wang, Zhao, and Hoi 2012), which apply parameter confidence information to improve online learning performance. Further, in order to solve the cost-sensitive classification tasks on-the-fly, online learning researchers have also proposed a few novel online learning algorithms to directly optimize some more meaningful cost-sensitive metrics (Wang, Zhao, and Hoi 2014; Zhao and Hoi 2013; Hoi and Zhao 2013).

**AUC Optimization.** AUC (Area Under ROC curve) is an important performance measure that has been widely used in imbalanced data distribution classification. The ROC curve explains the rate of the true positive against the false positive at various range of threshold. Thus, AUC represents the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one. Recently, many algorithms have been developed to optimize AUC directly (Cortes and Mohri 2003; Calders and Jaroszewicz 2007; Joachims 2005; Zhao et al. 2011; Gao et al. 2013). In (Joachims 2005), the author firstly presented a general framework for optimizing multivariate nonlinear performance measures such as the AUC, F1, etc. in a batch mode. However, it is worth investigating online learning algorithms for AUC optimization involving large-scale applications. Among the online AUC optimization approaches, two core online AUC optimization frameworks have been proposed very recently. The first framework is based on the idea of buffer sampling (Zhao et al. 2011; Kar et al. 2013), which employed a fixed-size buffer to represent the observed data for calculating the pairwise loss functions. A representative study is available in (Zhao et al. 2011), which leveraged the reservoir sampling technique to represent the observed data instances by a fixed-size buffer where notable theoretical and empirical results have been reported. Then, (Kar et al. 2013) studied the improved generalization capability of online learning algorithms for pairwise loss functions with the framework of buffer sampling. The main contribution of their work is the introduction of the stream subsampling with replacement as the buffer update strategy. The other framework which takes a different perspective was presented by (Gao et al. 2013). They extended the previous online AUC optimization framework with a regression-based one-pass learning mode, and achieved solid regret bounds by considering square loss for the AUC optimization task due to its theoretical consistency with AUC.

Despite the extensive works in both the fields of online learning and AUC optimization, to the best of our knowledge, our current work represents a first effort to explore adaptive gradient optimization and second order learning techniques for online AUC optimization. In particular, we explore the second order statistics of data to update the classifier adaptively at each stage and take full advantage of the geometrical information available in the data.

## An Adaptive Gradient Method for OAM

### Problem Setting

We aim to learn a linear classification model that maximizes AUC for a binary classification problem. Without loss of generality, we assume positive class to be less than negative class. Denote $(\mathbf{x}_t, y_t)$ as the training instance received at the $t$-th trial, where $\mathbf{x}_t \in \mathbb{R}^d$ and $y_t \in \{-1, +1\}$, and $\mathbf{w}_t \in \mathbb{R}^d$ is the weight vector learned so far.

Given this setting, let us define the AUC measurement (Hanley and McNeil 1982) for binary classification task. Given a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i) \in \mathbb{R}^d \times \{-1, +1\} | \ i \in [n]\}$, where $[n] = \{1, 2, \ldots, n\}$, we divide it into two sets naturally: the set of positive instances $\mathcal{D}_+ = \{(\mathbf{x}_i^+, +1) | \ i \in [n_+]\}$ and the set of negative instances $\mathcal{D}_- = \{(\mathbf{x}_j^-, -1) | \ j \in [n_-]\}$, where $n_+$ and $n_-$ are the numbers of positive and negative instances, respectively. For a linear classifier $\mathbf{w} \in \mathbb{R}^d$, its AUC measurement on $\mathcal{D}$ is defined as follows:

$$\text{AUC}(\mathbf{w}) = \frac{\sum_{i=1}^{n_+} \sum_{j=1}^{n_-} \mathbb{I}_{(\mathbf{w}\cdot\mathbf{x}_i^+ > \mathbf{w}\cdot\mathbf{x}_j^-)} + \frac{1}{2}\mathbb{I}_{(\mathbf{w}\cdot\mathbf{x}_i^+ = \mathbf{w}\cdot\mathbf{x}_j^-)}}{n_+ n_-},$$

where $\mathbb{I}_\pi$ is the indicator function that outputs a $'1'$ if the prediction $\pi$ holds and $'0'$ otherwise. We replace the indicator function with the following convex surrogate, i.e., the square loss from (Gao et al. 2013) due to its consistency with AUC (Gao and Zhou 2012)

$$\ell(\mathbf{w}, \mathbf{x}_i^+ - \mathbf{x}_j^-) = (1 - \mathbf{w} \cdot (\mathbf{x}_i^+ - \mathbf{x}_j^-))^2,$$

and find the optimal classifier by minimizing the following objective function

$$\mathcal{L}(\mathbf{w}) = \frac{\lambda}{2}\|\mathbf{w}\|_2^2 + \frac{\sum_{i=1}^{n_+}\sum_{j=1}^{n_-}\ell(\mathbf{w}, \mathbf{x}_i^+ - \mathbf{x}_j^-)}{2n_+n_-}. \quad (1)$$

where $\frac{\lambda}{2}\|\mathbf{w}\|_2^2$ is introduced to regularize the complexity of the linear classifier. Note, the optimal $\mathbf{w}_*$ satisfies $\|\mathbf{w}_*\|_2 \leq 1/\sqrt{\lambda}$ according to the strong duality theorem.

## Adaptive Online AUC Maximization

Now, we are ready to introduce the proposed Adaptive Online AUC Maximization (AdaOAM) algorithm. Following the similar approach in (Gao et al. 2013), we modify the loss function $\mathcal{L}(\mathbf{w})$ in (1) as a sum of losses for individual training instance $\sum_{t=1}^{T}\mathcal{L}_t(\mathbf{w})$ where

$$\mathcal{L}_t(\mathbf{w}) = \frac{\lambda}{2}\|\mathbf{w}\|_2^2 + \frac{\sum_{i=1}^{t-1}\mathbb{I}_{[y_i \neq y_t]}(1 - y_t(\mathbf{x}_t - \mathbf{x}_i)^\top\mathbf{w})^2}{2|i \in [t-1] : y_i y_t = -1|},$$

for i.i.d. sequence $\mathcal{S}_t = \{(\mathbf{x}_i, y_i)|i \in [t]\}$, and it is an unbiased estimation to $\mathcal{L}(\mathbf{w})$. $X_t^+$ and $X_t^-$ are denoted as the sets of positive and negative instances of $\mathcal{S}_t$ respectively, and $T_t^+$ and $T_t^-$ are their respective cardinalities. Besides, $\mathcal{L}_t(\mathbf{w})$ is set as 0 for $T_t^+T_t^- = 0$. If $y_t = 1$, the gradient is

$$\nabla\mathcal{L}_t(\mathbf{w}) = \lambda\mathbf{w} + \mathbf{x}_t\mathbf{x}_t^\top\mathbf{w} - \mathbf{x}_t + \frac{\sum_{i:y_i=-1}\mathbf{x}_i + (\mathbf{x}_i\mathbf{x}_i^\top - \mathbf{x}_i\mathbf{x}_t^\top - \mathbf{x}_t\mathbf{x}_i^\top)\mathbf{w}}{T_t^-}.$$

Using $c_t^- = \frac{1}{T_t^-}\sum_{i:y_i=-1}\mathbf{x}_i$ and $S_t^- = \frac{1}{T_t^-}\sum_{i:y_i=-1}(\mathbf{x}_i\mathbf{x}_i^\top - \mathbf{c}_t^-[\mathbf{c}_t^-]^\top)$ refer to the mean and covariance matrix of negative class, respectively, the gradient can be simplified as

$$\nabla\mathcal{L}_t(\mathbf{w}) = \lambda\mathbf{w} - \mathbf{x}_t + \mathbf{c}_t^- + (\mathbf{x}_t - \mathbf{c}_t^-)(\mathbf{x}_t - \mathbf{c}_t^-)^\top\mathbf{w} + S_t^-\mathbf{w}.$$

Similarly, if $y_t = -1$,

$$\nabla\mathcal{L}_t(\mathbf{w}) = \lambda\mathbf{w} + \mathbf{x}_t - \mathbf{c}_t^+ + (\mathbf{x}_t - \mathbf{c}_t^+)(\mathbf{x}_t - \mathbf{c}_t^+)^\top\mathbf{w} + S_t^+\mathbf{w},$$

where $c_t^+ = \frac{1}{T_t^+}\sum_{i:y_i=1}\mathbf{x}_i$ and $S_t^+ = \frac{1}{T_t^+}\sum_{i:y_i=1}(\mathbf{x}_i\mathbf{x}_i^\top - \mathbf{c}_t^+[\mathbf{c}_t^+]^\top)$ are the mean and covariance matrix of positive class, respectively.

Upon obtaining gradient $\nabla\mathcal{L}_t(\mathbf{w})$, due to $\|\mathbf{w}_*\| \leq 1/\sqrt{\lambda}$, the model can be updated with $\mathbf{w}_t = \Pi_{\frac{1}{\sqrt{\lambda}}}(\mathbf{w}_{t-1} - \eta_t\hat{\mathbf{g}}_t(\mathbf{w}_{t-1}))$, where $\Pi_{\frac{1}{\sqrt{\lambda}}}(\mathbf{w}) = \min(1, \frac{1/\sqrt{\lambda}}{\|\mathbf{w}\|_2})\mathbf{w}$, $\eta_t$ is the learning rate for the t-th iteration, $\hat{\mathbf{g}}_t$ is the adaptive gradient calculated using the Adaptive Gradient Updating (AGU) strategy, which will be detailed in the next subsection.

Finally, Algorithm 1 summarizes the proposed AdaOAM method. If setting $\Gamma_t^+ = S_t^+$ and $\Gamma_t^- = S_t^-$, the covariance matrices are updated as follows:

$$\Gamma_t^+ = \Gamma_{t-1}^+ + \mathbf{c}_{t-1}^+[\mathbf{c}_{t-1}^+]^\top - \mathbf{c}_t^+[\mathbf{c}_t^+]^\top + \frac{\mathbf{x}_t\mathbf{x}_t^\top - \Gamma_{t-1}^+ - \mathbf{c}_{t-1}^+[\mathbf{c}_{t-1}^+]^\top}{T_t^+},$$

$$\Gamma_t^- = \Gamma_{t-1}^- + \mathbf{c}_{t-1}^-[\mathbf{c}_{t-1}^-]^\top - \mathbf{c}_t^-[\mathbf{c}_t^-]^\top + \frac{\mathbf{x}_t\mathbf{x}_t^\top - \Gamma_{t-1}^- - \mathbf{c}_{t-1}^-[\mathbf{c}_{t-1}^-]^\top}{T_t^-}.$$

**Algorithm 1** The AdaOAM Algorithm

**Input**: The regularization parameter $\lambda$, the learning rate $\{\eta_t\}_{t=1}^T$, $\delta \geq 0$ for AGU update.
**Initialize** $\mathbf{w}_0 = \mathbf{0}$, $\mathbf{c}_0^+ = \mathbf{c}_0^- = \mathbf{0}$, $T_0^+ = T_0^- = 0$, $\Gamma_0^+ = \Gamma_0^- = [0]_{d\times d}$.
**for** $t = 1, 2, \ldots, T$ **do**
  Receive an incoming instance $(\mathbf{x}_t, y_t)$;
  **if** $y_t = +1$ **then**
    $T_t^+ = T_{t-1}^+ + 1$, $T_t^- = T_{t-1}^-$;
    $\mathbf{c}_t^+ = \mathbf{c}_{t-1}^+ + \frac{1}{T_t^+}(\mathbf{x}_t - \mathbf{c}_{t-1}^+)$ and $\mathbf{c}_t^- = \mathbf{c}_{t-1}^-$;
    Update $\Gamma_t^+$ and $\Gamma_t^- = \Gamma_{t-1}^-$;
    $\hat{\mathbf{g}}_t$ updated by AGU;
  **else**
    $T_t^- = T_{t-1}^- + 1$, $T_t^+ = T_{t-1}^+$;
    $\mathbf{c}_t^- = \mathbf{c}_{t-1}^- + \frac{1}{T_t^-}(\mathbf{x}_t - \mathbf{c}_{t-1}^-)$ and $\mathbf{c}_t^+ = \mathbf{c}_{t-1}^+$;
    Update $\Gamma_t^-$ and $\Gamma_t^+ = \Gamma_{t-1}^+$;
    $\hat{\mathbf{g}}_t$ updated by AGU;
  **end if**
  $\mathbf{w}_t = \Pi_{\frac{1}{\sqrt{\lambda}}}(\mathbf{w}_{t-1} - \eta_t\hat{\mathbf{g}}_t)$;
**end for**

## Adaptive Gradient Updating

In order to perform feature-wise gradient updating, we employ the second order gradient optimization technique, i.e., Adaptive Gradient Updating (AGU) strategy, inspired by (Duchi, Hazan, and Singer 2011), which is detailed in the following Algorithm 2.

**Algorithm 2** Adaptive Gradient Updating (AGU)

**Input**: $\delta \geq 0$.
**Output**: Adjusted gradient $\hat{\mathbf{g}}_t$.
**Variables**: $s \in \mathbb{R}^d$, $H \in \mathbb{R}^{d\times d}$, $g_{1:t,i} \in \mathbb{R}^t$ for $i \in 1, \ldots, d$.
**Initialize** $g_{1:0} = [\,]$.
**for** $t = 1, 2, \ldots, T$ **do**
  Suffer loss $\mathcal{L}_t(\mathbf{w})$;
  Receive gradient $\mathbf{g}_t = \nabla\mathcal{L}_t(\mathbf{w})$;
  Update $g_{1:t} = [g_{1:t-1}\ \mathbf{g}_t]$, $s_{t,i} = \|g_{1:t,i}\|_2$;
  $H_t = \delta I + diag(s_t)$;
  $\hat{\mathbf{g}}_t = H_t^{-1}\mathbf{g}_t$;
**end for**
Return $\hat{\mathbf{g}}_t$

The intuition of this strategy is very natural, which considers the rare occurring features as more informative and discriminative than those frequently occurring features. Thus, these informative rare occurring features should be updated with higher learning rates by incorporating the geometrical property of the data observed in earlier stages. Besides, by using the previously observed gradients, the update process can mitigate the effects of noise and speed up the convergence rate intuitively. In order to reduce the computation efforts incurred, we adopted the roots of the diagonal matrices (approximation to the Hessian of the loss functions). Further, the smooth parameter $\delta > 0$ is introduced to make the

diagonal matrix invertible and the algorithm robust, which is usually set as a very small value.

## Theoretical Analysis

This section presents our main theoretical results. First, we give the regret bound of the proposed AdaOAM algorithm.

**Theorem 1.** *Assume* $\|\mathbf{w}_t\| \leq 1/\sqrt{\lambda}, \quad (\forall t \in [T])$ *and the diameter of* $\chi = \{\mathbf{w}|\|\mathbf{w}\| \leq \frac{1}{\sqrt{\lambda}}\}$ *is bounded via* $sup_{\mathbf{w},\mathbf{u}\in\chi}\|\mathbf{w}-\mathbf{u}\|_\infty \leq D_\infty$*, we have*

$$\sum_{t=1}^{T}[\mathcal{L}_t(\mathbf{w}_t) - \mathcal{L}_t(\mathbf{w}_*)] \leq 2D_\infty \sum_{i=1}^{d}\sqrt{\sum_{t=1}^{T}[(\lambda \mathbf{w}_{t,i})^2 + C(r_{t,i})^2]},$$

*where* $C \leq (1+\frac{2}{\sqrt{\lambda}})^2$*, and* $r_{t,i} = \max_{j<t}|\mathbf{x}_{j,i} - \mathbf{x}_{t,i}|$.

*Proof.* We first define $\mathbf{w}_*$ as $\mathbf{w}_* = \arg\min_{\mathbf{w}} \sum_t \mathcal{L}_t(\mathbf{w})$.

Based on the regularizer $\frac{\lambda}{2}\|\mathbf{w}\|^2$, it is easy to obtain $\|\mathbf{w}_*\|^2 \leq 1/\lambda$ due to the strong convexity property, and it is also reasonable to restrict $\mathbf{w}_t$ by $\|\mathbf{w}_t\|^2 \leq 1/\lambda$. Denote the projection of a point $\mathbf{w}$ onto $\|\mathbf{u}\|_2 \leq \frac{1}{\sqrt{\lambda}}$ according to L2-norm by $\Pi_{\frac{1}{\sqrt{\lambda}}}(\mathbf{w}) = \arg\min_{\|\mathbf{u}\|\leq\frac{1}{\sqrt{\lambda}}}\|\mathbf{u}-\mathbf{w}\|_2$. After introducing the above, our adaptive subgradient descent employs the following update:

$$\mathbf{w}_{t+1} = \Pi_{\frac{1}{\sqrt{\lambda}}}(\mathbf{w}_t - \eta H_t^{-1}\mathbf{g}_t).$$

Concretely, for some small fixed $\delta \geq 0$, we set $H_t = \delta I + diag(s_t)$ and $\psi_t(\mathbf{g}_t) = \langle \mathbf{g}_t, H_t\mathbf{g}_t\rangle$. We also denote the dual norm of $\|\cdot\|_{\psi_t}$ by $\|\cdot\|_{\psi_t^*}$, in which case $\|\mathbf{g}_t\|_{\psi_t^*} = \|\mathbf{g}_t\|_{H_t^{-1}}$.

From (Duchi, Hazan, and Singer 2011), it is known that

$$\sum_{t=1}^{T}\|\mathbf{g}_t\|_{\psi_t^*}^2 \leq 2\sum_{i=1}^{d}\|g_{1:T,i}\|_2.$$

In addition, we consider the composite mirror descent method to update the gradient in our case. So we arrive at

$$\mathbf{w}_{t+1} = \arg\min_{\|\mathbf{w}\|\leq\frac{1}{\sqrt{\lambda}}}\{\eta\langle \mathbf{g}_t, \mathbf{w}\rangle + \eta\varphi(\mathbf{w}) + B_{\psi_t}(\mathbf{w}, \mathbf{w}_t)\},$$

where $B_{\psi_t}(\mathbf{w}, \mathbf{w}_t)$ is the Bregman divergence associated with a strongly convex and differentiable function $\psi_t$. In our case, the regularization function $\varphi \equiv 0$. Thus, we have a regret bound of

$$\sum_{t=1}^{T}[\mathcal{L}_t(\mathbf{w}_t) - \mathcal{L}_t(\mathbf{w}_*)] \leq \sqrt{2}D_\infty \sum_{i=1}^{d}\|g_{1:T,i}\|_2,$$

where $\chi = \{\mathbf{w}|\|\mathbf{w}\| \leq \frac{1}{\sqrt{\lambda}}\}$ is bounded via $sup_{\mathbf{w},\mathbf{u}\in\chi}\|\mathbf{w}-\mathbf{u}\|_\infty \leq D_\infty$. Next, we would like to analyze the features' dependency on the data of the gradient. Since

$$(g_{t,i})^2 \leq \Big[\lambda\mathbf{w}_{t,i} + \frac{\sum_{j=1}^{t-1}(1 - y_t\langle \mathbf{x}_t - \mathbf{x}_j, \mathbf{w}\rangle)y_t(\mathbf{x}_{j,i} - \mathbf{x}_{t,i})}{T_t^-}\Big]^2$$
$$\leq 2(\lambda\mathbf{w}_{t,i})^2 + 2C(\mathbf{x}_{j,i} - \mathbf{x}_{t,i})^2 = 2(\lambda\mathbf{w}_{t,i})^2 + 2C(r_{t,i})^2,$$

where $C \leq (1+\frac{2}{\sqrt{\lambda}})^2$ is a constant to bound the scalar of the second term of the right side, and $r_{t,i} = \max_{j<t}|\mathbf{x}_{j,i} - \mathbf{x}_{t,i}|$, we have

$$\sum_{i=1}^{d}\|g_{1:T,i}\|_2 = \sum_{i=1}^{d}\sqrt{\sum_{t=1}^{T}(g_{t,i})^2}$$
$$\leq \sqrt{2}\sum_{i=1}^{d}\sqrt{\sum_{t=1}^{T}[(\lambda\mathbf{w}_{t,i})^2 + C(r_{t,i})^2]}.$$

Finally, combining the above inequalities, we arrive at

$$\sum_{t=1}^{T}[\mathcal{L}_t(\mathbf{w}_t) - \mathcal{L}_t(\mathbf{w}_*)] \leq 2D_\infty \sum_{i=1}^{d}\sqrt{\sum_{t=1}^{T}[(\lambda\mathbf{w}_{t,i})^2 + C(r_{t,i})^2]}.$$

□

From the proof above, we can conclude that Algorithm 2 should have a lower regret than non-adaptive algorithms due to its dependence on the geometry of the underlying data space. If the features have been normalized and sparse, the gradient terms in the bound $\sum_{i=1}^{d}\|g_{1:T,i}\|_2$ should be much smaller than $\sqrt{T}$, which leads to lower regret and faster convergence. If the feature space is relative dense, then the convergence rate will be $O(1/\sqrt{T})$ for the general case as in OPAUC and OAM methods.

## Experimental Results

In this section, we evaluate the proposed AdaOAM algorithm in terms of AUC performance, convergence rate, and examine its parameter sensitivity. The framework of experiments is based on the open-source library for large-scale online learning LIBOL [1] (Hoi, Wang, and Zhao 2014).

### Comparison Algorithms

We compare the proposed algorithm with other state-of-the-art online AUC optimization algorithms. Specifically, the algorithms considered in our empirical studies include:

- **OAM$_{\text{seq}}$**: the OAM algorithm with reservoir sampling and sequential updating method (Zhao et al. 2011);
- **OAM$_{\text{gra}}$**: the OAM algorithm with reservoir sampling and online gradient updating method (Zhao et al. 2011);
- **OPAUC**: the one-pass AUC optimization algorithm proposed in (Gao et al. 2013);
- **AdaOAM**: the proposed adaptive gradient approach for online AUC maximization.

### Experimental Testbed and Setup

To examine the performances of the proposed AdaOAM in comparison to existing state-of-the-art methods, we conduct extensive experiments on various benchmark datasets by maintaining consistency to the previous studies on online AUC maximization (Zhao et al. 2011; Gao et al. 2013). Table 1 shows the details of 6 binary-class datasets in our

[1] http://libol.stevenhoi.org/

Table 1: Details of benchmark machine learning datasets.

| Dataset | # instances | # dimensions | $T_-/T_+$ |
|---|---|---|---|
| german | 1,000 | 24 | 2.3333 |
| svmguide3 | 1,243 | 22 | 3.1993 |
| vehicle | 846 | 18 | 3.2513 |
| acoustic | 78,823 | 50 | 3.3165 |
| svmguide4 | 300 | 10 | 5.8181 |
| w1a | 2,477 | 300 | 33.4028 |

Table 2: Evaluation on benchmark datasets.

| Algorithm | german | | svmguide3 | |
|---|---|---|---|---|
| | AUC | Time(s) | AUC | Time(s) |
| $\text{OAM}_{seq}$ | 0.7333 ± 0.0367 | 0.9610 | 0.7001 ± 0.0444 | 1.2152 |
| $\text{OAM}_{gra}$ | 0.7208 ± 0.0361 | 0.9777 | 0.6969 ± 0.0471 | 1.1969 |
| OPAUC | 0.7263 ± 0.0683 | 0.0201 | 0.7205 ± 0.0376 | 0.0242 |
| AdaOAM | **0.7719 ± 0.0371** | 0.0430 | **0.7358 ± 0.0366** | 0.0500 |
| Algorithm | vehicle | | acoustic | |
| | AUC | Time(s) | AUC | Time(s) |
| $\text{OAM}_{seq}$ | 0.7760 ± 0.0446 | 0.8202 | 0.8665 ± 0.0148 | 77.2125 |
| $\text{OAM}_{gra}$ | 0.7531 ± 0.0468 | 0.7966 | 0.8675 ± 0.0109 | 77.7828 |
| OPAUC | 0.7597 ± 0.0311 | 0.0138 | 0.8881 ± 0.0022 | 3.5908 |
| AdaOAM | **0.7968 ± 0.0274** | 0.0325 | **0.8949 ± 0.0020** | 5.4712 |
| Algorithm | svmguide4 | | w1a | |
| | AUC | Time(s) | AUC | Time(s) |
| $\text{OAM}_{seq}$ | 0.7829 ± 0.0519 | 0.2755 | 0.8622 ± 0.0479 | 3.6978 |
| $OAM_{gra}$ | 0.7619 ± 0.0876 | 0.2684 | 0.8741 ± 0.0424 | 3.7090 |
| OPAUC | 0.7404 ± 0.0779 | 0.0043 | 0.9015 ± 0.0338 | 2.1306 |
| AdaOAM | **0.8190 ± 0.0894** | 0.0103 | **0.9180 ± 0.0386** | 2.1241 |

experiments. All of these can be downloaded from LIBSVM [2] and UCI machine learning repository [3]. Note that several datasets (svmguide4, vehicle, acoustic) are originally multi-class, which were converted to class-imbalanced binary datasets in our experiments.

In the experiments, the features have been normalized fairly. Each dataset has been randomly divided into 5 folds, in which 4 folds are for training and the remaining fold is for testing. We also generate 4 independent 5-fold partitions per dataset to further reduce the variations. Therefore, the reported AUC value is an average of 20 runs for each dataset. 5-fold cross validation is conducted on the training sets to decide the learning rate $\eta \in 2^{[-10:10]}$ and the regularization parameter $\lambda \in 2^{[-10:2]}$. For $\text{OAM}_{gra}$ and $\text{OAM}_{seq}$, the buffer size is fixed at 100 as suggested in (Zhao et al. 2011). All experiments were run with MATLAB on a computer workstation with 16GB memory and 3.20GHz CPU.

## Evaluation on Benchmark Datasets

Table 2 summarizes the average AUC performance of the compared algorithms over the 6 datasets. It is clear from the results that the proposed method is superior to the other three existing online AUC optimization algorithms considered for comparison. In particular, AdaOAM not only achieves the best AUC scores among all the methods, but also runs as efficiently as OPAUC, which is computationally more efficient and scalable than both $\text{OAM}_{seq}$ and $\text{OAM}_{gra}$.

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

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

Table 3: Details of anomaly detection datasets.

| Dataset | # instances | # dimensions | $T_-/T_+$ |
|---|---|---|---|
| webspam | 350,000 | 254 | 1.5397 |
| cod-rna | 271,617 | 8 | 2.0000 |
| smartBuilding | 20,000 | 14 | 85.2069 |
| malware | 71,709 | 122 | 188.7063 |

Table 4: Evaluation on anomaly detection datasets.

| Algorithm | webspam | | cod-rna | |
|---|---|---|---|---|
| | AUC | Time(s) | AUC | Time(s) |
| $\text{OAM}_{seq}$ | 0.9634 ± 0.0050 | 997.0423 | 0.9615 ± 0.0109 | 59.6678 |
| $OAM_{gra}$ | 0.9626 ± 0.0050 | 992.7027 | 0.9379 ± 0.0062 | 59.9986 |
| OPAUC | 0.9538 ± 0.0062 | 20.6425 | 0.9190 ± 0.0032 | 0.8085 |
| AdaOAM | **0.9647 ± 0.0057** | 20.7096 | **0.9672 ± 0.0020** | 2.0404 |
| Algorithm | smartBuilding | | malware | |
| | AUC | Time(s) | AUC | Time(s) |
| $\text{OAM}_{seq}$ | 0.5962 ± 0.0553 | 28.9850 | 0.9596 ± 0.0167 | 70.8177 |
| $\text{OAM}_{gra}$ | 0.5969 ± 0.0665 | 23.4805 | 0.9529 ± 0.0145 | 69.7704 |
| OPAUC | 0.5989 ± 0.0649 | 0.3039 | 0.9091 ± 0.0150 | 12.4623 |
| AdaOAM | **0.7001 ± 0.0592** | 0.7678 | **0.9669 ± 0.0137** | 14.2976 |

## Application to Online Anomaly Detection

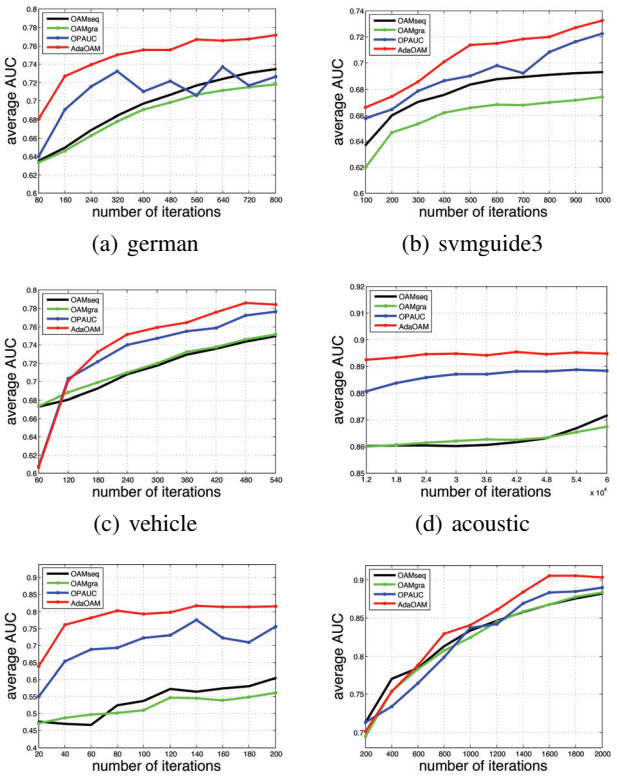
The AdaOAM can also be potentially applied to solving online anomaly detection problems. Concretely, we explore online anomaly detection tasks in the following four application domains:

- **Webspam**: We apply our algorithm to detect malicious web pages using the "webspam" dataset.
- **Bioinformatics**: We apply our algorithm to solve a bioinformatics problem with the "Cod-RNA" dataset, which aims to detect non-coding RNAs from some large sequenced genomes.
- **Sensor Faults**: We apply our algorithm to identify sensor faults in buildings with the "smartBuilding" dataset (Michaelides and Panayiotou 2009), where the sensors monitor the concentration of the contaminant of interest (such as CO2) in different zones in a building.
- **Malware App**: We apply our algorithm to detect mobile malware app with a "malware" app permission dataset, which is built from the Android Malware Genome Project [4] (Zhou and Jiang 2012). In our experiment, we adopt the dataset preprocessed by (Peng et al. 2012) after data cleansing and duplication removal.

Table 3 summarizes the details of these datasets related to the above four different domains.

Table 4 summarizes the performance for online anomaly detection task. From Table 4, we observe that the proposed AdaOAM algorithm also outperforms other methods. Although $\text{OAM}_{seq}$ and $\text{OAM}_{gra}$ obtain comparably good results, their computational costs are very high, which are impractical for real-world learning tasks. Again, the AdaOAM proves its efficiency for real-world applications.

[4] http://www.malgenomeproject.org/

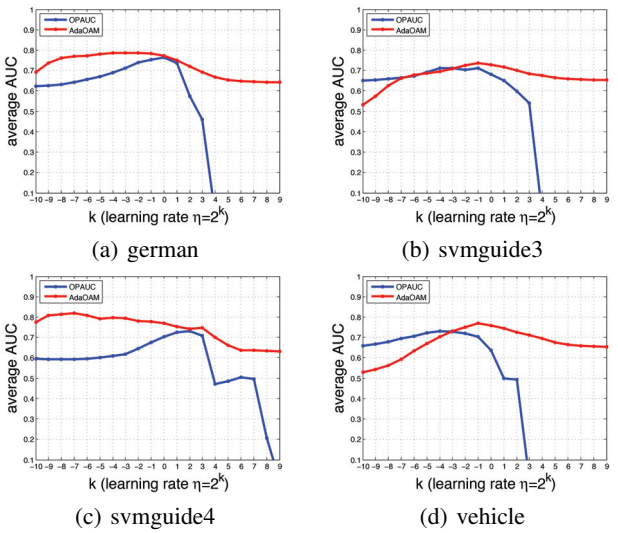



(a) german (b) svmguide3

(c) vehicle (d) acoustic

(e) svmguide4 (f) w1a

Figure 1: Parameter sensitivity on benchmark datasets.




(a) german (b) svmguide3

(c) svmguide4 (d) vehicle

Figure 2: Parameter sensitivity on benchmark datasets.

## Evaluation of Convergence Rate

We now examine the convergence rate for the considered algorithms as shown in Figure 1. From the results, AdaOAM has once again converged faster than the other three, which is consistent to our theoretical analysis that AdaOAM can effectively exploit second order information in achieving a faster convergence and more robust performances.

## Evaluation of Parameter Sensitivity

We now examine the parameter sensitivity of the AdaOAM algorithm. Since the AdaOAM algorithm provides a per-feature adaptive learning rate at each iteration, the value for the learning rate $\eta$ is less important than it is with the standard SGD. Due to the page limit, we randomly select the results of four datasets in our study on the learning rate parameter $\eta$. The results obtained are summarized in Figure 2.

In (Gao et al. 2013), the authors claimed that OPAUC was insensitive to the parameter settings. From Figure 2, we find that AdaOAM is even more robust to the learning rate settings. The updating strategy by OPAUC is based on simple SGD, which usually requires efforts of tuning the proper learning rate parameter. However, the adaptive gradient strategy is theoretically sound for learning rate adaptation because it takes full advantages of historical gradient information available in learning phase. Therefore, AdaOAM is less sensitive to parameter settings. In other words, AdaOAM has the natural effect of decreasing the learning rate with increasing iterations.

## Conclusion and Future Work

In this paper, we have proposed an Adaptive Online AUC Maximization approach, which considered the historical component-wise gradient information for more efficient and adaptive learning. Our proposed algorithm employs the second order information to speed up the convergence rate of online AUC maximization, and is less sensitive to parameter setting than simple SGD updating. We show that the regret bound for online AUC maximization can be significantly reduced via the proposed algorithm. We have conducted an extensive set of experiments by comparing with a number of competing online AUC optimization algorithms on both benchmark datasets and real-world anomaly detection datasets. The promising empirical results thus demonstrated the effectiveness of our proposed algorithm.

For future work, we aim to improve the regret bound for AdaOAM. In our case, the L2-norm regularization could have been recognized as a strongly convex function to increase convexity. According to (Bartlett, Hazan, and Rakhlin 2007; Shalev-Shwartz, Singer, and Srebro 2007), the regularization term helps meliorate the algorithm's actions $\mathbf{x}_t$ to the optimal. Therefore, the second order adaptive gradient updating strategy may achieve $O(\log T/T)$ rate in strongly convex cases because gradient and projection steps are both used in the AdaOAM algorithm.

## Acknowledgments

This research is partially supported by Multi-plAtform Game Innovation Centre (MAGIC) in Nanyang Technological University. MAGIC is funded by the Interactive Digital Media Programme Office (IDMPO) hosted by the Media Development Authority of Singapore. IDMPO was established in 2006 under the mandate of the National Research Foundation to deepen Singapore's research capabilities in

interactive digital media (IDM), fuel innovation and shape the future of media.

## References


Bartlett, P. L.; Hazan, E.; and Rakhlin, A. 2007. Adaptive online gradient descent. In *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007*.

Calders, T., and Jaroszewicz, S. 2007. Efficient auc optimization for classification. In *PKDD*, 42–53.

Cesa-Bianchi, N., and Lugosi, G. 2006. *Prediction, learning, and games*. Cambridge University Press.

Cortes, C., and Mohri, M. 2003. Auc optimization vs. error rate minimization. In *NIPS*.

Crammer, K.; Dekel, O.; Keshet, J.; Shalev-Shwartz, S.; and Singer, Y. 2006. Online passive-aggressive algorithms. *Journal of Machine Learning Research* 7:551–585.

Crammer, K.; Kulesza, A.; and Dredze, M. 2009. Adaptive regularization of weight vectors. In *Advances in Neural Information Processing Systems 22: 23rd Annual Conference on Neural Information Processing Systems 2009. Proceedings of a meeting held 7-10 December 2009, Vancouver, British Columbia, Canada.*, 414–422.

Dredze, M.; Crammer, K.; and Pereira, F. 2008. Confidence-weighted linear classification. In *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008*, 264–271.

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

Gao, W., and Zhou, Z.-H. 2012. On the consistency of auc optimization. *CoRR* abs/1208.0645.

Gao, W.; Jin, R.; Zhu, S.; and Zhou, Z.-H. 2013. One-pass auc optimization. In *ICML (3)*, 906–914.

Hanley, J. A., and McNeil, B. J. 1982. The meaning and use of the area under of receiver operating characteristic (roc) curve. In *Radiology*.

Hoi, S. C. H., and Zhao, P. 2013. Cost-sensitive double updating online learning and its application to online anomaly detection. In *Proceedings of the 13th SIAM International Conference on Data Mining, May 2-4, 2013. Austin, Texas, USA.*, 207–215.

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

Hoi, S. C. H.; Wang, J.; and Zhao, P. 2014. LIBOL: a library for online learning algorithms. *Journal of Machine Learning Research* 15(1):495–499.

Joachims, T. 2005. A support vector method for multivariate performance measures. In *ICML*, 377–384.

Kar, P.; Sriperumbudur, B. K.; Jain, P.; and Karnick, H. 2013. On the generalization ability of online learning algorithms for pairwise loss functions. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, 441–449.

Michaelides, M. P., and Panayiotou, C. G. 2009. SNAP: fault tolerant event location estimation in sensor networks using binary data. *IEEE Trans. Computers* 58(9):1185–1197.

Orabona, F., and Crammer, K. 2010. New adaptive algorithms for online classification. In *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada.*, 1840–1848.

Peng, H.; Gates, C. S.; Sarma, B. P.; Li, N.; Qi, Y.; Potharaju, R.; Nita-Rotaru, C.; and Molloy, I. 2012. Using probabilistic generative models for ranking risks of android apps. In *ACM Conference on Computer and Communications Security*, 241–252.

Rosenblatt, F. 1958. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review* 65(6):386.

Rudin, C., and Schapire, R. E. 2009. Margin-based ranking and an equivalence between adaboost and rankboost. *Journal of Machine Learning Research* 10:2193–2232.

Shalev-Shwartz, S.; Singer, Y.; and Srebro, N. 2007. Pegasos: Primal estimated sub-gradient solver for SVM. In *Machine Learning, Proceedings of the Twenty-Fourth International Conference (ICML 2007), Corvallis, Oregon, USA, June 20-24, 2007*, 807–814.

Wang, J.; Zhao, P.; and Hoi, S. C. H. 2012. Exact soft confidence-weighted learning. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*.

Wang, J.; Zhao, P.; and Hoi, S. C. H. 2014. Cost-sensitive online classification. *IEEE Trans. Knowl. Data Eng.* 26(10):2425–2438.

Zhao, P., and Hoi, S. C. H. 2013. Cost-sensitive online active learning with application to malicious URL detection. In *The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013, Chicago, IL, USA, August 11-14, 2013*, 919–927.

Zhao, P.; Hoi, S. C. H.; Jin, R.; and Yang, T. 2011. Online AUC maximization. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, 233–240.

Zhao, P.; Hoi, S. C. H.; Wang, J.; and Li, B. 2014. Online transfer learning. *Artif. Intell.* 216:76–102.

Zhao, P.; Hoi, S. C. H.; and Jin, R. 2011. Double updating online learning. *Journal of Machine Learning Research* 12:1587–1615.

Zhou, Y., and Jiang, X. 2012. Dissecting android malware: Characterization and evolution. In *IEEE Symposium on Security and Privacy*, 95–109.